Contents

List of Figures

List of Tables

Acronyms

ACM Adaptive Case Management. 28, 29, 178

ADoc Adaptive Document. 29–32, 34, 39, 40

B-step Business step. 36, 46, 79, 90, 280

BA Business Artifact. 1–3, 5, 8, 11, 19, 23, 29–41, 47, 48, 59–61, 78, 87, 102, 193, 194, 198, 254

BALSA Business Artifacts with Lifecycle Services and Associations. 40, 48, 60, 193, 194, 198

BEL Business Entities with Lifecycle. 29

BELA Business Entity Lifecycle Analytics. 34

BPEL Business Process Execution Language. 2, 16, 21, 22, 34, 78, 315, 330, 331, 339, 342, 344, 354, 355

BPM Business Process Management. 2–5, 11–14, 20–23, 34, 36, 38, 104, 106, 109, 110, 114, 141, 178, 194, 197, 325, 327

BPMN Business Process Model and Notation. 2, 4, 15, 16, 21, 26, 34, 60, 65–68, 70, 74, 75, 78, 123, 178, 194, 195, 199, 262, 265, 316, 322, 331, 332, 334, 338, 343, 348, 351–355

BPMS Business Process Management System. 13, 14

BPR Business Process Re-engineering. 12, 305, 308, 309

CEP Complex Event Processing. 24

CHERRIES Checklist for Reporting Results of Internet E-Surveys. 158, 391

CMIS Content Management Interoperability Services. 9

SLR Systematic Literature Review. 4, 5, 24, 103–108, 112, 114, 119, 120, 122–124, 128–133, 135, 136, 157, 158, 160, 191–200, 305, 311, 316, 343, 347, 358, 391–394

SPEM Software Process Engineering Metamodel. 311, 313, 316

StArt State of the Art through Systematic Reviews. 104, 108, 110, 113, 114, 117, 118, 392

UML Unified Modeling Language. 22, 29, 65–68, 70, 71, 74, 75, 194, 195, 197, 199, 262, 265, 280, 288

UML AD Unified Modeling Language Activity Diagram. 2, 4, 16, 21, 22, 26, 65–68, 70, 194, 195, 322, 334, 338, 351, 354

UNISA University of South Africa. 109, 114, 117, 133, 178, 390, 391

WF-Net Workflow Net. 21, 320, 322, 337–339

WfM Workflow Management. 12

XPDL XML Process Definition Language. 16, 21

YAWL Yet Another Workflow Language. 16, 21, 322, 323, 325, 326, 334, 351, 354

Symbols

$\mathcal{A}$ is a set of annotators used to indicate characteristics of elements in $\mathcal{E}$. 54–59, 139–143, 145–147, 149–154

$\widehat{\mathscr{A}}$ set of TODO(manual activation) in case type. 52, 53, 55, 80, 85, 94, 101, 300, 301

A number of arcs [Men+06] (see Table B.12). 310, 312, 323, 324, 327, 329, 330, 368

$\mathcal{A}_z$ a subset of $\mathcal{A}$ for $\ulcorner z \urcorner$. 56, 57, 142

AC activity complexity [Car05b] (see Table B.8). 137, 316, 367

ACC average connector cohesion [Dan+96] (see Table B.1). 307, 359

ACCSA average cognitive complexity of structured activity [Muk+10b] (see Table B.30). 129, 345, 358

Ach GSM function from *Mst*to a finite, non-empty sets of *Stry*called *achieving sentry* of m. xxi, 43, 44, 79–82, 86

$\boldsymbol{Ach}^R$ GSM binary relationship of $Mst \times Stry$ implementing Ach. 44

ADP average degree of place [Mao10a] (see Table B.31). 122, 346, 359

ADT average degree of transition [Mao10a] (see Table B.31). 346, 359

AEC average event cohesion [Dan+96] (see Table B.1). 306, 359

AEPC average execution path complexity [Mao10a] (see Table B.31). 346, 359

$\mathbf{AEPC}_{CI}$ average execution path complexity based on cognitive informatics [Mao10b] (see Table B.32). 347, 359

AFC average function cohesion [Dan+96] (see Table B.1). 306, 359

AGD average gateway degree [La +11b] (see Table B.35). 351, 359

AIR assign/invoke ratio [HB09] (see Table B.28). 343, 358

ALSA average length of structured activity [Muk+10b] (see Table B.30). 129, 345, 359

$\mathbf{AND}_j$ number of AND joins [Men+06] (see Table B.12). 121, 323, 368

$\mathbf{AND}_s$ number of AND splits [Men+06] (see Table B.12). 323, 368

APL average path length [Kre10] (see Table B.33). 348, 360

ATC activity type count [LÏ5] (see Table B.40). 356, 358

$\boldsymbol{Att}$ a set of GSM attributes partitioned into *data attributes* Att_{data} and *status attributes* Att_{status}.. xxii, 41–43, 45, 46, 79, 82

$\boldsymbol{Att}_{data}$ GSM data attributes. 42, 79, 86

$\boldsymbol{Att}_{milestones}$ GSM set of Boolean milestone attributes (subset of Att_{status}). 42–45, 81, 86

$\boldsymbol{Att}_{stages}$ GSM set of Boolean stage attributes (subset of Att_{status}). 42, 45, 81, 86

$\boldsymbol{Att}_{status}$ GSM set of Boolean status attributes. 42, 43, 46

$\boldsymbol{Att}_{\mathcal{T}}$ a set of GSM attributes partitioned into *data attributes* Att_{data} and *status attributes* Att_{status} derived from a CMMN Case type.. 85, 86

$\mathcal{B}$ behaviour in a case type. xxii, 50, 54, 85, 100, 299

$\mathcal{B}^P$ behavior $\mathcal{B}$ for program $\lceil P \rceil$. 54

BAD basic activity distribution [LÏ5] (see Table B.40). 122, 356, 360

$\mathcal{B}_\Gamma$ behaviour in a CMMN Case type derived from a GSM Artifact type. 79

$\mathcal{C}$ a tuple that represents a case model $\mathcal{C} = \langle \mathcal{E}, \mathcal{U}, \mathcal{V}, \mathcal{A} \rangle$. 54, 56, 138–142, 144–152

C McCabe's cyclomatic number [Bor+09a] (see Table B.27). 341, 364

C sequence control-flow complexity [Fu+10] (see Table B.29). 343, 385

c process cohesion [RV04] (see Table B.7). 122, 314, 381

$C'(\mathcal{M})$ cumulative method complexity for method $\mathcal{M}$ [RB96]. 68, 70, 74, 75

$C'(\mathcal{M}_T)$ technique complexity for method $\mathcal{M}$ [RB96]. 69, 70

CA activity coupling [Gar+03] (see Table B.5). 121, 312, 313, 358

$\bar{\mathbf{C}}_A$ average activity complexity [Tja99] (see Table B.3). 309, 358

CADAC cognitive activity depth arc control flow [Çoş14] (see Table B.39). 122, 129, 137, 355, 360

CAR copy/assign ratio [HB09] (see Table B.28). 343, 362

CAS CMMN annotator size metric. xiii, 139, 140, 144–147, 157, 160–164, 166, 169, 173, 191

CAS_{DAC} CMMN Number of autocomplete decorators. (see Table 7.2). 139

CAS_{DC} CMMN Number of collapsed decorators. (see Table 7.2). 139

CAS_{DCP} CMMN Number of collapsed planning table decorators. (see Table 7.2). 139

CAS_{DE} CMMN Number of expanded decorators. (see Table 7.2). 139

CAS_{DEP} CMMN Number of expanded planning table decorators. (see Table 7.2). 139

CAS_{DMA} CMMN Number of manual activation decorators. (see Table 7.2). 139

CAS_{DR} CMMN Number of required decorators. (see Table 7.2). 139

CAS_{DRN} CMMN Number of repetition decorators. (see Table 7.2). 139

CAS_{MC} CMMN Number of case markers. (see Table 7.2). 139

CAS_{MH} CMMN Number of non-blocking human markers. (see Table 7.2). 139

CAS_{MHB} CMMN Number of participant markers. (see Table 7.2). 139

CAS_{MP} CMMN Number of process markers. (see Table 7.2). 139

CAS_{MT} CMMN Number of timer markers. (see Table 7.2). 139

CAS_{SE} CMMN Number of entry criteria sentries. (see Table 7.2). 139

CAS_{SX} CMMN Number of exit criteria sentries. (see Table 7.2). 139

CC CMMN complexity metric. xiii, 141, 142, 149–155, 157, 159–164, 166, 169, 172–174, 177, 178, 390, 394

CC cross-connectivity [Van+08b] (see Table B.21). 334, 362

CC McCabe's cyclomatic number [Mao10a] (see Table B.31). 346, 363

CCBP cognitive complexity [Muk+10b] (see Table B.30). 122, 129, 137, 345, 360

$\mathtt{CC_{YAWL}}$ cognitive complexity for YAWL [GL06a] (see Table B.14). 122, 137, 323, 325, 360

CFC CFC [Abr+10] (see Table B.34). 349, 350, 362

CFC Cardoso's control flow complexity [HB09] (see Table B.28). 342, 362

CFC control-flow complexity [Car05b] (see Table B.8). 122, 129, 315, 316, 321, 322, 324, 327, 330, 331, 333, 334, 338, 343, 350, 354, 362

$\texttt{CFC}_{abs}$ absolute control-flow complexity [Car08] (see Table B.20). 333, 362

$\texttt{CFC}^{\texttt{BPEL}}_{\texttt{Process}}$ control-flow complexity for BPEL process [Car07b] (see Table B.17). 331, 362

$\texttt{CFC}_{rel}$ relative control-flow complexity [Car08] (see Table B.20). 333, 334, 384

CH connector heterogeneity [Men07] (see Table B.16). 330, 361

ch process cohesion [Van+08a] (see Table B.24). 337, 381

CI complexity index [Lat01] (see Table B.4). 122, 310, 360

CL CMMN length metric. xiii, 140, 141, 147–149, 157, 160–164, 166, 169, 173

CLA connectivity level between activities [Rol+06b] (see Table B.9). 129, 318, 361

CLP connectivity level between pools [Rol+06b] (see Table B.9). 129, 318, 361

CNC CNC [Abr+10] (see Table B.34). 349, 360

CNC coefficient of network complexity [Car+06] (see Table B.10). 321, 329, 360

CNC coefficient of network connectivity [Men07] (see Table B.16). 329, 360

$\texttt{CNC}_K$ coefficient of network complexity (Kaimann) [Lat01] (see Table B.4). 310, 360

$\texttt{CNC}_P$ coefficient of network complexity (Pascoe) [Lat01] (see Table B.4). 121, 310, 360

Complexity complexity of process integrated with rules [Klu15] (see Table B.41). 356, 361

Complexity variety-based complexity [CP08] (see Table B.23). 336, 386

$\texttt{Complexity}_{CF}$ control flow complexity [Ant+11] (see Table B.36). 352, 362

$\texttt{Complexity}_{DF}$ data flow complexity [Ant+11] (see Table B.36). 353, 363

CP coupling [Van+07a] (see Table B.19). 333, 362

cp process coupling [Van+08a] (see Table B.24). 337, 382

CS CMMN size metric. xiii, 138, 140, 144–147, 152, 157, 160–164, 166, 169, 173, 191

CS_{DI} CMMN Number of case file items. (see Table 7.2). 139

CS_{OC} CMMN Number of connectos. (see Table 7.2). 139

CS_{PDT} CMMN Number of discretionary tasks. (see Table 7.2). 139

CS_{PE} CMMN Number of event listeners. (see Table 7.2). 139

CS_{PM} CMMN Number of milestones. (see Table 7.2). 139

CS_{PT} CMMN Number of tasks. (see Table 7.2). 139

CS_{SC} CMMN Number of cases. (see Table 7.2). 139

CS_{SDS} CMMN Number of discretionary stages. (see Table 7.2). 139

CS_{SPF} CMMN Number of plan fragments. (see Table 7.2). 139

CS_{SS} CMMN Number of stages. (see Table 7.2). 139

CTS CMMN total size metric. 140, 144, 146, 147, 157, 163, 191, 192, 196, 200, 395

CW cognitive weight [GL06b] (see Table B.11). 137, 323, 360

CYC cyclicity [Men07] (see Table B.16). 330, 363

Cycle cycle [Men+06] (see Table B.12). 323, 362

$\mathcal{D}$ set of data attributes in a Case type. xxv, 50, 53, 54, 85, 90, 91, 100, 101, 299

Δ density of the process graph [Men07] (see Table B.16). 326, 330, 334, 364

D process difficulty [Car+06] (see Table B.10). 137, 311, 314, 315, 321, 354, 382

$\mathcal{D}^P$ data attributes $\mathcal{D}$ for program $\lceil P \rceil$. 54

$\mathcal{D}_{casefile}$ is a single element set, in a case type, containing the case file. 50, 85, 299

$\mathcal{D}_{container}$ is a set of data containers in a case type. 50, 85, 299

$\mathcal{D}_{discrete}$ is a set of all of the discrete data in a case type. 50, 86, 299

DC number of decisions [HB09] (see Table B.28). 343, 373

$\overline{\mathtt{d}_C}$ average degree of connectors [Men07] (see Table B.16). 122, 330, 359

$\widehat{\mathtt{d}_C}$ maximun degree of a connector [Men07] (see Table B.16). 329, 366

$\mathtt{d}_e$ density of an EPC [Men06] (see Table B.13). 324, 325, 363

Depth depth [La +11b] (see Table B.35). 137, 351, 364

DFI data flow intensity [HB09] (see Table B.28). 343, 363

$\mathcal{D}_\Gamma$ set of data attributes in a CMMN Case type derived from a GSM Artifact type. 79

diam diameter [La +11b] (see Table B.35). 351, 364

diam diameter [Men07] (see Table B.16). 328, 364

DMC different modeling concepts [La +11b] (see Table B.35). 137, 352, 364

DOP degree of parallelism [HB09] (see Table B.28). 342, 363

δ_S aggregated depth fraction [Lv09] (see Table B.25). 339, 358

DSM Durfee square metric [KN12] (see Table B.37). 136, 354, 364

$\mathtt{d}_w$ density of a workflow-net [Men06] (see Table B.13). 324, 363

$\mathtt{d}_y$ density of a YWAL model [Men06] (see Table B.13). 324, 363

$\mathcal{E}$ a set of modeling elements. (see Table 3.3). 54, 55, 57–59, 138–147, 149–155

$\widehat{\mathscr{E}}$ entry criteria binary relationship in case type. 52, 54, 55, 80, 85, 86, 99, 101, 300, 301

$\mathcal{E}_z$ a subset of $\mathcal{E}$ for $\ulcorner z \urcorner$. 56–59, 138, 142

EAC extension activity count [LÏ5] (see Table B.40). 122, 356, 364

ECaM extended Cardoso [Lv09] (see Table B.25). 338, 364

ECyM extended cyclomatic [Lv09] (see Table B.25). 338, 364

EDB extension activity distribution [LÏ5] (see Table B.40). 356, 365

$\mathtt{E}_{end}$ number of end events [Men+06] (see Table B.12). 323, 371

$\mathtt{E}_{int}$ number of internal events [Men+06] (see Table B.12). 323, 375

EM extended measure [Sob99] (see Table B.2). 308, 364

$\mathtt{E}_{start}$ number of start events [Men+06] (see Table B.12). 323, 379

EType GSM set of event types (or message types). 41–43, 79

$\boldsymbol{EType}_{gen}$ GSM set of generated events (or messages). 42, 45, 46, 86, 100, 101, 299

$\boldsymbol{EType}_{inc}$ GSM set of incoming events (or messages). 42, 45, 46, 79, 80, 86

$\boldsymbol{EType}_{\mathfrak{I}}$ GSM set of event types (or message types) derived from a CMMN Case type. 85, 86

$\mathcal{E}v$ set of event listeners in a case type. 50–54, 85, 92–94, 101, 299, 300

$\mathcal{E}v^P$ set of event listeners in program $\lceil P \rceil$. 54

$\mathcal{E}v_\Gamma$ set of event listeners in a case type derived from a GSM Artifact type. 79, 80

F number of functions [Men+06] (see Table B.12). 323, 324, 331, 338, 373

Φ structuredness ratio [Men07] (see Table B.16). 329, 386

$\mathtt{F}_i\mathtt{F}_o$ fan-in / fan-out [GL06b] (see Table B.11). 323, 365

Flexibility flexibility [Tja99] (see Table B.3). 122, 309, 365

Γ A GSM Type. 41, 43, 44, 79, 84, 85, 88–101, 299–301

$\Gamma_{\mathcal{T}}$ A GSM Type derived from a CMMN Case type. 85

GREM global ripple effect measure [Sob99] (see Table B.2). 308, 365

Guards GSM function from *Stg*to a finite, non-empty set of *Stry*. xxvii, 43, 44, 79, 80, 86, 101, 301

$\boldsymbol{Guards}^R$ GSM relationship of *Stg* × *Stry* implementing *Guards*. 44

$\mathcal{H}$ hierarchy of a case type. 50–52, 85, 86, 101, 299, 300

$\mathcal{H}_\Gamma$ hierarchy of a case type derived from a GSM Artifact type. 79, 80

HH height of hierarchy [Kre10] (see Table B.33). 137, 348, 365

HKM Henry and Kafura metric [Abr+10] (see Table B.34). 349, 365

$\mathtt{H}_{SC}$ structural complexity [Che08] (see Table B.22). 336, 385

IF information flow complexity [SH14] (see Table B.38). 355, 365

IF4BP information flow complexity [Muk+10b] (see Table B.30). 129, 344, 365

$\mathtt{I}_{f^V}$ Dehmer's graph entropy [Bor+09a] (see Table B.27). 341, 363

Integration integration [Tja99] (see Table B.3). 309, 365

Inv GSM function from *Mst*to a finite set of *Stry*called *invalidating sentry* of *m*. xxvii, 43, 44, 79–82, 86

$\boldsymbol{Inv}^R$ GSM relationship of *Mst* × *Stry* implementing *Inv*. 44

JC join complexity [Men+06] (see Table B.12). 323, 324, 365

JSR join-split-ratio [Men+06] (see Table B.12). 324, 327, 365

k process coupling [RV04] (see Table B.7). 122, 315, 382

$\mathcal{L}$ a set of case elements $\mathcal{L} \subset \mathcal{M}$. (see Table 3.3). 56–59, 138, 140, 145–147, 149, 151

Λ maximum depth of all nodes [Men07] (see Table B.16). 122, 329, 366

$\mathtt{LBC}_T$ log-based complexity [Car07a] (see Table B.18). 332, 365

Lcyc Life cycle model of a GSM type. 41, 43, 79

$\boldsymbol{Lcyc}_{\mathcal{J}}$ Life cycle model of a GSM type derived from a CMMN Case type. 85, 86

$\mathcal{M}$ a set of scope elements where $\mathcal{M} \subset \mathcal{E}$. (see Table 3.3). 55–59, 140, 141, 144

$\widehat{\mathscr{M}}$ manual activation relationship in case type. 52, 55, 80, 85, 93–95, 101, 300, 301

$\mathcal{M}$ Method $\mathcal{M}$ [RB96]. 69, 70

$M_{\mathcal{M}}$ the model M of a method $\mathcal{M}$ [RB96]. 70

M_T the model M of a technique T is a six-tuple $M_T = \langle O, P, R, X, r, p \rangle$ [RB96]. 69, 70

`MaxND` maximum nesting depth [GL06b] (see Table B.11). 121, 322, 366

`MCC` McCabe's cyclomatic number [Car+06] (see Table B.10). 321, 354, 363

`MCC` McCabe's cyclomatic number [HB09] (see Table B.28). 342, 363

`MeanND` mean nesting depth [GL06b] (see Table B.11). 122, 322, 366

$\boldsymbol{\mathcal{M}i}$ set of milestones in a case type. 50–54, 80, 85, 86, 92–94, 101, 299, 300

$\boldsymbol{\mathcal{M}i}^P$ set of milestones in program $\lceil P \rceil$. 54

$\boldsymbol{\mathcal{M}i}_\Gamma$ set of milestones in a case type derived from a GSM Artifact type. 79

`MM` connector mismatch [Men07] (see Table B.16). 330, 361

`MM` modularization measure [Sob99] (see Table B.2). 121, 308, 366

`MO` modules overhead [La +11b] (see Table B.35). 351, 366

`MS` model size [La +11b] (see Table B.35). 351, 373

Mst GSM set of milestones. xxi, xxvii, xxxvi, 41, 43, 44, 79–82

$\boldsymbol{Mst}_{\mathcal{J}}$ GSM set of milestones derived from a CMMN Case type. 85, 86

$\widehat{\mathscr{N}}$ repetition relationship in case type. 52, 53, 55, 80, 85, 95, 101, 300, 301

`N` process length [Car+06] (see Table B.10). 137, 149, 152, 310, 321, 327, 329, 330, 334, 335, 338, 345, 382

$n(O_{\mathcal{M}})$ count of meta-model objects $O_{\mathcal{M}}$ for method $\mathcal{M}$ [RB96]. 68–70, 74, 75

$n(O_T)$ count of meta-model objects $O_{\mathcal{M}}$ for technique T [RB96]. 69, 70

$n(P_{\mathcal{M}})$ count of meta-model properties $P_{\mathcal{M}}$ for method $\mathcal{M}$ [RB96]. 68–70, 74, 75

$n(P_T)$ count of meta-model properties $P_{\mathcal{M}}$ for technique T [RB96]. 69, 70

$n(R_{\mathcal{M}})$ count of meta-model relationships $R_{\mathcal{M}}$ for method $\mathcal{M}$ [RB96]. 68–70, 74, 75

$n(R_T)$ count of meta-model relationships $R_{\mathcal{M}}$ for technique T [RB96]. 69, 70

NA number of activities [Abr+10] (see Table B.34). 137, 349, 367

NA number of activities [GL06b] (see Table B.11). 137, 322, 367

NA number of activities [Gar+03] (see Table B.5). 121, 128, 137, 312, 313, 322, 367

NAf number of artifacts [Abr+10] (see Table B.34). 349, 368

NAS number of associations [Abr+10] (see Table B.34). 349, 368

NC number of connectors [Abr+10] (see Table B.34). 349, 370

NCD number of complex decision/merge [Rol+06b] (see Table B.9). 128, 318, 320, 369

NCG number complex gateways [Abr+10] (see Table B.34). 349, 369

NCl number of classes [Kre10] (see Table B.33). 137, 348, 368

NCS number of collapsed sub-process [Rol+06b] (see Table B.9). 129, 136, 317, 319, 369

NCSA number of collapsed ad-hoc sub-process [Rol+06b] (see Table B.9). 317, 319, 368

NCSC number of collapsed compensation sub-process [Rol+06b] (see Table B.9). 317, 319, 369

NCSL number of collapsed looping sub-process [Rol+06b] (see Table B.9). 317, 319, 369

NCSMI number of collapsed multiple instance sub-process [Rol+06b] (see Table B.9). 317, 319, 369

ND nesting depth [Abr+10] (see Table B.34). 349, 350, 354, 366

NDO number of data objects [Abr+10] (see Table B.34). 136, 349, 381

NDOIn number of input data objects [Rol+06b] (see Table B.9). 129, 318–320, 374

NDOOut number of output data objects [Rol+06b] (see Table B.9). 128, 318–320, 376

NDOp number of different objects in all possible paths [Abr+10] (see Table B.34). 349, 370

NDos number of domains [Kre10] (see Table B.33). 348, 370

NDRA number of precedence dependences between activities [Gar+03] (see Table B.5). 312, 313, 377

NDWP number of dependences [Gar+04a] (see Table B.6). 128, 313, 370

NDWPIn number of input dependences [Gar+04a] (see Table B.6). 313, 374

NDWPOut number of output dependences [Gar+04a] (see Table B.6). 313, 377

NE number of edges [Kre10] (see Table B.33). 348, 368

NECaE number of end cancel event [Rol+06b] (see Table B.9). 318, 320, 370

NECoE number of end compensation event [Rol+06b] (see Table B.9). 318, 320, 370

NEDDB number of exclusive decision/merge data-based [Rol+06b] (see Table B.9). 128, 318, 320, 372

NEDEB number of exclusive decision/merge event-based [Rol+06b] (see Table B.9). 128, 318, 320, 372

NEE number of end events [Abr+10] (see Table B.34). 349, 371

NEEE number of end error event [Rol+06b] (see Table B.9). 318, 320, 371

NELE number of end link event [Rol+06b] (see Table B.9). 318, 320, 371

NEMsE number of end message event [Rol+06b] (see Table B.9). 129, 318, 320, 371

NEMuE number of end multiple event [Rol+06b] (see Table B.9). 318, 320, 371

NEN number of edges per node [Kre10] (see Table B.33). 348, 370

NENE number of end none event [Rol+06b] (see Table B.9). 318, 320, 371

NETE number of end terminate event [Rol+06b] (see Table B.9). 318, 320, 372

NF number of feedbacks [Kre10] (see Table B.33). 348, 372

NFO number of flow objects [Abr+10] (see Table B.34). 349, 372

NFOSP number of flow objects in smallest path [Abr+10] (see Table B.34). 349, 373

NG number of groups [Abr+10] (see Table B.34). 349, 373

NGa number of gates (fan-in + fan-out) [Abr+10] (see Table B.34). 349, 373

NGDE number gateway data based exclusive [Abr+10] (see Table B.34). 349, 366

NGDI number gateway data based inclusive [Abr+10] (see Table B.34). 349, 366

NGEE number gateway event based exclusive [Abr+10] (see Table B.34). 349, 366

NH number of handles [GL06b] (see Table B.11). 322, 373

NICaE number of intermediate cancel event [Rol+06b] (see Table B.9). 317, 319, 374

NICoE number of intermediate compensation event [Rol+06b] (see Table B.9). 317, 319, 374

NID number of incusive decision/merge [Rol+06b] (see Table B.9). 128, 318, 320, 373

NIE number of intermediate events [Abr+10] (see Table B.34). 349, 374

NIEE number of intermediate error event [Rol+06b] (see Table B.9). 317, 319, 374

NIG number of input gates (fan-in) [Abr+10] (see Table B.34). 349, 374

NILE number of intermediate link event [Rol+06b] (see Table B.9). 317, 319, 375

NIMsE number of intermediate message event [Rol+06b] (see Table B.9). 129, 317, 319, 375

NIMuE number of intermediate multiple event [Rol+06b] (see Table B.9). 317, 319, 375

NINE number of intermediate none event [Rol+06b] (see Table B.9). 317, 319, 375

NIRE number of intermediate rule event [Rol+06b] (see Table B.9). 318, 319, 375

NIS number of independent sets [Kre10] (see Table B.33). 348, 373

NITE number of intermediate timer event [Rol+06b] (see Table B.9). 129, 318, 319, 375

NL number of lanes [Abr+10] (see Table B.34). 350, 378

NL number of lanes [Rol+06b] (see Table B.9). 318, 319, 378

NMF number of message flows [Abr+10] (see Table B.34). 350, 376

NMF number of message flows [Rol+06b] (see Table B.9). 129, 318, 376

NN number of nodes [Kre10] (see Table B.33). 137, 348, 367

NOA number of activities [Car+06] (see Table B.10). 137, 321, 354, 355, 367

NOA number of activities [HB09] (see Table B.28). 137, 342, 367

NOAC number of activities and control [HB09] (see Table B.28). 342, 367

NOAC number of activities and control flow elements [Car+06] (see Table B.10). 321, 367

NOACC number of activities, control structures, and copy [HB09] (see Table B.28). 342, 367

NOAJS number of activities joins and splits [Car+06] (see Table B.10). 321, 368

NOBA number of basic activities [Muk+10b] (see Table B.30). 137, 344, 345, 380

NOBP number of objects in biggest path [Abr+10] (see Table B.34). 350, 376

NOG number of output gates(fan-out) [Abr+10] (see Table B.34). 350, 377

NOSA number of structured activities [Muk+10b] (see Table B.30). 344, 345, 373

N$_P$ number of places [Mao10a] (see Table B.31). 346, 377

NP number of pools [Abr+10] (see Table B.34). 350, 377

NP number of pools [Rol+06b] (see Table B.9). 129, 318, 377

NPF number of parallel fork/join [Rol+06b] (see Table B.9). 128, 318, 320, 377

NPG number parallel gateways [Abr+10] (see Table B.34). 350, 377

NPP number of possible paths [Abr+10] (see Table B.34). 350, 377

NPR number of roles [Gar+03] (see Table B.5). 128, 312, 313, 378

N$_S$ number of services [Mao10a] (see Table B.31). 346, 378

NSE number of start events [Abr+10] (see Table B.34). 350, 379

NSF number of sequence flows [Abr+10] (see Table B.34). 350, 370

NSF number of sequence flows [Rol+06b] (see Table B.9). 318, 370

NSFA number of sequence flows between activities [Rol+06b] (see Table B.9). 129, 318, 378

NSFE number of sequence flows from events [Rol+06b] (see Table B.9). 129, 318, 378

NSFG number of sequence flows from gateways [Rol+06b] (see Table B.9). 128, 318, 378

NSL number of swimlanes [Abr+10] (see Table B.34). 350, 377

NSLE number of start link event [Rol+06b] (see Table B.9). 317, 319, 379

NSMsE number of start message event [Rol+06b] (see Table B.9). 317, 319, 379

NSMuE number of start multiple event [Rol+06b] (see Table B.9). 317, 319, 379

NSNE number of start none event [Rol+06b] (see Table B.9). 317, 319, 379

NSP number of sub-processes [Abr+10] (see Table B.34). 136, 350, 380

NSRE number of start rule event [Rol+06b] (see Table B.9). 317, 319, 379

NSTE number of start timer event [Rol+06b] (see Table B.9). 317, 319, 380

NSTP number of steps (tasks) [Gar+03] (see Table B.5). 312, 380

NT number of simple tasks [Rol+06b] (see Table B.9). 137, 317, 320, 380

$\texttt{N}_T$ number of transitions [Mao10a] (see Table B.31). 346, 380

NTA number of text annotations [Abr+10] (see Table B.34). 350, 380

NTC number of compensation tasks [Rol+06b] (see Table B.9). 317, 320, 369

NTIE total number of intermediate events [Rol+06b] (see Table B.9). 128, 319, 320, 374

NTL number of looping tasks [Rol+06b] (see Table B.9). 317, 320, 375

NTMI number of multiple instances tasks [Rol+06b] (see Table B.9). 317, 320, 376

NTSE total number of start events [Rol+06b] (see Table B.9). 128, 319, 320, 379

NUN number of unconnected nodes [Kre10] (see Table B.33). 348, 380

NWP number of work products [Gar+03] (see Table B.5). 128, 312, 313, 381

O_T a finite set of object types [RB96]. 69

$O_\mathcal{M}$ meta-model object $O_\mathcal{M}$ for method $\mathcal{M}$ [RB96]. 72

$\texttt{OR}_j$ number of OR joins [Men+06] (see Table B.12). 324, 376

$\texttt{OR}_s$ number of OR splits [Men+06] (see Table B.12). 324, 376

$\prod$ separability ratio [Men07] (see Table B.16). 314, 329, 332, 385

$\mathfrak{P}(O_T)$ is the power set of set O [RB96]. 69

P_T a finite set of property types [RB96]. 69

p_T a mapping of property types P to non-property types $\{O \cup R \cup X\}$ [RB96]. 69

PC process complexity [Abr+10] (see Table B.34). 350, 381

$\texttt{P}_{CIM}$ process context independency metric [Kho+09] (see Table B.26). 340, 382

$\texttt{P}_{CM}$ process coupling metric [Kho+09] (see Table B.26). 340, 382

PDOPIn rate of input data object over the total of data objects [Rol+06b] (see Table B.9). 319, 383

PDOPOut rate of output data object over the total of data objects [Rol+06b] (see Table B.9). 319, 383

PDOTOut rate of output data object over the total of tasks [Rol+06b] (see Table B.9). 319, 384

$\widehat{\varphi}$ set of conditions in case type. 52, 53, 80, 85, 101, 300

φ GSM Sentry **if** expression. 45, 52, 53, 81, 82, 94, 97, 101, 301

PLT rate of pools and lanes over the total of tasks [Rol+06b] (see Table B.9). 319, 384

$P_{\mathcal{M}}$ meta-model property $P_{\mathcal{M}}$ for method $\mathcal{M}$ [RB96]. 72, 74

PSM perfect square metric [KN12] (see Table B.37). 137, 354, 381

$\widehat{\mathscr{R}}$ required relationship in case type. 52, 53, 55, 80, 85, 95, 101, 300, 301

$\mathcal{R}$ tuple of case type rules. 50–52, 85, 299, 300

R Randi'c's connectivity index [Bor+09a] (see Table B.27). 341, 382

$\mathcal{R}^P$ case type rules in program $\lceil P \rceil$. 55

R_T a finite set of relationship types [RB96]. 69

r_T a mapping of relationships to objects labeled by roles [RB96]. 69

RD relational density [Kre10] (see Table B.33). 348, 384

RDWPIn ratio between input dependencies and number of dependencies [Gar+04a] (see Table B.6). 129, 313, 383

RDWPOut ratio between output dependencies and the number of dependencies [Gar+04a] (see Table B.6). 128, 313, 383

$\mathcal{R}_\Gamma$ tuple of case type rules derived from a GSM Artifact type. 79, 80

ρ process coupling/cohesion ratio [Van+08a] (see Table B.24). 337, 382

ρ process coupling/cohesion ratio [RV04] (see Table B.7). 315, 382

$R_{\mathcal{M}}$ meta-model relationship $R_{\mathcal{M}}$ for method $\mathcal{M}$ [RB96]. 74

RPRA ratio of process roles and activities [Gar+03] (see Table B.5). 128, 312, 313, 384

RSTPA ratio of steps and activities [Gar+03] (see Table B.5). 312, 383

RT restrictiveness estimator [Car+06] (see Table B.10). 321, 384

RT restrictiveness estimator [Lat01] (see Table B.4). 310, 384

RWPA ratio of work products and activities [Gar+03] (see Table B.5). 129, 312, 313, 384

$\widehat{\mathscr{S}}$ set of sentries in a case type. 52, 53, 80, 85, 86, 101, 300, 301

S cyclomatic number [Lat01] (see Table B.4). 129, 310, 327, 329–331, 363

S structuredness [La +11b] (see Table B.35). 351, 386

$\mathtt{S}_{A}$ number of arcs [Men07] (see Table B.16). 328, 368

SA structured activities [HB09] (see Table B.28). 343, 385

$\mathtt{S}_{C}$ number of connectors [Men07] (see Table B.16). 328, 370

SC split complexity [Men+06] (see Table B.12). 324, 362

$\mathtt{S}_{C_{and}}$ number of AND connectors [Men07] (see Table B.16). 328, 368

SCBP structural complexity [Muk+10b] (see Table B.30). 129, 345, 385

$\mathtt{S}_{C_{or}}$ number of OR connectors [Men07] (see Table B.16). 328, 376

$\mathtt{S}_{C_{xor}}$ number of XOR connectors [Men07] (see Table B.16). 328, 381

$\mathtt{S}_{E}$ number of events [Men07] (see Table B.16). 137, 328, 372

$\mathtt{S}_{E_E}$ number of end events [Men07] (see Table B.16). 328, 371

$\mathtt{S}_{E_S}$ number of start events [Men07] (see Table B.16). 328, 379

$\mathtt{S}_{F}$ number of functions [Men07] (see Table B.16). 328, 373

Simplicity simplicity [Tja99] (see Table B.3). 309, 385

$\mathtt{Size}_{A}$ activity size of process [Ant+11] (see Table B.36). 137, 352, 358

$\mathtt{Size}_{CF}$ size of control flow graph [Ant+11] (see Table B.36). 353, 367

$\mathtt{Size}_{DF}$ size of data flow graph [Ant+11] (see Table B.36). 353, 385

$\mathtt{Size}_{R}$ resource size [Ant+11] (see Table B.36). 353, 384

$\mathtt{S}_{j_{and}}$ number of AND joins [Men07] (see Table B.16). 328, 368

$\mathtt{S}_{j_{or}}$ number of OR joins [Men07] (see Table B.16). 328, 376

$\mathtt{S}_{j_{xor}}$ number of XOR joins [Men07] (see Table B.16). 328, 381

SM structuredness [Lv09] (see Table B.25). 122, 338, 339, 386

$\mathtt{S}_{N}$ number of nodes [Men07] (see Table B.16). 137, 328, 367

$\mathtt{S}_{S_{and}}$ number of AND splits [Men07] (see Table B.16). 328, 368

$\mathtt{S}_{S_{or}}$ number of OR splits [Men07] (see Table B.16). 328, 376

$\mathtt{S}_{S_{xor}}$ number of XOR splits [Men07] (see Table B.16). 328, 381

$\mathcal{S}t$ set of stages in a Case type. 50–54, 56, 80, 85, 86, 88, 89, 93, 101, 299

$\mathcal{S}t^P$ set of stages in program $\lceil P \rceil$. 54, 56

$\mathcal{S}t_{discretionary}$ is a set of discretionary stages in a case type. 50–53, 85, 91–93, 96, 101, 299, 300

$\mathcal{S}t_{fragment}$ is a set of plan fragments in a case type. 50, 51, 85, 96, 101, 299, 300

$\mathcal{S}t_{planned}$ is a set of planned stages in a case type. 50–53, 85, 86, 91–93, 101, 299, 300

$\mathcal{S}t_{\Gamma}$ set of stages in a case type derived from a GSM Artifact type. 79

Stg GSM set of stages. xxvii, xxxvi, xxxvii, 41–44, 79–81, 88

$\boldsymbol{Stg}_{\mathcal{T}}$ GSM set of stages derived from a CMMN Case type. 85, 86

Stry GSM set of sentries. xxi, xxvii, xxxvii, 41, 43, 44, 79, 80, 101, 300, 301

$\boldsymbol{Stry}_{\mathcal{T}}$ GSM set of sentries derived from a CMMN Case type. 85, 86

Submilestones GSM function from *Stg*to non-empty, finite subsets of *Mst*. xxxvi, 43, 44, 79, 80, 86

$\boldsymbol{Submilestones}^R$ GSM relationship of $Stg \times Mst$ implementing *Submilestones*. 43, 44

Substages GSM function from *Stg*to a finite subset of *Stg*. xxxvi, 43, 79–81, 86, 88, 101, 300

$\boldsymbol{Substages}^R$ GSM relationship of $Stg \times Stg$ implementing *Substages*. 43

$\mathcal{T}$ a CMMN Type. 50, 53, 54, 79, 84, 85, 100, 101, 299

T Technique T [RB96]. 69, 70

`T` number of trees in a graph [Lat01] (see Table B.4). 311, 314, 315, 324, 327, 329, 333, 335, 338, 380

$\mathcal{T}a$ set of tasks in a case type. xxxvi, 50–54, 80, 85, 86, 91–93, 299, 300

$\mathcal{T}a^P$ set of tasks in program $\lceil P \rceil$. 54

$\mathcal{T}a_{discretionary}$ is a set of discretionary tasks in a case type. 51, 85, 96, 101, 300

$\mathcal{T}a_{planned}$ is a set of planned tasks in a case type. 51, 53, 101, 300

$\mathcal{T}a_{\Gamma}$ set of tasks in a case type derived from a GSM Artifact type. 79

Tsk GSM set of tasks. xxxvii, 41–43

Tasks GSM function from the *atomic stages* in *Stg*to $\mathcal{T}\!a$. xxxvii, 43, 79, 80, 86

Tasks^R GSM relationship of *Stg* × *Tsk* implementing *Tasks*. 43

Tsk$_{\mathcal{T}}$ GSM set of tasks derived from a CMMN Case type. 85, 86

TBAC total basic activity count [LÏ5] (see Table B.40). 137, 356, 380

Terminators GSM function *Stg*to a finite, non-empty sets of *Stry*. xxxvii, 43, 44, 79, 80, 86, 101, 301

Terminators^R GSM relationship of *Stg* × *Stry* implementing *Terminators*. 44

$\mathcal{T}_\Gamma$ a CMMN Case type derived from a GSM Artifact type. 79

TNA total number of activities [Rol+06b] (see Table B.9). 137, 317, 367

TNCS total number of collapsed sub-processes [Rol+06b] (see Table B.9). 128, 136, 319, 369

TNDO total number of data objects [Rol+06b] (see Table B.9). 136, 319, 320, 381

TNE number of events [Abr+10] (see Table B.34). 137, 350, 372

TNE total number of events [Rol+06b] (see Table B.9). 128, 137, 320, 372

TNEE total number of end events [Rol+06b] (see Table B.9). 128, 320, 371

TNG number of gateways [Abr+10] (see Table B.34). 350, 373

TNG total number of gateways [Rol+06b] (see Table B.9). 128, 320, 373

TNS transfer number per service [Mao10a] (see Table B.31). 346, 386

TNT number of tasks [Abr+10] (see Table B.34). 137, 350, 367

TNT total number of tasks [Rol+06b] (see Table B.9). 137, 318–320, 367

TS token split [Men07] (see Table B.16). 330, 386

TSAC total structured activity count [LÏ5] (see Table B.40). 137, 356, 385

$\mathcal{U}$ a binary scope relationship in which two elements x and y in $\mathcal{E}$ are related if and only if they are contained in the same scope ($[\![x, y]\!] \in \mathcal{U}$ is an unordered pair). 54, 57–59, 144, 148–150

$\mathcal{U}_z$ a subset of $\mathcal{U}$ for $\ulcorner z \urcorner$. 56, 57

$\mathcal{V}$ is a binary relationship in which two elements x and y in $\mathcal{E}$ are related if and only if an event from one (x) triggers the other (y) ($\langle x, y\rangle \in \mathcal{V}$ is an ordered pair). 54, 57–59

V process volume [Car+06] (see Table B.10). 137, 321, 334, 338, 339, 382

$\mathcal{V}_z$ a subset of $\mathcal{V}$ for $\ulcorner z \urcorner$. 56, 57

W Wiener's index [Bor+09a] (see Table B.27). 341, 386

W_i Weight for element i (part of the calculation for CC). 141, 142, 149–151, 153, 154

WH width of hierarchy [Kre10] (see Table B.33). 137, 348, 386

$\widehat{\mathscr{X}}$ exit criteria binary relationship in case type. 52, 80, 85, 86, 100, 101, 300, 301

Ξ sequentiality ratio [Men07] (see Table B.16). 329, 385

$\wr x$ the graphical representation of x in a CMMN model. 54

$\widehat{\mathscr{X}}^P$ exit criteria binary relationship in program $\lceil P \rceil$. 54, 55

x, y, z denote elements of $\mathcal{E}$. 59

X_T a finite set of role types [RB96]. 69

ξ GSM Sentry **on** event expression. 45, 47, 52, 53, 81, 94, 95, 97, 101, 301

$\texttt{XOR}_j$ number of XOR joins [Men+06] (see Table B.12). 324, 381

$\texttt{XOR}_s$ number of XOR splits [Men+06] (see Table B.12). 324, 381

$\ulcorner z \urcorner$ is a module defined by z ($z \in \mathcal{M}$). 57, 141

Chapter 1

Introduction

The purpose of this study was to explore complexity metrics for artifact-centric business process models. Opportunities for research into data-centric processes were opened with the introduction of case handling by van der Aalst and Berens [vB01]. In addition, the introduction of Business Artifact (BA) by Kumaran et al. [Kum+03] and Nigam and Caswell [NC03] expanded data-centric research opportunities into declarative processes. Today, research on declarative processes now includes understandability [Fah+10; Hai+13], process discovery [di +15; Mag+13], verification [Bur+12; Gon+15] and hybrid process models [de +15a; Par+13].

The publication of Case Management Model and Notation (CMMN) [OMG14a] by the Object Management Group (OMG) [Gro17] in 2014 introduced a standard specification for declarative processes [Mar+13]. However, complexity metrics for declarative process models have not as of yet been studied. This is significant because case management products based on the standards of the new declarative process model are appearing in the market place, thus creating a need to characterize their complexity. To close this gap, this study explores complexity metrics for the newly published artifact-based declarative case management modeling standard, CMMN. Being exploratory research, this thesis touches on what may seem, at first glance, like unrelated topics.

1.1 Motivation

Process Modeling is used by organizations to describe and document their policies and procedures [zI10]. In most cases, a process model describes a business process using a Graphical Process Model (GPM) in order to facilitate human communication and comprehension of the process across the organization [Swa07]. It is important for organizations to understand, manage, and improve their process portfolio. The tools and methods used to achieve these objectives are collectively referred to as Business Process Management (BPM) [zI10]. Process Modeling, in particular, is used by organizations for process improvement or to implement BPM projects [Ind+09a].

BPM technology provides a formal way to describe and automate business processes. Such processes consist of an organized set of business activities that are required to achieve a business goal. The activities can either be automatically executed by a computer system or manually executed by a person interacting with a computer system. Therefore, BPM can be defined as a network of value-added activities designed to achieve a business goal [Ko09].

Process Models allow employees to understand and communicate the operations of the organization. When a process model is automated, members of the Information Technology (IT) department must be able to communicate with the business users in order to facilitate understanding of the actual process. Having an understanding of the process model is not only important for communication between the IT department and business users, but it also facilitates the communication and transfer of knowledge regarding BPM technology throughout the organization [Swa07].

Complexity Metrics can be used to describe how easy or difficult it is for a human being to understand a business process [GL06b]. The complexity of a process model may reflect the true complexity of the problem, but often the process model is unnecessarily complex and overly complex process models create problems within an organization. When users, process engineers or systems analysts do not understand the process, and the process is automated, errors are highly probable [Lv09]. Therefore, identifying overly complex processes is important in managing BPM projects.

Process Modeling approaches can be categorized into imperative and declarative process models. Imperative process models require all the execution alternatives to be explicitly described in the model, using control-flow to specify *how* the process should work, while declarative process models only describe the essential characteristics and constraints of the process by specifying *what* the process should do [Fah+09; Fah+10; Pic+12]. Imperative process model languages include the Business Process Execution Language (BPEL) [OAS07], the Unified Modeling Language Activity Diagram (UML AD) [OMG09c], and the Business Process Model and Notation (BPMN) [OMG13]. Declarative process model languages include DECLARE [Pes+07; van+09] and Guard-Stage-Milestone (GSM) [Hul+11b].

The declarative GSM approach to BA processes was introduced by Hull et al. [Hul+11b]. A key aspect of the BA approach is the tied coupling of the process with its data represented by an information model which is described by a life cycle [Sol+13b]. There are multiple approaches to the artifact centric business process paradigm, including GSM [Kun+15]. The

CMMN specification lists two GSM papers [Hul+11a; Hul+11b] as part of the non-normative references, which implies that GSM influenced the CMMN specification.

Extensive research has been conducted on the complexity metrics of imperative process modeling languages. Between 1996 and 2015, starting with the work produced by Daneva et al. [Dan+96] and ending with that done by Kluza [Klu15], more than a hundred metrics for imperative process modeling languages were proposed. However, as far as this researcher is aware, no research has been done on the complexity metrics for declarative process model languages.

1.2 Problem Statement

Research into complexity metrics for imperative process modeling languages has been very successful [van13], however research into complexity metrics for declarative process models has not been conducted. With the introduction of the CMMN standard based on GSM the use of declarative processes is becoming mainstream, and the need for complexity metrics will increase. Unless research on complexity metrics for declarative processes is advanced our knowledge of process technology complexity will be incomplete. A better understanding of complexity metrics for declarative processes will improve our understanding of these processes, and will help vendors and users of the new CMMN standard.

The working thesis assumes that although CMMN models are very different from traditional BPM models a useful set of complexity metrics can be derived from the research that has been conducted on BPM complexity metrics and applied to declarative processes models such as CMMN models.

1.3 Objectives and Contributions

The main objectives of this study were to identify and validate complexity metrics for artifact-based process models. In particular, this study focused on CMMN [OMG14a], a type of BA [Mar+13] heavily influenced by GSM. The study addressed the following objectives:

1. To formalize CMMN as the basis to identify metrics.
2. To formalize the relationship between CMMN and GSM.
3. To assess the method complexity of CMMN.
4. To analyze the applicability of BPM complexity metrics to CMMN.

5. To identify complexity metrics for CMMN.
6. To validate the identified complexity metrics.

1.3.1 Contributions

This research makes contributions in the areas of formalizing CMMN using first-order logic, proposing complexity metrics for CMMN, identifying and comparing the model complexity of CMMN against other process notations, and clarifying the relationship between CMMN and GSM. The contributions made by this research can be summarized as follows:

1. Formal descriptions of a CMMN program, a case type, and a case model using first-order logic and based on the CMMN standard were developed. These were required in order to formally reason about the notation, and to define and validate the CMMN complexity metrics.
2. Formal transformations in two directions between CMMN case types and GSM artifact types were developed. The transformations helped to clarify the relationships between CMMN and GSM. The transformation from a GSM artifact type into a case type was relatively simple and straight forward to describe. The resulting case type modeled using CMMN was visually similar to the original artifact type, which made it easy for a human being to understand. However, the transformation of a case type into a GSM artifact type was far more complex. Despite this, the transformation still allows CMMN to use the formal operational semantics of GSM. It also allows the formal verification work developed for GSM to be applied to CMMN.
3. The method complexity of CMMN was evaluated using the meta-model-based approach which was introduced by Rossi and Brinkkemper [RB96]. The results were compared to other popular process methods, including BPMN, UML AD, and Event-driven Process Chain (EPC) [van99], all of which have undergone similar evaluations by other authors.
4. A set of metrics for CMMN was developed. After a Systematic Literature Review (SLR) of BPM complexity metrics, a set of metrics for the CMMN standard was identified and validated. Most BPM metrics are based on the control-flow of the process, which makes these metrics unsuitable for the CMMN declarative style. The metrics were theoretically validated using the formal framework for software measurements as defined by Briand et al. [Bri+96] and the complexity metrics were further validated using Weyuker's properties [Wey88] for software complexity measures.

1.3.2 Significance

A better understanding of complexity metrics for CMMN will improve the understanding of the declarative process and have practical implications for implementors and users of the new CMMN standard. This research advances the literature on the areas of method complexity [Ind+09b; Rec+09; SC02], complexity metrics for process models [Car08; Çoş14; Muk+10a], GSM transformations [Bel+12; Ev15; Sol+12], declarative processes [Hai+13; Pic+12; Pre+14], and research on CMMN [Hau+15; Kur+15; Sch+13] by characterizing CMMN method complexity, identifying complexity metrics for CMMN, and exploring the relationship between CMMN and GSM. This is important because:

- CMMN is a new process standard and results from this study could inform the evolution of CMMN.
- By formalizing CMMN, this study provides the basis for future research on the standard.
- Products based on CMMN will benefit from having the ability to incorporate the useful set of metrics produced by this research.
- The characterization of the CMMN method complexity provides new insights into method complexity and advances the ability to compare method notations.
- By clarifying and formalizing the relationship between GSM and CMMN, new areas of research have become possible including the verification of CMMN processes, and the visualization of GSM processes.
- The research contributes to the understanding of model complexity for BPM by adding declarative processes to the current knowledge regarding model complexity for imperative processes.

1.4 Outline

The thesis has been organized as follows:

Chapter 2 introduces the key concepts and basic terminology relevant to the development of this thesis. Chapter 3 introduces BA-based processes including the GSM type and its evolution into the CMMN standard. Chapter 4 explores CMMN method complexity and compares it to other modeling methods. Chapter 5 describes transformations between CMMN and GSM. Chapter 6 presents a SLR on BPM complexity metrics and describes the frameworks commonly used to validate complexity metrics. Chapter 7 analyses the applicability of BPM metrics for CMMN, proposes a set of CMMN metrics and sub-metrics, and describes the theoretical validation of the proposed metrics. Chapter 8 describes an experiment that was conducted in order to complete the empirical validation for the proposed metrics. Chapter 9 concludes the thesis, outlines the contributions of this research, and suggests further areas of research. The structure of this thesis is depicted in Figure 1.1.

1.5 Previous Publications

This thesis is the result of several years of research. during which time partial results were published. The chapters of this thesis include material from those publications.

Peer-reviewed publications:

[Mar+16] M. A. Marin et al. "Implementing Deterministic Finite State Machines using Guard-Stage-Milestone". In: *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference (SAICSIT'16)*. Johannesburg, South Africa: ACM Press, New York, USA, 2016.

The paper was prepared by Mike A. Marin. Corrections and reviews were made by all of the authors. All of the authors read and approved the final manuscript. Chapter 3, Chapter 5 and Appendix D file 30 (FSM-2-GSM.pdf) use material from this publication.

[Mar+15b] M. A. Marin et al. "Metrics for the Case Management Modeling and Notation (CMMN) Specification". In: *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference (SAICSIT'15)*. Ed. by R. J. Barnett et al. Stellenbosch, South Africa: ACM Press, New York, USA, 2015.

The paper was prepared by Mike A. Marin. Corrections and reviews were made by all of the authors. All of the authors read and approved the final manuscript. Chapter 7 uses material from this publication.

[Mar+15a] M. A. Marin et al. "Case Management: An Evaluation of Existing Approaches for Knowledge-Intensive Processes". In: *4^{th} International Workshop on Adaptive Case Management and other non-workflow approaches to BPM (AdaptiveCM)*. Innsbruck, Austria, 2015, pp. 1–12.

The paper was prepared by Mike A. Marin and Matheus Hauden. Mike A. Marin contributed the provenance of case management, the comparison of the different case management definitions in the literature with the Knowledge Intensive Processes (KiP) characteristics, the comparison of KiP with CMMN, and the mapping of KiP requirements against CMMN. Matheus Hauden contributed the motivation, the synthesis of a case management definition, and the conclusion of the paper. Corrections and reviews were made by all

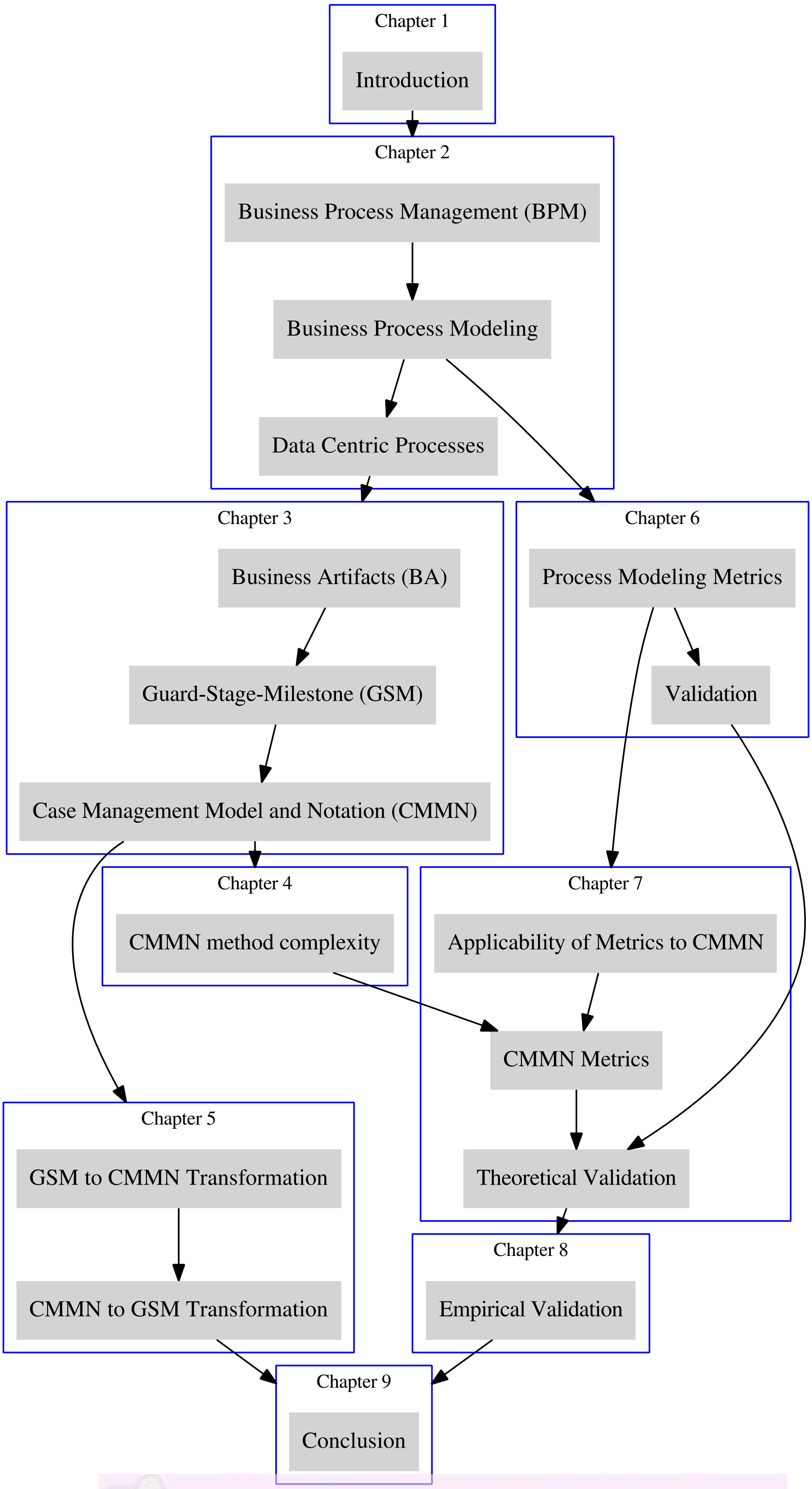


Figure 1.1: Thesis structure

of the authors. All of the authors read and approved the final manuscript. Chapter 2 uses material from this publication.

[Mar+14b] M. A. Marin et al. "Measuring Method Complexity of the Case Management Modeling and Notation (CMMN)". in: *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference (SAICSIT'14)*. Ed. by J. P. van Deventer et al. Centurion, Gauteng, South Africa: ACM Press, New York, USA, 2014, pp. 209–216.

The paper was prepared by Mike A. Marin. Corrections and reviews were made by all of the authors. All of the authors read and approved the final manuscript. Chapter 4 uses material from this publication.

[Mar+13] M. A. Marin et al. "Data Centric BPM and the Emerging Case Management Standard: A Short Survey". In: *Business Process Management Workshops*. Ed. by M. La Rosa and P. Soffer. Vol. 132. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2013, pp. 24–30. DOI: `10.1007/978-3-642-36285-9_4`.

The first draft of the paper was prepared by Mike A. Marin. Richard Hull and Roman Vaculín contributed towards the provenance of BAs. Corrections and reviews were made by all of the authors. All of the authors read and approved the final manuscript. Chapter 3 uses material from this publication.

Non peer-reviewed publications:

[Mar16a] M. A. Marin. "Introduction to the Case Management Model and Notation (CMMN)". in: *Computing Research Repository* (2016). arXiv: `abs/1608.05011`

The paper is an extend version of the tutorial used as part of the experiment described in Chapter 8.

[Mar16c] M. A. Marin. *The Case Management Model and Notation (CMMN) version 1.0 Tutorial*. Accessed: August 08, 2016. 2016. URL: `http://cmmn.byethost4.com`

The online tutorial was prepared as part of the experiment described in Chapter 8.

[MH15] M. A. Marin and M. Hauder. "Case Management: A Data Set of Definitions". In: *Computing Research Repository* (2015). arXiv: `abs/1507.04004`.

The dataset was prepared by Mike A. Marin. Corrections and reviews were made by all of the authors. All of the authors read and approved the final manuscript. Chapter 3 uses material from this publication.

[MB15] M. A. Marin and J. A. Brown. "Implementing a Case Management Modeling and Notation (CMMN) System using a Content Management Interoperability Services (CMIS) compliant repository". In: *Computing Research Repository* (2015). arXiv: `abs/1504.06778`.

The paper was prepared by Mike A. Marin and Jay A. Brown. Mike A. Marin's contribution focused on the overall implementation requirements of CMMN. Jay A. Brown contributed the section that examined the interaction with a Content Management Interoperability Services (CMIS) [OAS13] repository. Corrections and reviews were made by all of the authors. All of the authors read and approved the final manuscript. Chapter 3 uses material from this publication.

Chapter 2

Background

This chapter introduces the key concepts and basic terminology used in Business Process Management (BPM), business process modeling, and case management that were relevant to the development of this thesis.

This chapter is organized as follows. Section 2.1 provides an introduction to BPM. Section 2.2 introduces process modeling and its different aspects, including model types, focus, and the role of process metrics. This section also provides background material that will be referred to in Chapter 6. Section 2.3 provides an introduction to case management and offers background material that will be used in Section 3.3. Section 2.4 describes the evolution of Business Artifact (BA)s and their influence on the Case Management Model and Notation (CMMN) [OMG14a] specification, and also provides background material for Chapter 3. Please note that some of the material that appears in Section 2.3 was previously published in Marin et al. [Mar+15a], and that Section 2.4 expands on material that was previously published in Marin et al. [Mar+13]. The examples referred to in Section 2.2.1 were also previously published in Marin et al. [Mar+14b].

2.1 Business Process Management

Organizations are more likely to achieve their business goals when employees, Information Technology (IT) systems and other enterprise resources are well integrated. BPM plays an important role in facilitating this integration [Wes12]. Many organizations adopt BPM in order to improve their operations and to better serve their customers [MM15]. This adoption of BPM technology has created a vibrant software and consultancy market [Pit15]. Today most activities that an organization performs are supported by information systems, and BPM helps to organize these activities [Wes12]. This has led to a very healthy and saturated BPM software market that, as described by Fleming and Silverstein [FS15], grew at

8.2% in 2014 to an estimated 3.2 billion US dollars. BPM provides the "concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business processes" [Wes12]. It combines computer science and management science with both communities showing increased interest in the topic [van13; Wes12]. BPM has evolved into an interdisciplinary field that combines methods from business administration, organizational theory, computer science, and information systems [Pit15] and is motivated by real applications [Rei+10a]. However, the interdisciplinary mix of methodologies and approaches in BPM can lead to confusion [Rei+10a]. The scope and definition of BPM is often still confused with Business Process Re-engineering (BPR), and Workflow Management (WfM) [Ko09]. According to de Bruin and Doebeli [dD10], the lack of clarity around BPM is attributable to at least three interpretations of the term which include:

1. BPM as a technology for process management. This interpretation views BPM as a set of software tools that automate and manage processes.
2. BPM as an approach to manage the life cycle of processes. This interpretation views BPM as managing the process life cycle. This view combines the technological view with the management of that technology.
3. BPM as an organizational process. This interpretation perceives BPM as a management discipline that uses a process centric view. Therefore, this interpretation does not view BPM from a technological perspective.

This thesis focuses on BPM from a technological perspective. This perspective views BPM as an extension of WfM systems. Traditionally, both of these technologies (i.e., WfM and BPM) were based on Petri Net token semantics [van15].

BPM technology automates business processes by organizing the IT assets in the order required to achieve a desired business outcome. Business processes refer to how organizations deliver products and services to their customers, with organizations trying to outperform each other by improving their business processes and executing them better [Dum+13]. Therefore, business processes are central to BPM [Pit15], and are important to understand if one wants to know how the organization operates [Wes12]. Furthermore, these processes are important because they have an influence on the organization's IT systems [Wes12]. This thesis uses Weske's [Wes12] definition of process, which describes a business process as a set of coordinated activities that realize a business goal and are performed by an organization using a technical environment. A process is sometimes referred to as a workflow.

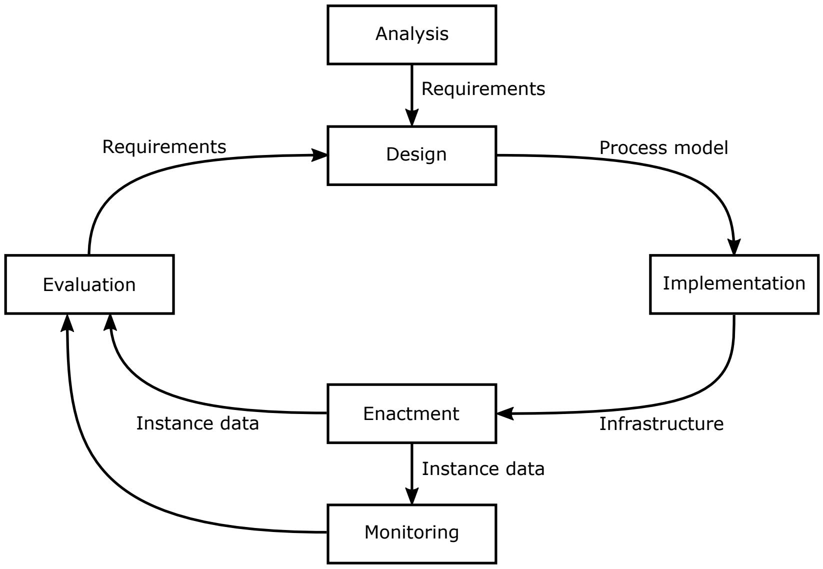


Figure 2.1: BPM life cycle (adapted from Mendling [Men08])

2.1.1 Process Lifecycle

Business processes or workflows are IT assets and therefore have a life cycle. BPM literature describes multiple versions of a business process life cycle [de +14], as can be evidenced in work done by Bouneffa and Ahmad [BA14], di Ciccio et al. [di +14], Dumas et al. [Dum+13], Schulte et al. [Sch+15], and Weske [Wes12]. This thesis uses Mendling's [Men08] *apud* [zur04] version illustrated in Figure 2.1. This version's life cycle views BPM as technology and describes the outputs of each phase, including the process model which is the focus of this thesis. The life cycle in Figure 2.1 consists of six phases:

Analysis. This phase focuses on project goals, and the organizational structures in the environment of the business process [zur04]. The business process and its requirements are identified in this phase. The requirements for the business process constitute the output of this phase.

Design. In this phase the process is engineered and includes the identification and definition of activities and the ordering needed to implement the requirements that were identified in the previous phase, which will achieve the business goal [Men08; zur04]. Business process modeling and validation techniques, including simulation and verification, are used in this phase [Wes12]. The output of this phase is a process model.

Implementation. During this phase, the necessary infrastructure required to support the business process, including the Business Process Management System (BPMS), is configured [zur04]. The implementation includes the integration of the business process with other information systems and employee training [Wes12]. The output of this phase is the infrastructure required to support the business process.

Enactment. In this phase the BPMS is able to create process instances and to execute them according to the process model that is being enacted [Wes12]. Resources needed by the process are allocated and participants in the process are presented with activities that they must complete [zur04]. The output of enactment is a set of instance data, normally stored in log files or database tables.

Monitoring. Two aspects are measured during monitoring. From the information system's perspective the operations and performance of the BPMS are measured, and from the organization's perspective process measurements are taken [zur04]. Visual monitoring tools are normally used to present the state of process instances [Wes12]. This monitoring phase normally occurs simultaneously with the enactment phase.

Evaluation. The information gathered during the enactment and monitoring phases is used for resource planning and to generate new requirements for the business process. Audit trails from the BPMS and measurements from the monitoring tools are used for staffing and resource planning, as well as to identify adjustments to the process [zur04].

This thesis focuses on process models that correspond with the output of the design phase.

2.2 Business Process Modeling

The activity of documenting and organizing the IT assets and the corresponding user interactions is called business process modeling or process modeling. Organizations make use of business process modeling to describe the business processes that are to be automated by describing the activities that need to be performed in order to achieve a business goal. Process modeling is an important activity in a successful BPM project [Fle+14]. A business process model, also referred to as a process model, is normally described in a visual manner and represents the way that business representatives conduct the operation of a business [BR05].

A process model visualizes the main activities of a process including the actors and systems involved in performing those activities [Rei+10b]. Modeling is a symbolic representation of a specific part of reality [Hen12]. Guizzardi [Gui05] defined the term "conceptualization" as the set of concepts used to describe the abstractions of a domain. A cognitive model refers to a particular situation abstracted using a conceptualization. These two concepts (i.e., conceptualization and cognitive model) reside in the mind of a person or a community of persons [Gui05]. A concrete artifact is required to communicate and document a cognitive model and this artifact is referred to as a model [Gui05]. To create a model a modeling language is required. Figure 2.2 describes the relationships between the conceptualization and the model.

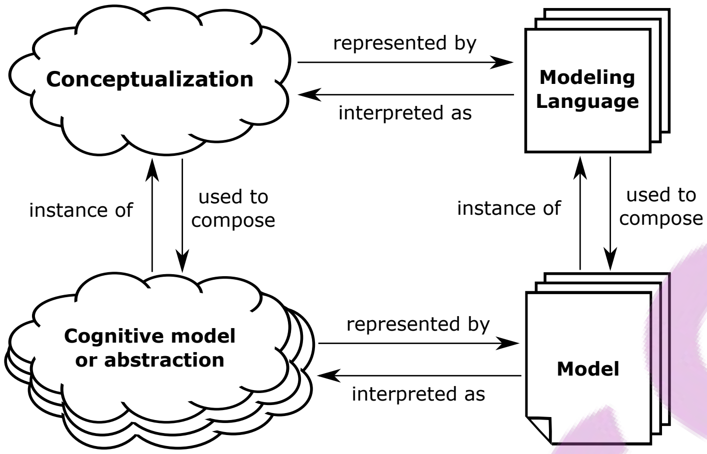


Figure 2.2: Relationships between the conceptualization and the model (adapted from [Gui05; Hen12])

According to Pittke [Pit15], the *apud* [Sta73] model has three characteristics:

- It contains a mapping of reality or mapping of other models.
- It is a reduction of reality because it only includes the information relevant to the user creating the model, therefore it does not include all of the characteristics and attributes of the original conceptualization.
- It is pragmatic because it is used to substitute for the original in specific situations.

According to Pinggera et al. [Pin+12a; Pin+12b], a person engaged in modeling a process follows an iterative process comprising three phases:

Comprehension. The person forms a mental model of the behavior being modeled.

Modeling. The person maps the mental model to the particular model.

Reconciliation. The person reorganizes the process model to enhance understandability.

2.2.1 Process Modeling Notations

A process model is described by a process modeling language that uses a notation with a set of visual symbols to represent the abstract concepts [Fig+13]. The notation consists of a visual vocabulary (graphical symbols), a visual grammar, and visual semantics [Moo09]. The visual vocabulary and grammar form the concrete syntax of the notation [Moo09]. Therefore, a process modeling notation focuses on the visual aspect of the process modeling language [Fig+13].

Not every business process language uses a process modeling notation. Examples of business process languages with process modeling notations include Business Process Model and Notation (BPMN) [OMG13], Unified Modeling Language Activity Diagram (UML AD) [OMG09c], Event-driven Process Chain (EPC) [van99], Business Process Execution Language (BPEL) [OAS07], Yet Another Workflow Language (YAWL) [Ter+10], and CMMN. Some business process languages do not provide a notation, these include XML Process Definition Language (XPDL) [WfM12] and BPEL.

For illustrative purposes, Figure 2.3 shows a simple insurance claim process described by Korherr [Kor08]. Korherr modeled the example in UML AD (Figure 2.3.a), EPC (Figure 2.3.b), and BPMN (Figure 2.3.c). This thesis added the CMMN model in order to demonstrate how the process can be modeled using CMMN (Figure 2.3.d). This very simple example is designed for illustrative purposes only and does not reflect the complexity of a real insurance claim process. The example involves seven human activities (`Record the claim`, `Calculate payment`, `Contact the garage`, `Check customer history`, `Review results`, `Pay for the damage`, and `Do not pay for the damage`).

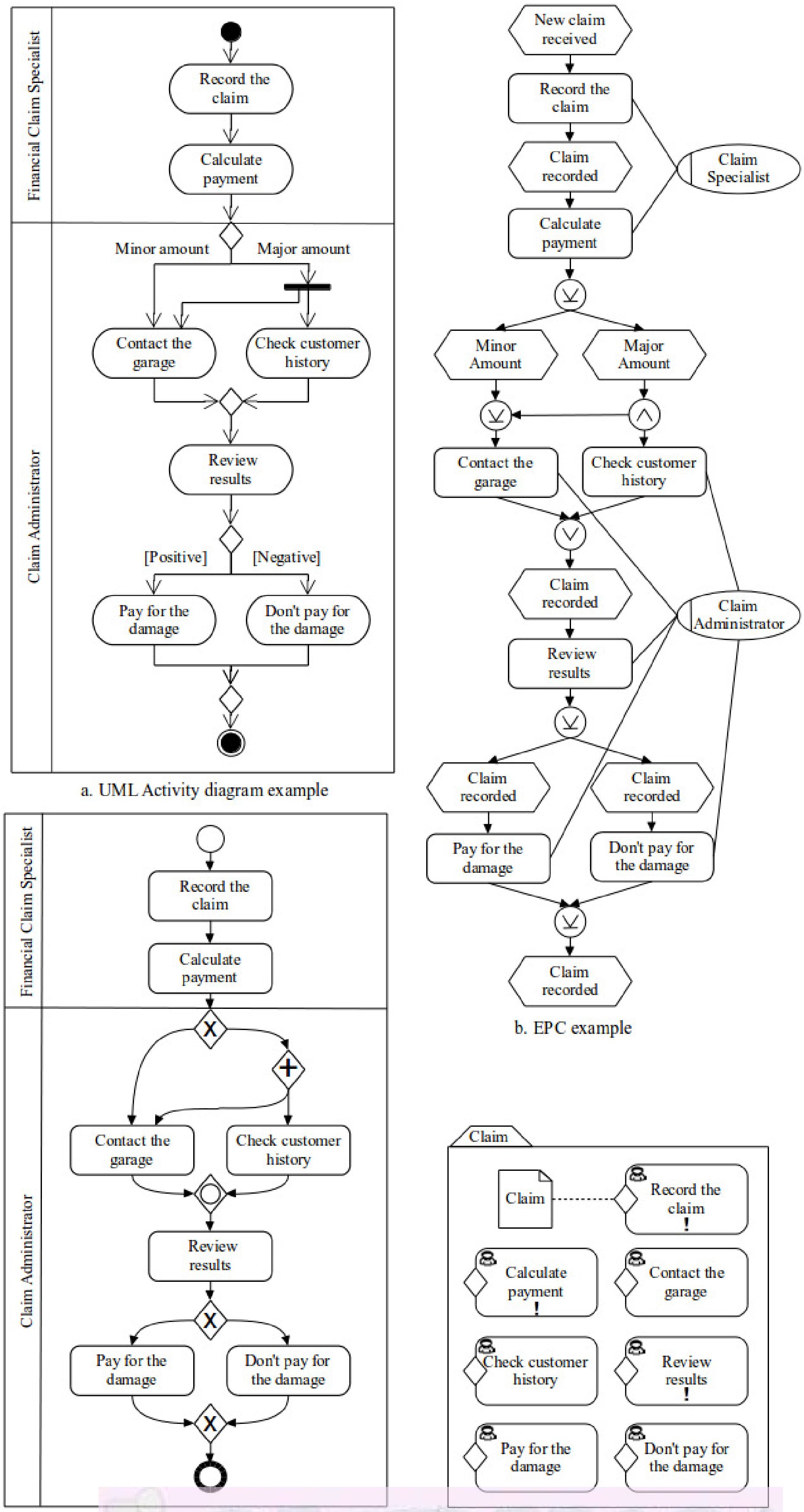
Note that UML AD and BPMN models are imperative process notations, while EPC and CMMN are event-driven notations. In this example, very little of the CMMN modeling notation was required. In CMMN, the seven human activities are modeled using human tasks (rounded rectangles with a human icon in the upper left corner) with entry criteria (diamond icon). Additionally, in this example a case file item representing the claim document was modeled, and a connector (dashed line) representing event propagation between case file item and entry criteria was used.

2.2.2 Modeling Types

The model should describe the activities of the process and provide answers to the following questions: why are they executed, how are they executed, who executes them, and what data do they manipulate [Gia01]? As described by Giaglis [Gia01], *apud* [Cur+92] as a modeling technique should be capable of describing one or more of the following perspectives:

1. The *functional perspective* indicates *what* activities are executed.
2. The *behavioral perspective* indicates *when* activities are executed.
3. The *organizational perspective* indicates *where* and by *whom* the activity is executed.
4. The *informational perspective* indicates the *data* produced or manipulated in the process.

Zachman [Zac87] proposed a framework for enterprise architecture, that was further refined by Sowa and Zachman [SZ92], consisting of a matrix describing the enterprise stakeholders' perspectives mapped into six communication questions. The six communication questions



a. UML Activity diagram example

b. EPC example

c. BPMN example

d. CMMN example

Figure 2.3: Example of a simple insurance claim process in four process notations

have been used by Caetano et al. [Cae+12], Pereira et al. [Per+11a; Per+11b], and Sousa et al. [Sou+07] as dimensions (i.e., what, where, who, when, why and how) to describe and identify the activities of the business process. Each dimension is associated with a corresponding concept (i.e., information entity, organizational unit, actor, business schedule, business goal, and business process) that represents the business process. Figure 2.4 presents the six dimensions and concepts that can be defined as follows:

How. The *business process* itself. This dimension focuses on the function of the process that can be decomposed into activities [Sou+07]. Corresponds with Giaglis's [Gia01] *functional perspective*.

What. The *information entities* consumed, used, or produced by the process. This dimension focuses on the data manipulated by the process [Sou+07]. Corresponds with Giaglis's [Gia01] *informational perspective*.

Why. The *business goals* that the process tries to achieve. This dimension focuses on the motivation for the process [Sou+07]. Not present in Giaglis [Gia01].

Where. The *organization units* where the process takes place. This dimension focuses on the network of locations in which the process operates [Sou+07]. Corresponds with Giaglis's [Gia01] *organizational perspective*.

When. The *business schedule* for the process. This dimension focuses on the time required to accomplish specific activities, and is related to events [Per+11a]. Corresponds with Giaglis's [Gia01] *behavioral perspective*.

Who. The *actors* performing the process, which can be systems or people. This dimension focuses on the roles used to support the process [Cae+12]. Corresponds with Giaglis's [Gia01] *organizational perspective*.

Giaglis's perspectives [Gia01] and Caetano et al.'s [Cae+12] dimensions and concepts can be used to describe the part of reality being expressed by a process model (called the domain of interest by Henderson-Sellers [Hen12]). A model is an abstraction of reality [Gui05] and represents a specific part of reality [Hen12]. Therefore, although initially developed to describe views and to identify activities in a business process, the work of Caetano et al. [Cae+12], Pereira et al. [Per+11a; Per+11b], and Sousa et al. [Sou+07] can be used to identify focus areas for process modeling notations. These focus areas produce different modeling approaches. Table 2.1 shows a mapping of Zachman's [Zac87] and Sowa and Zachman's [SZ92] dimensions with Caetano et al.'s [Cae+12] core concepts, Sousa et al.'s [Sou+07] focus areas, and Giaglis's [Gia01] perspectives along with the matching modeling approaches.

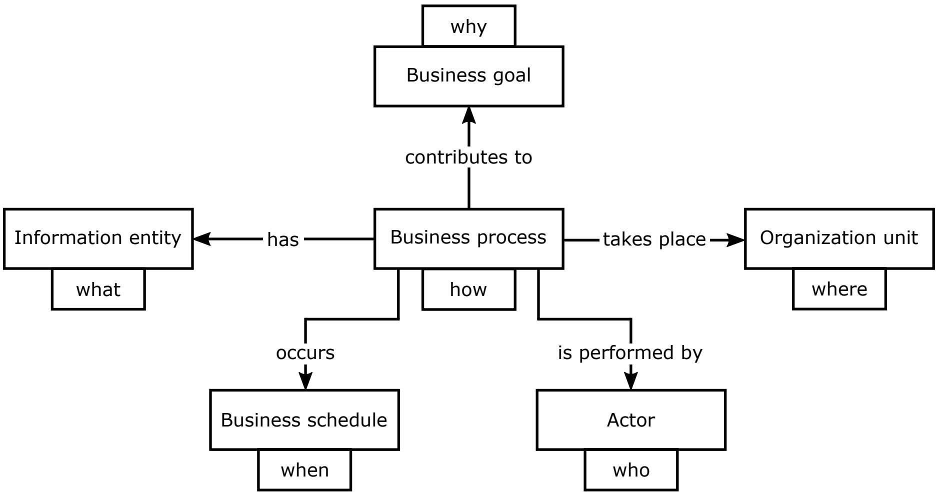


Figure 2.4: Business process six core concepts (adapted from Pereira et al. [Per+11a])

Table 2.1: Modeling Approaches (synthesized by this researcher)

Dimension [SZ92; Zac87]	**Core Concepts** [Cae+12; Per+11a; Per+11b]	**Focus** [Sou+07]	**Perspective** [Gia01]	**Modeling Approach**
How	Business process	Function	Functional (what)	imperative, declarative, hybrid, activity-centric, control-flow, constraint-based, rules-centric
What	Information entity	Data	Information	data-centric, data-driven, artifact-centric, data-aware, object-aware, object-centric, entity-centric
Why	Business goal	Motivation		goal-oriented
Where	Organizational unit	Network	Organizational	context-aware, context-oriented
When	Business schedule	Time	Behavioral	event-driven
Who	Actor	People	Organizational	user-driven, KiP, human-centric, people-centric collaborative, subject-oriented

Some authors have categorized process models using subsets of the dimensions as shown in Table 2.1. di Ciccio et al. [di +14] and La Rosa et al. [La +11a] describe three perspectives or dimensions: the control-flow perspective (corresponding with the *how* dimension), the data perspective (corresponding with the *what* dimension), and the resource perspective (corresponding with the *who* dimension). Barukh and Benatallah [BB14] describe the process representation (models or languages) as: activity-centric (uses flow-control corresponding with the *how* dimension), rules-centric (describes the business policy in a declarative manner, which also corresponds with the *how* dimension), and artifact-centric (uses BAs as first class citizens corresponding with the *what* dimension).

While other categorizations that use context not available in the model are still possible, these have not been explored in this thesis. For example, using flexibility paradigms [Buc+15; Mej+15; Reg+06] processes can be categorized into the *abstraction level of change*, *subject of change*, and *properties of change* [Mej+15; Reg+06].

2.2.2.1 Modeling Dimensions

While there are many other perspectives on process modeling, all of these can be categorized into the modeling approaches that have been described in Table 2.1. Although some researchers have focused on the imperative versus declarative dimension [Pic+12; Pre+14; Rei+13], others have focused on the activity-centric versus data-centric dimension [Ev15; Hai+13; Hai+16; van+13]. Additionally, there have also been researchers who have focused on the continuum between structured and unstructured dimensions [di +14], or from simple processes that ordinary workers can perform to complex processes that require experts [MF11]. This section explores these perspectives and their relationships with the six dimensions [SZ92; Zac87].

From Imperative to Declarative

As described in Table 2.1, modeling approaches based on the *how* dimension [SZ92; Zac87] include imperative, activity-centric, control-flow, declarative, constraint-based, rules-centric, and hybrid process modeling approaches. These modeling approaches can be placed on a continuum from imperative to declarative.

While some researchers focus on the imperative versus declarative dimension [Pic+12; Pre+14; Rei+13], others focus on the continuum that exists between the structured (imperative) and the unstructured (declarative) dimension [di +14]. Barukh and Benatallah [BB14] describe process paradigms as being structured, semi-structured, and unstructured. The distinction between imperative processes and declarative processes has its origins in programming languages. One of the many classifications of programming languages distinguishes between imperative and declarative languages [Aho+07; AM14]. The IEEE [IEE10] utilizes the same dichotomy between imperative and declarative, but uses the term “procedural programming language”, which has a similar meaning to the term “imperative” and “nonprocedural programming language” which has a similar meaning to the term “declarative”. This classification is commonly used by those researching BPM to classify process models [Pic+12; Pre+14; Rei+13]. In terms of this perspective, activity-centric and control-flow approaches are imperative, and constraint-based and rules-centric approaches are declarative. In addition, some researchers have proposed a mix or hybrid that combines the two approaches [CV15; Par+13; WS13].

Imperative processes explicitly define the sequence of steps in the process and their transitions, and strictly prescribe how the system should work [Pv06; Pre+14; Web+09]. Any sequence of steps that is not specified is disallowed [de +15a]. Burattin et al. [Bur+12] classify BPMN, UML AD, EPC, and BPEL as imperative process models. Other imperative business process languages include XPDL. Some authors use the terms “imperative process models” and “procedural process models” interchangeably [Fah+09; Pre+14]. Procedural BPM languages can be reduced to Workflow Net (WF-Net) [van95], which is a type of Petri Net introduced by van der Aalst [van95; van97]. As such, Petri Nets can be considered the theoretical foundation for imperative process models. Figure 2.3.c illustrates an example of an imperative process model that was designed using BPMN.

The activity-centric approach is an imperative approach that uses control-flow as its modeling technique [Ev15; Rus+14]. Chiao et al. [Chi+13] use the terms “activity-driven” and “activity-centric” interchangeably. BPMN and YAWL are considered to be activity-centric approaches [Rus+14] as is UML AD [Ev15].

In declarative processes all of the process steps are allowed unless forbidden by a rule [de +15a]. Therefore, declarative processes define the process based on its constraints [di +15], and not by using the process flow. de Giacomo et al. [de +15a] classify DECLARE [Pes+07; van+09], Guard-Stage-Milestone (GSM) [Hul+11b], and CMMN as declarative process models.

Constraint-based modeling is another declarative approach that attempts to increase flexibility [Rus+14]. Constraint-based models describe the activities and the relationships among them using constraints to prohibit undesired behaviors [Rus+14]. DECLARE is considered to be a constraint-based approach [Rus+14].

Researchers have recognized that the distinction between imperative processes and declarative processes is not binary, and that a mixture of approaches is often preferable [de +15a; Pre+14]. Hybrid process models combine the characteristics of both imperative process models and declarative process models [Yu+15] to achieve a balance between the structure of imperative processes and the flexibility of declarative processes. Several hybrid process models and tools have been proposed, including the Case Analytics Workbench [Yu+15], BPMN-D [de +15a], CPN Tools 4 [WS13], CombiS-BP editor [Par+13], and the combination of DECLARE with BPMN [Her14].

From Control-flow to Data-driven

Imperative processes have been the traditional focus of BPM. According to Dumas et al. [Dum+16], the separation of concerns between imperative control-flow and data aspects in BPM has allowed for the creation of foundational theories, but has also limited the theory and methods of BPM that are now under pressure to support ad-hoc and more flexible processes.

This has led to other approaches to BPM that in addition to the declarative processes described above, are being explored. One of these approaches is to explore the continuum between imperative and declarative approaches and data-centric approaches, combining the *how* dimension with the *what* dimension. Although some researchers see this continuum as running from imperative to data-centric, Marrella et al. [Mar+15c] noticed that in both imperative and declarative approaches data can be relegated to input and output for activities. Therefore, it is more realistic to see this continuum from the *how* (focus on function) to the *what* (focus on data) dimensions.

Meyer et al. [Mey+11] describe a continuum from control-flow to data-driven, as follows: control-flow driven, data-aware control-flow driven, control-flow and data-driven, control-aware data-driven, and finally data-driven. Eshuis and van Gorp [Ev15] developed a hybrid process model that starts with an object-centric design that is later translated into GSM. They use UML AD for the activity-centric portion of the model, and Unified Modeling Language (UML) [OMG09c] state machines to describe the object's life cycle for the data-centric portion.

From Data-aware to Artifact-centric

As can be seen in Table 2.1 modeling approaches based on the *what* dimension [SZ92; Zac87] include data-centric, data-driven, artifact-centric, data-aware, object-aware, object-centric, and entity-centric modeling approaches. These modeling approaches can be placed on a continuum from data-aware to artifact-centric. Data-aware means that the process modeling language can accommodate simple data normally used for routing purposes. BPEL is an example of a data-aware approach [Hab+08]. Data-centric processes couple control-flow with data, treating the data as a first class citizen to make the process more flexible [Ev15]. Data-centric processes normally support the Knowledge Intensive Processes (KiP), which is based on a semi-structured approach that is difficult to achieve using activity-centric imperative models [Ev15]. Data-centric approaches use finite-state machines or rules for modeling techniques that describe the data life cycle [Ev15]. A data-centric process is based on the availability of data and its values to evolve the process [Mar+15c]. Similarly, the progress in object-aware processes is based on the processing of business data represented by business objects [Rei12].

Object-aware models treat business objects and business processes as equal entities and allow processes to manipulate business objects [Chi+14; Rei12]. Object-aware processes consider the behavior of individual objects and the interaction between object instances [Chi+14]. The execution of object-aware processes is guided by the availability of object instances and their values [Chi+14].

Object-centric models distribute the process among several interacting components, with each component representing the life cycle of an object [WK08]. In terms of this perspective, a BA is considered to be object-centric. GSM and Data-Centric Dynamic Systems (DCDS) follow the BA's framework [Cal+13; MC16; Rus+14], where the model allows for the definition of activities that rely on the data changes and states [Rus+14]. In both object-centric and artifact-centric approaches data and behavior are managed together in logical units [Dum+16]. Chiao et al. [Chi+14] use the terms "object-aware" and "object-centric" interchangeably.

From Ordinary Workers to Experts

As shown in Table 2.1, modeling approaches based on the *who* dimension includes user-driven, KiP, human-centric, people-centric, and collaborative approaches. Some of these modeling approaches can be placed on a continuum based on the type of user that is expected to interact with the executing process. The continuum may range from processes that ordinary workers can perform to complex processes that require experts [MF11]. Approaches like KiP and case management expect expert users that require flexibility and the ability to influence the outcome of the process.

Subject-oriented Business Process Management (S-BPM) [Fle+12] was introduced in 1994 by Fleischmann [Fle94]. S-BPM is based on a decentralized view of the process, where a process is an interaction between subjects [Fle+13]. Traditional BPM processes are based on a token traversing a control-flow, while S-BPM is based on subjects communicating using messages [Kan+16].

Other Dimensions

The *why*, the *where*, and the *when* dimensions presented in Table 2.1 give rise to other process model approaches that will be explored in this section.

Modeling approaches based on the *why* dimension [SZ92; Zac87] include the goal-oriented modeling approach. Goal-oriented processes focus on what has to be done in order to achieve a goal, and much less on how to perform the process [Küs+14]. Most of the work on goal-oriented processes is based on agents, or requirements. Some goal-oriented process models use a hierarchy of goals and are executed by agents, such as GO-BPMN [CG08] or Go4Flex [Bra+10]. Another approach to goal-oriented processes is based on requirements, and assumes that a business process encodes how the organization achieves its goals, therefore runtime changes to the process must be done in such a way as to preserve the original goals [AR16].

Modeling approaches based on the *where* dimension [SZ92; Zac87] include context-aware, and context-oriented modeling approaches. Some context-aware modeling approaches only consider location as the context of the process [Hei+15], but others allow for the modeling of other context events. CAptEvo is a context-aware framework where the context is modeled using a state machine in addition to the process [Buc+11; Buc+12]. Context-Aware Meta Execution-Workflow (C-MEW) is another context-aware workflow environment [Jai+16]. In both, CAptEvo and C-MEW the term “context” refers to more than just location. In C-MEW context information such as the user performing the task and the location is considered external context, but internal context such as task assignments to users and error exceptions can also be handled [Jai+16].

Modeling approaches based on the *when* dimension [SZ92; Zac87] include the event-driven modeling approach. The event-driven modeling approach uses event-action rules to describe a process [Kap+13] and is normally based on Complex Event Processing (CEP) [Fab+16; Sof12] [Lea09]. The benefit of CEP for process modeling has been widely recognized in the more than 130 papers that Krumeich et al. [Kru+14] found to be relevant in their Systematic Literature Review (SLR). EPC is an example of an event-driven process modeling approach. However, EPC only supports simple events, but has been extended to support CEP by Krumeich et al. [Kru+15].

2.2.3 Process Modeling Metrics

Extensive research on imperative, activity-centric business process modeling metrics has been conducted over the last 20 years. The SLR performed for this thesis (see Chapter 7) identified Daneva et al.’s 1996 study [Dan+96] as being one of the first published studies on business process modeling complexity metrics, and Kluza’s 2015 study [Klu15] as being one of the last.

Researchers have identified similarities between process models and programming languages [Ant+11; Car+06; GL06b; GL07; GD05; RV04; Van+07b; Van+08a], including object oriented languages [Khl+09]. This has informed two areas of research, which are discussed below.

First, the similarities between process models and programming languages have been used by some authors to adapt programming language metrics for use in process models. For example, Cardoso et al. [Car+06] adapted commonly used programming language metrics for use as metrics for business process models, including the number of lines of code for a program, the McCabe’s cyclomatic complexity [McC76], and the Halstead metrics [Hal87]. As a further example, Khlif et al. [Khl+09] adapted object oriented coupling metrics for use in process models.

Secondly, the fact that there are similarities between process models and programming languages has led some authors to classify process modeling metrics using software engineering classifications. Authors such as Cardoso [Car07a], Kluza and Nalepa [KN12], Reijers and Vanderfeesten [RV04], and Vanderfeesten et al. [Van+07a; Van+07b; Van+08a] have classified process modeling metrics using the five design principles proposed by Conte et al. [Con+86] and Troy and Zweben [TZ81], namely coupling, cohesion, complexity, modularity, and size. Others such as Khlif et al. [Khl+09] have used a subset of the five design principles and classified process modeling metrics into coupling, cohesion and complexity. Others such as Antonini et al. [Ant+11] have used Morasca's [Mor99; Mor08] classification of software attributes to classify process modeling metrics into coupling, cohesion, complexity, size, and length. The categories typically used to classify process modeling metrics include:

Coupling describes the number of interconnections between modules.

Cohesion describes the relationships of the elements within a module.

Complexity refers to structural complexity and describes the simplicity of the design.

Modularity describes how modular a design is.

Size describes the number of components.

Length describes how the components are organized.

Although this classification is useful, metrics for all of the categories have been used as process complexity metrics. In particular, coupling metrics have been used by Mendling [Men06] as process complexity metrics, and size metrics have been used by several researchers including García et al. [Gar+04a], Mendling [Men07], and Rolón et al. [Rol+05] as process complexity metrics. This led Vanderfeesten et al. [Van+08a] to question the classification of process model metrics into coupling, cohesion, complexity, modularity, and size, arguing that Cardoso et al. [Car+06], Gruhn and Laue [GL06b], and Latva-Koivisto [Lat01] use size as a complexity metric, and that Mendling [Men06] uses a coupling metric for the same purpose. As explained by Sánchez-González et al. [S+10a] there is a lack of consensus in the literature concerning the measuring concepts. Mendling [Men07] classifies the process model metrics into size, density, partitionability, connector interplay, cyclicity, and concurrency, where complexity is not considered a distinct class of metric.

The term "process complexity metric" has been defined by Cardoso [Car05d] as the degree to which a business process is difficult to analyze, understand or explain. Cardoso's definition was influenced by Zuse [Zus93] who defined the complexity of an object as being some measure of the mental effort required to understand that object. Gruhn and Laue [GL06a] define it as a measurement that can tell us whether a model is easy or difficult to comprehend.

Mendling [Men07] looked at the complexity of a process model as it is perceived by a human being doing the modeling. Both Cardoso's and Mendling's understandings are based on the IEEE's [IEE90] definition of complexity, which is defined as the degree to which a system or component has a design or implantation that is difficult to understand and verify. For the purpose of this thesis, complexity metrics include all those metrics that the researcher uses to measure complexity, and is independent from how these metrics may have been classified by other researchers.

In addition to complexity metrics for process models, it is also important to understand the complexity of the process modeling method being used to describe the process model. The popularity of process modeling has led to an increase in process modeling methods [Ind+09b]. Method complexity allows us to compare process modeling notations, which is important because it is expected to affect the learnability, ease of use, and overall use of a method [Rec+09; RB96]. The term "process modeling method" is synonymous with the term "process modeling language", therefore some process modeling methods include BPMN, UML AD, CMMN, etc. Some studies have concluded that the complexity of process modeling methods can have a negative impact on user perception and affect the usage of a particular method [Ind+09b].

2.3 Case Management

Some of the first references to the term *case management* within the context of IT and organizational processes were made by Berkley and Eccles [BE91] in 1991, and Davenport and Nohria [DN94] in 1994. In both instances, case management was used to empower workers (case managers) to work across functional areas. They focused on the role of the case manager, which was considered revolutionary because case managers broke the functional division of labor which was prevalent at the time. It was recognized that case management was useful in processes that dealt with customers both internal and external to the organization. The goal was to make the back room and the front room indistinguishable from a customer perspective. Case management was seen as a way for organizations with complex customer service processes to provide better service to their customers. From an IT perspective, the challenge was to provide adequate tools for the case managers. The two approaches, which the researchers envisioned, were first the creation of completely new integrated information systems, and second the creation of information systems with networked links to the production systems. Neither solution was adequate at the time. Projects focusing on the creation of new integrated systems were very large and risky, had a tendency to take a long time to implement and often failed to meet the objectives at the end. Projects that provided case workers with workstations that were connected to production systems were equally challenging because this perpetuated the existing issues with the production systems. Davenport and Nohria [DN94] concluded that the case manager role required innovative thinking about business processes, their relationship to customers, the role of information in a process, and the power of individual employees. They acknowledged that case management may not be relevant to all businesses and all processes, but that it has the potential to affect all of the customer facing organizations.

Case handling was first introduced in 2001 by van der Aalst and Berens [vB01] and van der Aalst et al. [van+05]. It uses the case as the central concept for the process. Activities are less rigid than workflow activities and a balance between data-centric and process-centric approaches is expected. The process is not just driven by the process flow, but also by the data. Although workers have more control of the process, they still need to be aware of the whole case. The ability to execute, redo, and skip activities is important in order to provide the required flexibility. In 2005, van der Aalst and Ter Hofstede [vT05] identified four central features of case handling:

1. Case handling avoids context tunneling (i.e., it provides case workers with all the information about the case instead of narrowing the information to the activity)
2. Case handling is data driven (i.e., it enables activities based on the available information instead of only using control-flow)
3. Case handling separates work distribution from authorization (i.e., query mechanisms can be used to navigate through active cases)
4. Case handling allows workers to view, add, and modify data outside an activity

Reijers et al. [Rei+03] described three characteristics associated with case handling systems. First, the system's focus is on the case; secondly, the process is data driven; thirdly, parts of the process model are implicit. In a traditional workflow, the designer specifies what is permitted (explicit modeling). Modeling in case handling is less prescriptive with only the preferred or normal path being modeled (implicit modeling). Case handling treats both data and process as first class citizens [vT05]. Case handling concepts were implemented in a set of products that included FLOWer of Pallas Athena, the Staffware Case Handler, and the COSA Activity Manager [vT05].

In 2006, Kaan et al. [Kaa+06] introduced *case management* as an alternative to *case handling*. The flexibility required by case handling impaired some of the advantages of the workflow technology [Kaa+06]. The authors saw case handling as an alternative to workflow [Kaa+06; Rei+03]. *Case management* as defined by Kaan et al. [Kaa+06] enhances workflow technology by focusing on the tasks. The control-flow between tasks is retained, but a task is decomposed into work content and activities. The work content provides the flexibility required by case management without compromising the control-flow provided by the workflow. This initial definition of *case management* is at odds with current definitions. However, it does help with clarifying the distinction between case handling and case management. With the exception of Berkley and Eccles [BE91] and Davenport and Nohria [DN94], definitions of case handling and case management are technology based and rely on particular tool implementations.

Towards the end of the 2000s, the term "case management" was developed further when market analysts started adding to the definition of the term. The definition changed from a product and implementation definition to a more general market definition. The term "case management" evolved into a method or practice that could be implemented in multiple ways by different products. Several market analysts, between 2007 and 2009, including Heiser and Lotto [HL07], Kerremans [Ker08], and White [Whi09] popularized the term *case management*. They emphasized the collaborative nature of case management and the flexible interaction between human beings, content, and processes. Kerremans [Ker08] defined case management work as collaborative and non-deterministic, where the work depended more on human decision making and content than on a predefined process.

In 2009, Le Clair et al. [Le +09] introduced the term *dynamic case management* and defined it as a highly structured but collaborative and dynamic information intensive process driven by events. The case folder contains all of the information needed to process and manage the case. This definition is consistent with other market analyst's definitions such as Heiser and Lotto [HL07], Kerremans [Ker08], and White [Whi09].

In 2010, Swenson [Swe10a] popularized the term *Adaptive Case Management (ACM)*. However, the terms case management and ACM were not clearly defined. Swenson's [Swe10a] book contains five different definitions for these terms. These include three different definitions of the term "case management" including the definition in the glossary [de +10; McC10; Swe10a], and two different definitions of the term "ACM" [Pal10; Swe10b]. Some authors [Bur11] consider dynamic case management and ACM to be synonymous, however, Pucher [Puc10] distinguishes between the two based on a particular interpretation of Le Clair et al.'s [Le +09] definition. Pucher [Puc10] understands dynamic case management as being dynamic at runtime, as opposed to ACM where the case is created just-in-time as needed. In addition, Pucher's [Puc10] view of ACM implies case adaptation based on previous instances. *Emerging case management* as defined by Böhringer [B1̈1] suggests a bottom-up view of case management that leverages social software techniques like micro-blogging, activity streams and tagging.

The work of Motahari-Nezhad and Swenson [MS13] and Swenson [Swe13] distinguishes between *ACM* and *Production Case Management (PCM)*. Both definitions comply with a generic definition of case management. The distinction is based on who creates the case template and when it is created. In ACM the case template is created by the knowledge worker at the moment that it is needed. In PCM the case template is created by developers during a design phase, and is then used by the knowledge workers. Both ACM and PCM allow knowledge workers a high degree of flexibility and discretion on how to complete the case. Using this categorization and Pucher's [Puc10] observations, dynamic case management may fall in the PCM camp.

This thesis uses the following definition of case management:

> "Case management is an IT for KiP using a case folder as a central repository, where the course of action for the fulfillment of *goals* is highly *uncertain* and the execution gradually *emerges* according to the available *knowledge base*." [Mar+15a]

2.4 Business Artifacts

The first paper to focus on BAs [CH09; Hul08] was published in 2003 by Nigam and Caswell [NC03]. The BA approach has also been called Business Entities with Lifecycle (BEL) [Hul+11b; Pol13]. A key aspect of the BA approach is the tied coupling of the data with the process represented by an information model that is described by a life cycle [Sol+13b]. This makes BAs a data-centric approach.

Figure 2.5 illustrates the evolution of BA modeling from the initial description of Nigam and Caswell's [NC03] Operations Specification (OpS) and Kumaran et al.'s [Kum+03] Adaptive Document (ADoc) in 2003 to CMMN in 2014. Each node in Figure 2.5 corresponds with a modeling notation, solid lines indicate direct bibliographic reference, and dashed lines indicate the flow of ideas. For simplicity, edges implied by transitivity have been removed from the graph. Each node points to the first article that referenced that work.

Nigam and Caswell's [NC03] OpS is purely data-flow driven, with no distinction being made between control-flow and data-flow. OpS tries to achieve a balance between being understood by business users and being a formal characterization useful for verifications. Nigam and Caswell's work was influenced by the Integrated Definition for Function Modeling (IDEF) [May+92], UML [OMG09c], and Zachman's framework for information systems architecture [Zac87]. In OpS the BAs provide the foundation for representing the business processes, and the model is created by specifying the information models and the life cycles models for a family of related BA types. The information model is defined as a set of attribute/value pairs. The life cycles are specified using a collection of interrelated tasks and repositories. Tasks perform actions against BAs, and repositories store the artifacts when they are not being acted upon. At any given time, an artifact is either in a single repository or it is being acted upon by a

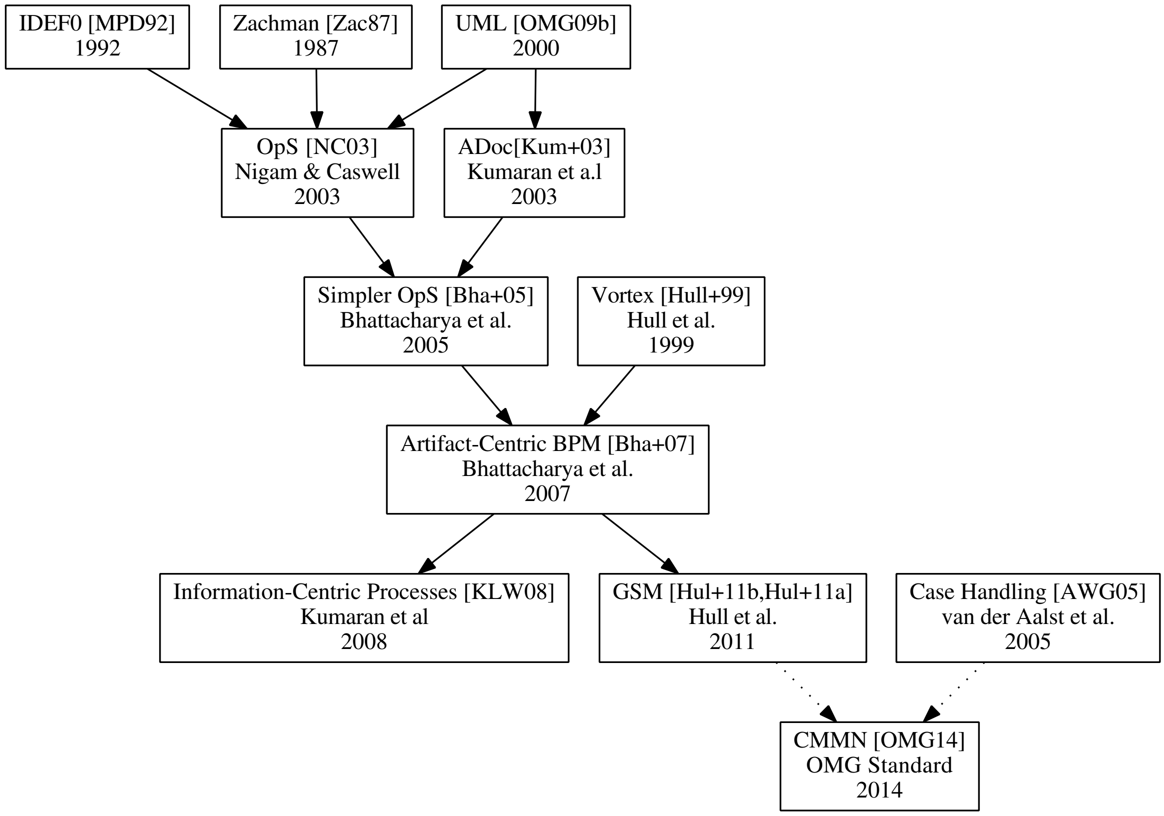


Figure 2.5: The evolution of business artifact modeling

single activity, which provides for transactional integrity, because only one activity can access the artifact at a time. Caswell and Nigam [CN05] formalized a mapping from OpS to an IT system. They introduced the concept of messages and a transport link that could carry an artifact or a message. This improved the model by describing how an OpS model could move from a requirement description to an executable environment.

Figure 2.6 shows the life cycle of a guest check artifact in a restaurant using the OpS notation. There are four activities performed against `guest checks`, shown as rounded rectangles. As the `guest check` moves through the business operations, if no activity is currently active it may be placed into the repositories `Active Guest Checks` or `Paid Guest Checks` (shown as circles). The other circles correspond with repositories used for related BAs, such as `Menu` or `Daily Specials`, whose associated activities are not shown.

In the same year that Nigam and Caswell [NC03] introduced BAs, Kumaran et al. [Kum+03] introduced the closely related *ADoc* model. An ADoc has three components: the dynamic data context, which corresponds with an information model; the event-driven behavior of the ADoc; and the collaborations, which enable groups of humans and/or automated agents to act on the ADoc. There is a loose relationship between the activities in BAs and the collaborations in ADoc. A single ADoc may have multiple collaborations active at the same time. Each collaboration may be specified in terms of one or more activities, and various

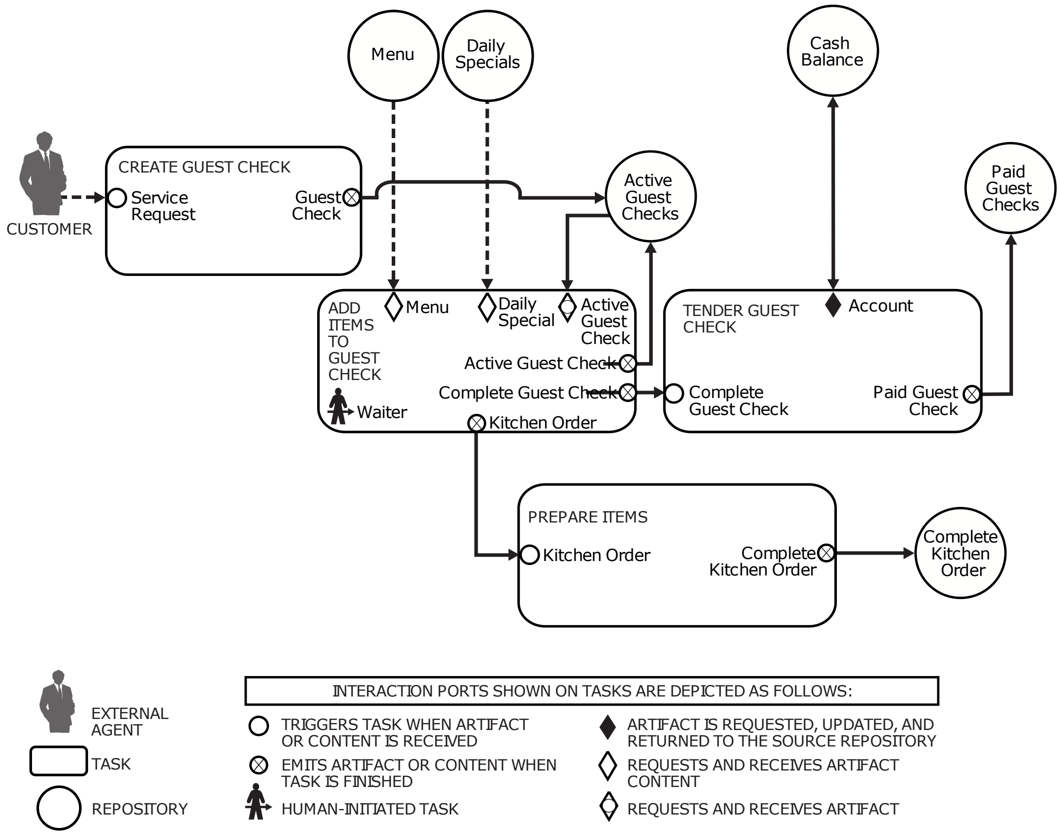


Figure 2.6: OpS representation of the life cycle of a guest check artifact in a restaurant [NC03]

models may be used to choreograph these activities. The activities in collaboration interact with the ADoc by sending events; these may in turn modify the dynamic data context by adding or modifying information. In the examples provided by Kumaran et al. [Kum+03] the dynamic data context includes an optional finite-state machine model that is used to record key states in the life cycle of the ADoc. The use of events between the collaborations and the dynamic data context provide more flexibility and parallelism than in OpS.

There are many similarities between OpS and Kumaran et al.'s [Kum+03] ADoc model although ADoc permits more modeling flexibility. Both models look at processes from the perspective of the information represented by the BA or the ADoc. Both make a clear distinction between BAs and business objects, where historically the latter were based on a more abstract object-oriented concept that did not explicitly model the life cycle aspect. There are also similarities between the BA and ADoc models and document management systems or document-driven processes where document events are used to drive the process [GM05; WK05].

Bhattacharya et al. [Bha+05] applied the artifact centered model to the pharmaceutical assay development process for drug discovery. They based their notation on a simplified version of the operational specification OpS [NC03]. Figure 2.7 shows the life cycle of an experiment record using the simplified OpS notation [Bha+05]. They focused on two primitives: tasks

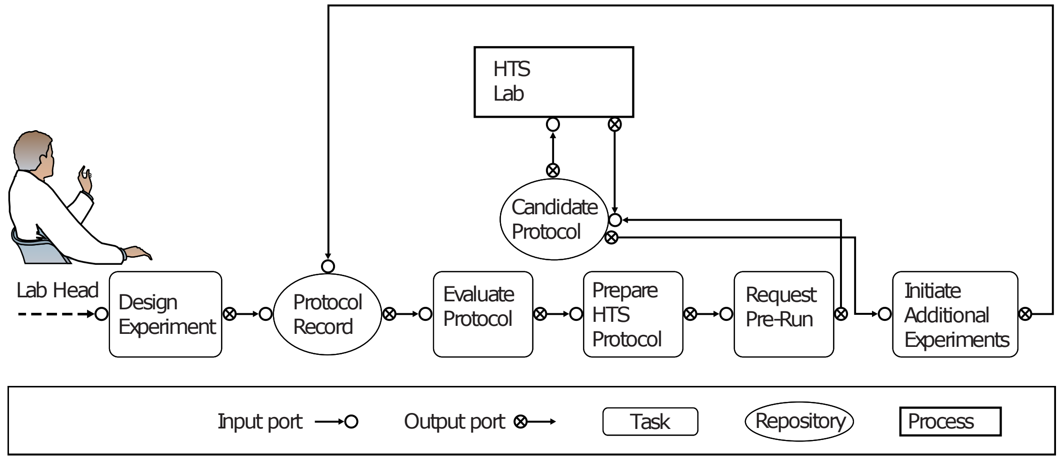


Figure 2.7: Life cycle of an experiment record [Bha+05]

and repository. Instead of using the artifact request operation as had been done in the original OpS, they introduced typed input and output ports with conditions that allowed the artifact to pass through the port. The main focus of Bhattacharya et al.'s work [Bha+05] was requirement gathering.

In 2007, Liu et al. [Liu+07] articulated a distinction between workflows and artifact models. They pointed out that traditional process models are *verb-centric* and so they describe the order in which activities are performed, while BAs are *noun-centric* and start by identifying the *things* that matter to the business [Liu+07]. Liu et al. based their work on the work done by Bhattacharya et al. [Bha+05], and provided a mapping between an OpS and colored Petri Net for formal analysis and verification.

Document engineering [GM05] has a strong relationship with the BA and ADoc approaches. For example, the document-driven workflows produced by Wang and Kumar [WK05] focus on the family of business documents that are used in a business process. They model the process using a flow with two kinds of nodes: document and activity. These nodes have connectives corresponding to *and*- and *or*- forks and joins. Figure 2.8 shows an order processing document flow using Wang and Kumar's [WK05] notation. In Wang and Kumar's work [WK05] each activity produces a distinct document; this contrasts with BA, which may be updated by multiple activities over a long period of time.

2.4.1 Finite-State Machine Based Lifecycles

Kumaran et al. [Kum+03] were the first to associate a state machine with an artifact life cycle. However, Kumaran et al. [Kum+08] present this association more explicitly by incorporating activities directly into the state machine as annotations on the state transitions between states. The activities can be thought of as the edges of the state machine that move the

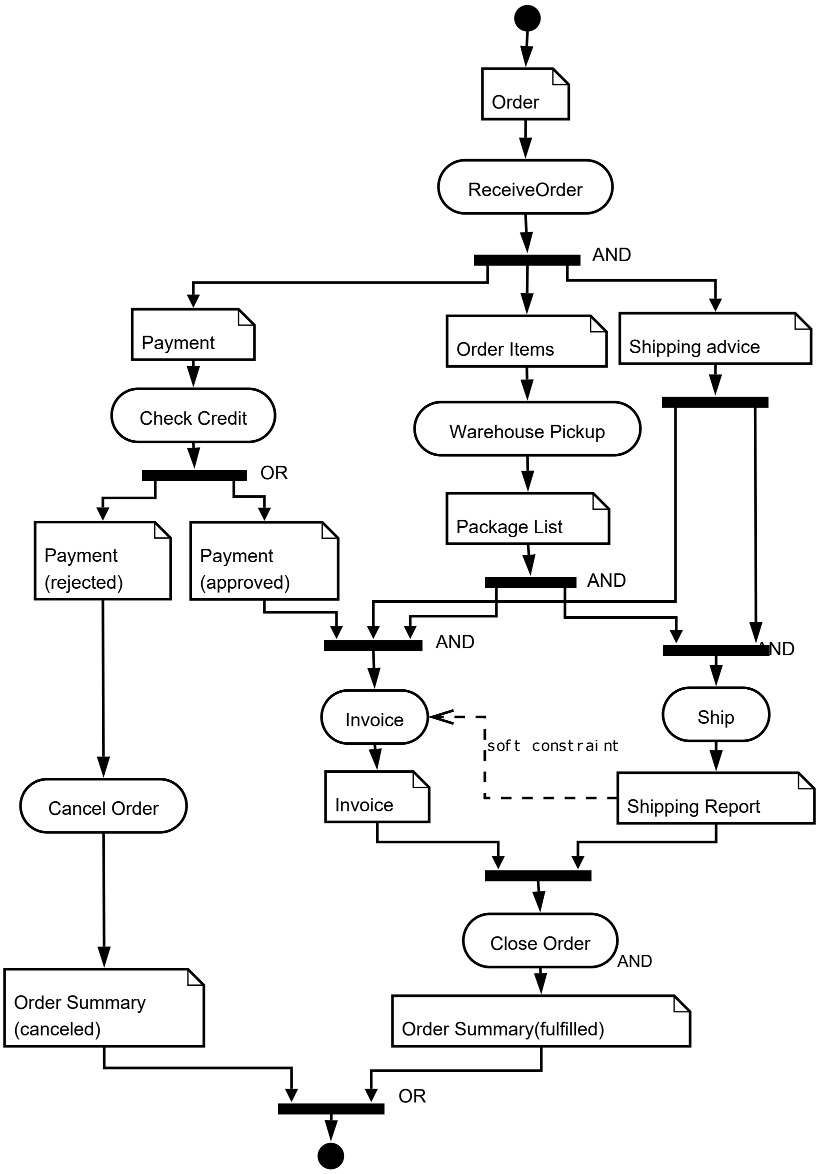


Figure 2.8: Order processing workflow [WK05]

business entity from one state to the next. This paper was also the first to use the term *business entity* in place of BA in order to avoid confusion regarding the use of the term *artifact* when working with business modeling practitioners. Figure 2.9 illustrates a process used for administering property damage claims using Kumaran et al.'s [Kum+08] notation. In contrast with OpS, where a single activity might access or modify two or more artifacts, in Kumaran et al.'s notation [Kum+08] the interaction between artifacts is limited to the activities of one artifact being able to send a message to a different artifact. While this makes artifact interaction a bit more cumbersome, it simplifies the description and explanation of each artifact's life cycle.

The explicit use of states in artifact life cycles provides a natural basis for adorning transitions with data-centric conditions and associated access controls based on both use roles and the

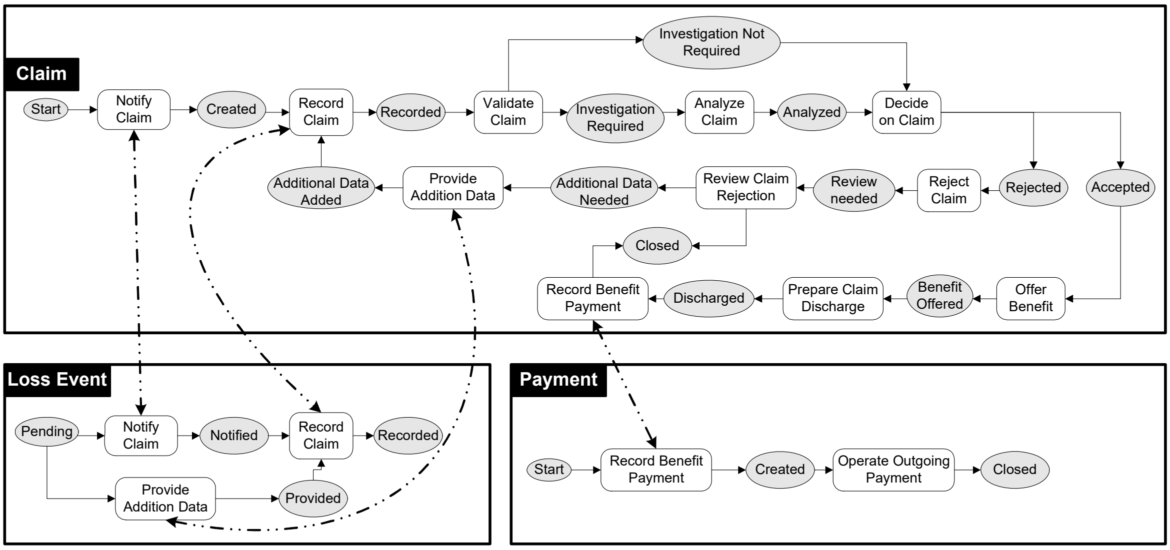


Figure 2.9: Administering property damage claims [Kum+08]

state that an artifact is in. The Model-Driven Business Transformation (MDBT) system [Kum+07] and Data4BPM [Nan+10], propose that an artifact be enabled to work side-by-side with a procedural BPM standard such as BPEL or BPMN.

The Business Entity Lifecycle Analytics (BELA) [Str+08] method introduces the concept of representing the life cycle by using milestones. The milestone represents a business-relevant operational objective that an artifact may achieve during its life. In BELA, these conditions focus mainly on which attributes have been assigned values. The use of milestones and achieving conditions has been extended further in the GSM approach.

The PHILharmonicFlows [KR11a; KR11b] project at the University of Ulm is another data-centric business process approach. Similar to BAs, PHILharmonicFlows is based on a strong integration of process and data. It provides support for business objects with finite-state machine based life cycles. Similar to such life cycles for artifacts, the states correspond to the achievement of key goals in the life of the object. Conditions on transitions and state-based access rights are supported. PHILharmonicFlows adopts a two-level framework; the micro processes represent data and the behavior of individual objects, while the macro processes consist of the activities that define the interactions between the objects [Rus+14]. This separation is reminiscent of the separation in ADoc between the data context and the collaborations described above. PHILharmonicFlows is considered to be a data-driven and object-aware approach [Chi+13; KR12; Rei12; RW12]. Chiao et al. [Chi+13] and Künzle and Reichert [KR12] define object-awareness as being a process that is structured and divided based on the object types it uses.

FlexConnect [Red+08; Red+09] is another framework related to finite-state machine based BAs and PHILharmonicFlows. This framework starts with business objects that are extended

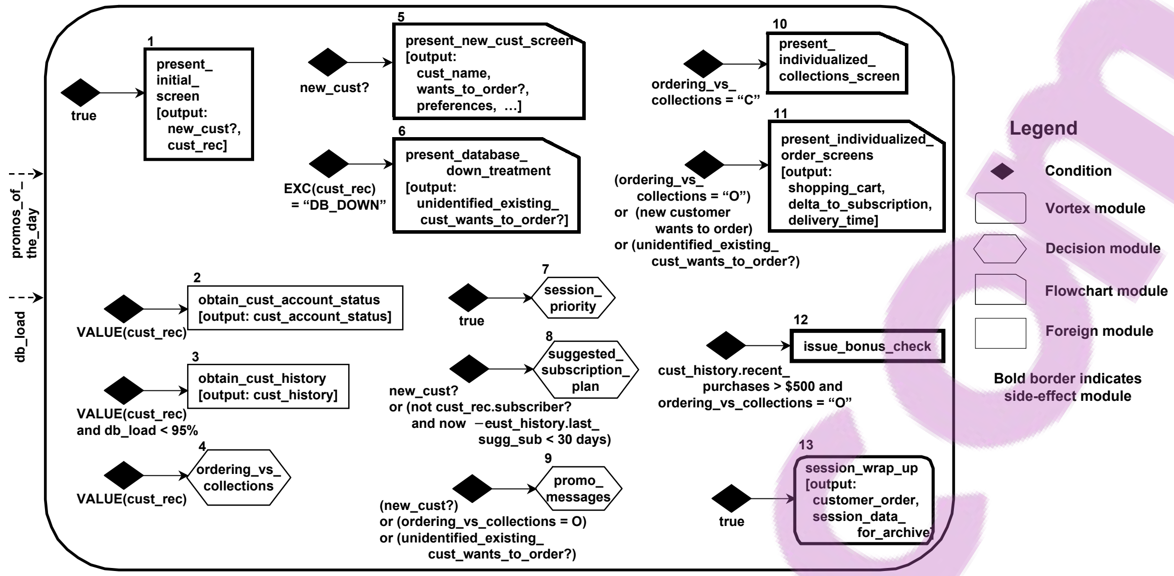


Figure 2.10: Process for a grocery store web-storefront [Hul+99]

to support finite-state machine life cycles, including messaging between objects. In contrast to the artifact approach, states in FlexConnect represent activities, and the transition from one state to the next corresponds implicitly with the achievement of a business goal.

2.4.2 Declarative Life Cycles

While the BA and object-aware frameworks described above embrace the data and the life cycle of data-centric objects, they are for the most part based on procedural models of process. The next step required by modern case management is the switch from procedural to declarative life cycles. Several proposals have appeared that use a declarative approach with data-centric models.

Some consider the first published work on declarative data-centric process frameworks to have been done on Vortex [Hul+99]. This model, introduced in 1999, supports highly flexible workflows. A key objective of Vortex was to enable business-level users to understand and modify workflow models. Similar to the artifact approach, a Vortex workflow is based on an object characterized by attribute/value pairs. Unlike artifacts, Vortex focuses on one object instance at a time. Figure 2.10 shows a Vortex module of a Vortex workflow object for managing web store-front interactions with a web storefront. Four kinds of activity modules are supported within this outer module: decision, flowchart, foreign (i.e., external), and Vortex. The Vortex modules enable a hierarchical structure within the process model. The activation of sub-modules in a Vortex module is governed by condition-based guards that refer primarily to the current state (i.e., data values of the underlying object). In Vortex, a directed graph is formed by adding an edge from one module to another if the second module or one of its guards reads an attribute that is written by the first. A significant limitation in Vortex is that the graph is required to be acyclic.

As previously discussed (see Section 2.3), van der Aalst and Berens [vB01] and van der Aalst et al. [van+05] wrote the first article that described a declarative case handling approach. The case handling approach presented was quite rich and provided for a collection of object definitions (basically a case folder) and a collection of activity definitions. Activities have a finite-state machine based life cycle, and events are used to move activities between the states. To formally specify the behavior of the activities a set of Event-Condition-Action (ECA) rules is used. The use of ECA rules allow for the creation of very flexible case models.

Bhattacharya et al. [Bha+07] were the first to publish on BAs that incorporated rules-based life cycles. This approach was called the *artifact-centric BPM* and was fairly abstract. In Bhattacharya et al.'s work [Bha+07] an artifact includes an information model, a family of activities, a state machine that characterizes the artifact's life cycle in coarse terms, and a set of condition-action rules whose actions can be used to execute an activity or to move from one state to another. In this first declarative artifact-based model, the activity execution is assumed to be sequential (i.e., two activities cannot run in parallel). Figure 2.11 shows a service provider's process model annotated with artifacts.

The GSM approach for specifying declarative life cycles was introduced by Hull et al. [Hul+11b] and further refined by Damaggio et al. [Dam+11] and Hull et al. [Hul+11a]. From a modeling perspective, GSM switches the focus from modeling the BA state machine (as discussed in Section 2.4.1) to a declarative GSM that resembles modeling in Vortex [Hul+99], but without Vortex's acyclicity limitation. The operational semantics for GSM were inspired by van der Aalst et al.'s [van+05] case handling approach, but are not restricted by the acyclicity condition imposed in the case handling approach. GSM introduces the concept of *sentries* that correspond to the application-specific ECA rules of the case handling model. The operational semantics of GSM are based on the notion of snapshot and Business step (B-step). A B-step corresponds intuitively to enabling the system to respond to a single incoming event, including the firing of all applicable ECA rules until stability is achieved. de Masellis et al. [de +15b] describe GSM types as business entities in a domain characterized by a data schema (information model), a life cycle, and a set of instances. Figure 3.3 presents an artifact type for the deal artifact [Esh+13].

DCDS is a new framework for data-centric processes that was introduced by Bagheri Hariri et al. [Bag+13]. DCDS is composed of two layers: the relational data layer holds the relevant information being processed and a process layer of atomic actions based on a declarative rule-based specification [Mar+15c]. The semantics of DCDS are based on snapshots [Cal+15] similar to GSM snapshots. The process layer is specified by actions, which are the basic building blocks and are activated using condition-action rules [Rus+13].

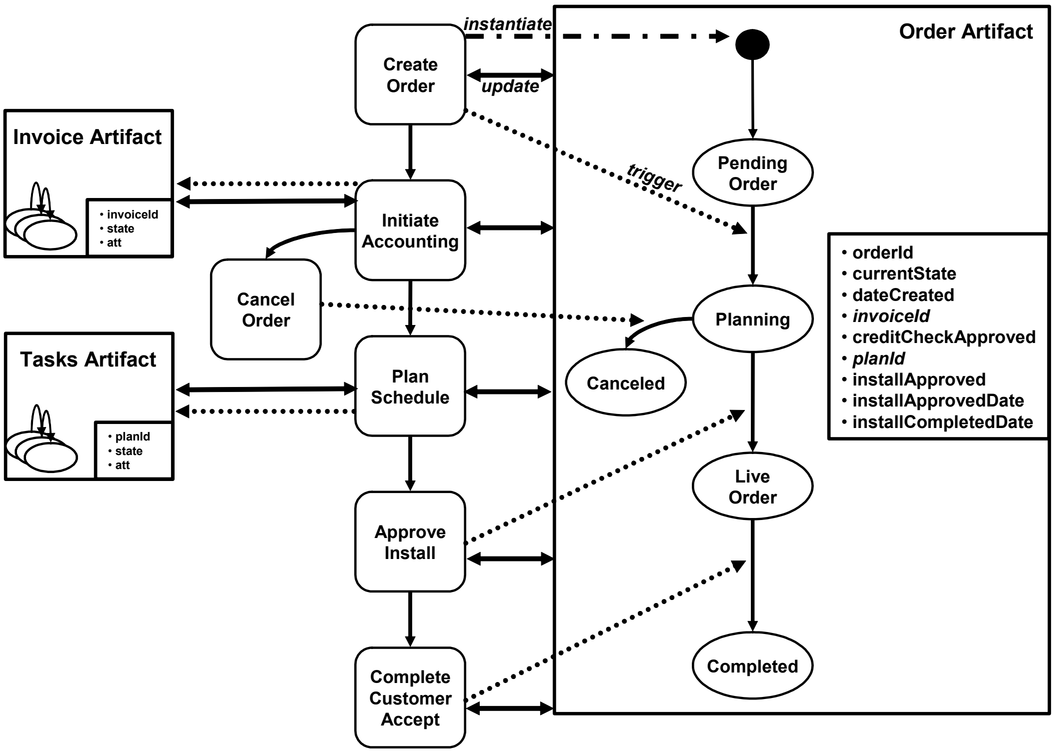


Figure 2.11: Service provider's process model annotated with artifacts [Bha+07]

2.4.3 The Case Management Model and Notation

In 2014, the Object Management Group (OMG) [Gro17] published CMMN [OMG14a] as the result of a 2009 request for a proposal for a Case Management Process Modeling (CMPM) [OMG09b] notation. As with any BA approach, there is a clear separation in CMMN between the case information model, based on a case file and case file elements, and the case behavioral model (life cycle). Furthermore, a case model may contain multiple case types, which interact via message passing.

The GSM approach [Dam+11; Hul+11a; Hul+11b] has had a substantial influence on CMMN [Mar+13]. In particular, the behavioral model for a case is composed of tasks, hierarchical stages, events, and milestones, which are variations of those concepts present in GSM. A significant deviation from GSM is the handling of milestones. In GSM milestones can be invalidated, whereas in CMMN they can only be achieved. As is the case with GSM, dependencies between tasks, stages, and milestones are described using sentries. Other differences between CMMN and GSM are described further in Section 3.4.

CMMN includes finite-state machine based life cycles for tasks and stages, similar to the life cycles of activities provided in van der Aalst et al. [van+05]. An important aspect of CMMN, which concerns the ability for case workers to alter the runtime plan, comes from the Cordys product [de 09]. The ability to alter the model at an instance level while it is executing at runtime was not present in any of the BA models. Additionally, CMMN allows for the modeling of other runtime behavior not present in BA approaches, such as the ability for case workers to disable activities.

de Carvalho et al. [de +16] have compared the flexibility of the process execution in CMMN to that of ConDec [Pv06]. Both languages have different goals, CMMN is a planning-by-doing language because case workers can affect the runtime process by adding discretionary items, while in ConDec execution is guided by constraints [de +16]. Additionally, ConDec has formal semantics while CMMN does not [de +16]. de Carvalho et al. [de +16] concluded that both languages can be extended to support more flexible processes.

2.5 Summary

This chapter provided an introduction and the basic terminology associated with BPM, process modeling, case management, BA, and CMMN. The terms introduced in this chapter are commonly used within the body of relevant literature on process management.

The material reviewed in this chapter provides background information for Chapters 3 and 6 and Section 3.3. Some material from this chapter was previously published in Marin et al. [Mar+13; Mar+14b; Mar+15a].

Chapter 3

Business Artifacts and Case Management

This chapter describes and compares two approaches to implementing BAs: the Guard-Stage-Milestone (GSM) [Hul+11b] approach and the Case Management Model and Notation (CMMN) [OMG14a] approach. The chapter contributes to the Business Artifact (BA) literature by providing formal definitions for a CMMN case type (see Definition 3.4), case program (see Definition 3.9), and case model (see Definition 3.10) using first-order logic. The formalizations of GSM and CMMN presented here will be used in Chapters 5 and 7 to reason about the notation and to validate the CMMN complexity metrics. This chapter also contributes to the BA literature by providing a clear and detailed comparison between GSM and CMMN. Material from this chapter has been published in Marin et al. [Mar+15b; Mar+16].

This chapter is organized as follows. Section 3.1 describes a framework that evaluates BA approaches. Section 3.2 describes and formalizes GSM and provides background material for Chapter 5. Section 3.3 describes and formalizes CMMN and provides background material for Chapters 5 and 7. Finally, Section 3.4 compares GSM to CMMN.

3.1 Business Artifacts with Lifecycle Services and Associations

As discussed in Section 2.4 (Chapter 2), there exist multiple approaches to BA, including Nigam and Caswell's [NC03] Operations Specification (OpS), and Kumaran et al.'s [Kum+03] Adaptive Document (ADoc) model. Some researchers have tried to merge a data-centric approach with a more traditional activity-centric approach [God15], others have pursued a purely data centric path such as can be found in Data-Centric Dynamic Systems (DCDS)

[Bag+13; Cal+15] or GSM [Dam+11; Hul+11a; Hul+11b]. The diversity of BA approaches led Bhattacharya et al. [Bha+09] and Hull [Hul08] to propose a framework that would categorize them.

BA processes are characterized by four dimensions [Bha+09; Hul08]. These dimensions describe a set of data centric business process approaches that can be categorized as BAs. The four dimensions are the information model, the macro-level life cycle, the services (tasks), and the associations or constraints that tie them together [Bha+09]. These four dimensions are outlined below:

Business Artifacts (Information Model). This describes the business relevant data encapsulated in BAs that is required in order to execute the process. Similar to an object, the *information model* of a BA has an identity and contains the status of the BA. The business relevant data can be described in several ways, including a database schema, name-value pairs [NC03], or ADocs [Kum+03].

Lifecycle. The macro-level *life cycle* describes how the artifact evolves and how the artifact's data is manipulated as a result of the evolution of the process. The *life cycle* may be described using flow charts, finite-state machines, state charts or by using declarative mechanisms [Kun+15].

Services (Tasks.) The *services* or *tasks* encapsulate the unit of work in the process. A *service* or *task* may change the state or values of one or more BAs.

Associations. *Associations* describe the rules and constraints that govern the BA. These can be specified procedurally or declaratively [Kun+15].

These four dimensions comprise the Business Artifacts with Lifecycle Services and Associations (BALSA) framework created by Bhattacharya et al. [Bha+09] and Hull [Hul08].

3.2 Guard-Stage-Milestone

The GSM approach was introduced by Hull et al. [Hul+11b], and was further refined by Damaggio et al. [Dam+11] and Hull et al. [Hul+11a]. GSM Artifact types can be described as business entities functioning in a domain that is characterized by a data schema (information model), a life cycle, and a set of instances [de +15b]. GSM is one of many approaches to the artifact centric business process paradigm [Kun+15].

GSM can be characterized using the four BALSA dimensions outlined below:

Information Model. The *information model* in GSM corresponds with the descriptions of business relevant information contained in the BA. Each BA has an identity that is represented by attributes in a database schema. The attributes are separated into two sets, namely data attributes and status attributes. Data attributes describe information about the BAs, and status attributes describe the status of the stages and milestones. Some of the examples of BAs provided by Hull [Hul08] include purchase orders, sales invoices, bills of lading, and insurance claims.

Lifecycle. The *life cycle* is modeled using guards, stages, and milestones (GSM model). A guard describes the condition that allows the stage to open and become active, representing the state of the BA. A milestone describes the condition that will close an open stage. In addition to milestones Sun et al. [Sun+12] introduced terminators to close open stages. Stages can have multiple guards and milestones. The inside of a stage may contain multiple sub-stages or a single task.

Services (Tasks). *Services* correspond to *tasks* that invoke Web Services. In this model *tasks* are contained in atomic stages, sharing the guards and milestones with the stage. *Services* are activated using events [Kun+15].

Associations. The *association* between the BA, its life cycle, services and external conditions is specified by using Event-Condition-Action (ECA) rules. The ECA rules are described as sentries, using the form

on <event> **if** <condition> **then** <action>

where the action applies to the stage, or milestone associated with the sentry.

3.2.1 Artifact Type

Several definitions of the GSM Artifact type, Schema or Model appear in the literature, because it is continuously being refined. The first formal GSM Schema was presented by Hull et al. [Hul+11b], and then refined by Damaggio et al. [Dam+11] and Hull et al. [Hul+11a]. Further refinements were made by Eshuis et al. [Esh+13] and Sun et al. [Sun+12] formalized a variation of the GSM Schema.

This section makes use of descriptions and definitions from Sun et al. [Sun+12], Eshuis et al. [Esh+13], Belardinelli et al. [Bel+12] and Damaggio et al. [Dam+11].

Definition 3.1. (GSM Artifact type): A GSM Artifact type Γ is a collection of attributes, event types, stages, tasks, milestones, sentries, and a life cycle. It is described by the following tuple

$$\Gamma = \langle Att, EType, Stg, Tsk, Mst, Stry, Lcyc \rangle$$

where,

Att is a set of *attributes*. *Att* is partitioned into *data attributes* Att_{data} and *status attributes* Att_{status}.

The *data attribute* sets Att_{data} store all of the data that is relevant to the business process. Elements in Att_{data} have types, including scalars, records of scalars, and sets of scalars.

The *status attribute* set Att_{status} is partitioned into *stage status attributes* Att_{stages} and *milestone status attributes* $Att_{milestones}$. Both Att_{stages} and $Att_{milestones}$ are sets of Boolean values representing the status of stages and milestones (always initialized to *false*).

$$\begin{aligned} Att &= Att_{data} \cup Att_{status} \\ &= Att_{data} \cup (Att_{stages} \cup Att_{milestones}) \end{aligned}$$

Since Att_{data}, Att_{stages}, and $Att_{milestones}$are pairwise disjoint, we have

$$\begin{aligned} Att_{data} \cap Att_{status} &= \varnothing \\ Att_{data} \cap Att_{stages} &= \varnothing \\ Att_{data} \cap Att_{milestones} &= \varnothing \\ Att_{stages} \cap Att_{milestones} &= \varnothing \end{aligned}$$

EType is a set of *event types*. *EType* is partitioned into *incoming events* $EType_{inc}$ and *generated events* $EType_{gen}$. An *event type* $e \in EType$ has the form $e(a_1, \ldots, a_n)$, where e is the name of the *event type*, and $\{a_1, \ldots, a_n\} \subseteq Att_{data}$ is called the *payload* of e. *Incoming events* $EType_{inc}$, update $Att_{data}$with the payload. *Generated events* $EType_{gen}$use $Att_{data}$to populate the payload.

$$EType = EType_{inc} \cup EType_{gen}$$

Since $EType_{inc}$ and $EType_{gen}$ are partitions of *EType*, we have

$$EType_{inc} \cap EType_{gen} = \varnothing$$

Stg is a set of *stages*. For each $S \in Stg$, there is a Boolean status attribute $S \in Att_{stages}$ ($Att_{stages} \subseteq Att_{status}$). The value of S is *true* if the stage is open, but otherwise it is *false*.

Tsk is a set of *tasks*. *Tasks* are the real activities conducting work in the GSM Artifact type, which are implemented by external services. *Tasks* are invoked via messages. There are three types of tasks:

- 1-way message;
- 2-way service invocation; and
- the creation of a new GSM instance.

Mst is a set of *milestones*. For each $m \in Mst$, there is a Boolean status attribute $m \in Att_{milestones}$ ($Att_{milestones} \subseteq Att_{status}$). The value of m is *true* if the milestone has been achieved and not invalidated, but otherwise it is *false*.

Stry is a set of *sentries*. See Definition 3.3 for details.

Lcyc is a *life cycle model*. See Definition 3.2 for details.

Sets *Att*, *EType*, *Stg*, *Tsk*, and *Mst* are pairwise disjoint. Intuitively, a GSM Artifact type is a record of structured data [Bel+12] with a life cycle model that describes how such data evolves over time.

Definition 3.2. (Lifecycle): The *life cycle model Lcyc* for a GSM Artifact type Γ is defined by the following tuple

$$Lcyc = \langle Substages, Tasks, Submilestones, Guards, Terminators, Ach, Inv\rangle$$

where,

Substages is a function from *Stg* to a finite subset of *Stg*. $Substages : Stg \rightarrow Stg$, is defined by the relationship ($Substages^R \subseteq Stg \times Stg$).

$$Substages^R = \{\langle S, S'\rangle \mid \ S \in Stg \ \wedge \ S' \in Stg \ \wedge \ S' \text{ is contained in } S\}$$

For $\langle S, S'\rangle \in Substages^R$, we say that $S' \in Substages(S)$. The relationship creates a forest, with roots called *top-level stages*, and the leaves called *atomic stages*. Non-leaf nodes are called *composite stages*. If $S \in Substages(S')$, then S is a *child-stage* of S' and S' is the *parent-stage* of S.

Tasks is a function from the atomic stages in *Stg* to *Tsk*. $Tasks : Stg \rightarrow Tsk$, is defined by the relationship ($Tasks^R \subseteq Stg \ \times \ Tsk$).

$$Tasks^R = \{\langle S, t\rangle \mid S \in Stg \ \wedge \ t \in Tsk \ \wedge \ \nexists S' : \langle S, S'\rangle \in Substages^R \ \wedge \ t \text{ implements } S\}$$

Tasks can be reused.

Submilestones define a function from *Stg* to a non-empty, finite subset of milestones *Mst*. $Submilestones : Stg \rightarrow Mst$, is defined by the relationship ($Submilestones^R \subseteq$

$Stg \times Mst$).

$$Submilestones^R = \{\langle S, m\rangle \mid S \in Stg \ \wedge \ m \in Mst \ \wedge \ m \text{ is contained in } S\}$$

If $m \in Submilestones(S)$, then m is a *child-milestone* of S and S is the *parent-stage* of m. Milestones cannot be shared between stages. Therefore,

$Submilestones(S) \ \cap \ Submilestones(S') = \varnothing$ for $S \neq S'$.

Guards is a function from Stg to a finite, non-empty set of $Stry$. $Guards : Stg \rightarrow Stry$, is defined by the relationship ($Guards^R \subseteq Stg \times Stry$).

$$Guards^R = \{\langle S, g\rangle \mid \ S \in Stg \ \wedge \ g \in Stry \ \wedge \ g \text{ opens } S\}$$

Guards describe when a stage $S \in Stg$ opens (i.e., start execution).

Terminators is a function from Stg to a finite, non-empty set in $Stry$. $Terminators : Stg \rightarrow Stry$, is defined by the relationship ($Terminators^R \subseteq Stg \times Stry$).

$$Terminators^R = \{\langle S, x\rangle \mid S \in Stg \ \wedge \ x \in Stry \ \wedge \ x \text{ closes } S\}$$

Terminators describe when a stage $S \in Stg$ closes (i.e., completes execution). *Terminators* were not included in Hull et al.'s original definition of GSM [Hul+11a]. However, these were introduced at a later time by Sun et al. [Sun+12].

Ach is a function from milestones Mst to a finite, non-empty set in $Stry$ called the *achieving sentry* of m ($m \in Mst$). $Ach : Mst \rightarrow Stry$, is defined by the relationship ($Ach^R \subseteq Mst \times Stry$).

$$Ach^R = \{\langle m, a\rangle \mid m \in Mst \ \wedge \ a \in Stry \ \wedge \ a \text{ achieves } m\}$$

Ach describes when a milestone $m \in Mst$ is achieved (i.e., its Boolean state $m \in Att_{milestones}$ becomes *true*).

Inv is a function from Mst to a finite set of $Stry$ called *invalidating sentry* of m ($m \in Mst$). $Inv : Mst \rightarrow Stry$, is defined by the relationship ($Inv^R \subseteq Mst \times Stry$).

$$Inv^R = \{\langle m, i\rangle \mid m \in Mst \ \wedge \ i \in Stry \ \wedge \ i \text{ invalidates } m\}$$

Inv describes when a milestone $m \in Mst$ has been invalidated (i.e., its Boolean state $m \in Att_{milestones}$ becomes *false*).

Definition 3.3. (GSM Sentry): A *sentry expression* x in $Stry$ for a GSM Artifact type Γ consists of two optional components, an *event expression* and a *condition.* A *sentry expression*

x is defined as follows:

$$\chi = \begin{cases} [\textbf{on}\ \xi\ \textbf{if}\ \varphi] & \text{event expression } \xi \text{ and condition } \varphi \\ [\textbf{on}\ \xi] & \text{event expression } \xi \text{ only} \\ [\ \textbf{if}\ \varphi] & \text{condition } \varphi \text{ only} \end{cases}$$

where, ξ is an *event expression* comprising one event; and φ is a well-formed Boolean expression that may use attributes from Att. We use square brackets ([ ]) to enclose *sentry expressions*.

The optional *event expression* ξ has one of the following forms:

Incoming event $e \in EType_{inc}$:

E:e when an external event e is received.

Generated event $e \in EType_{gen}$ includes:

$+m$ when milestone m is triggered (Boolean attribute $m \in Att_{milestones}$ transitioned from *false* to *true*).

$-m$ when milestone m has been invalidated (Boolean attribute $m \in Att_{milestones}$ transitioned from *true* to *false*).

$+S$ when stage S opens (Boolean attribute $S \in Att_{stages}$ transitioned from *false* to *true*).

$-S$ when stage S closes (Boolean attribute $S \in Att_{stages}$ transitioned from *true* to *false*).

I:t when task t starts.

C:t when task t completes.

The optional condition φ can access all the attributes in Att, including the state of stages ($S \in Att_{stages}$) and milestones ($m \in Att_{milestones}$). The condition φ is specified using an extended Object Constraint Language (OCL) [OMG14b].

Figure 3.1 illustrates a GSM type using the notation introduced in the original definition of GSM by Hull et al. [Hul+11a], with terminators being added from Sun et al. [Sun+12]. The diagram shows three stages ($S1$ to $S3$), two tasks ($T1$ and $T2$), three guards ($g0$ to $g2$), three milestones ($m1, mt2, mt3$), three terminators ($t1, mt2, mt3$), and an event propagation between $mt2$ and $g1$. Note that $mt2$ and $mt3$ represent both a milestone and a terminator, while $m1$ is only a milestone and $t1$ is only a terminator. The GSM data attributes are also shown in the diagram. In this example, there is an *id* and three other data attributes, as well as the status attributes for the stages and milestones.

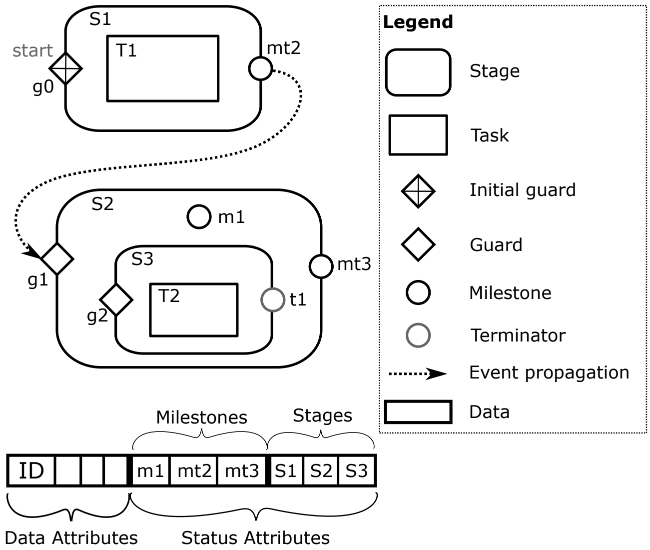


Figure 3.1: Example of the GSM notation

3.2.1.1 Operational Semantics

The operational semantics of GSM are based on two notions that are discussed below [Dam+11; de +15b; Hul+11a]:

Snapshot which describes the state of any GSM instance at a point in time. Informally, the *snapshot* of a GSM instance is represented by Att.

Business step (B-step) represents the impact of a single incoming event $E : e \in EType_{inc}$ on a snapshot. During the process of a B-step, internal events which are represented by $e \in EType_{gen}$ may be generated and processed until no further internal events can be processed. The semantics of a B-step follow a set of Prerequisite-Antecedent-Consequent (PAC) rules designed to preserve the consistency of the GSM Artifact type. A B-step is considered an atomic operation, and the result of a B-step is a new snapshot. External events that may arrive during the processing of a B-step are queued and processes after the current B-step is completed. Each incoming external event generates a new B-step.

A side effect of the PAC rules is the *toggle once principle*. The *toggle once principle* guarantees that each status attribute, in Att_{status}, changes value only once during a B-step [Sol+13b]. This is done in order to prevent infinite cycles, and to make each status change visible at the end of the B-step [Dam+11] in the resulting snapshot. The end result is that changes from snapshot to snapshot are business relevant [Sun+12]. Figure 3.2 shows an invalid GSM model with a problematic stage that contains a terminator that triggers a guard in the same stage, which violates the toggle once principle.

3.2.2 Example

This section makes use of an example from Eshuis et al. [Esh+13] to illustrate the GSM notation and definitions. The example originates from the description of the IBM Global Financing (IGF) artifact created by Chao et al. [Cha+09]. IGF is the largest Information Technology (IT) financing organization in the world. It finances hardware, software, and services to customers in the IT industry. The example focuses on a simplified version of the deal artifact and, in particular, describes the `Refine Deal` component of the process.

Figure 3.3 shows the artifact type for the simplified deal artifact. The original artifact type presented by Eshuis et al. [Esh+13] only included stages, with a `Check Credit` stage being described as an atomic stage encapsulating a single task. To cover the complete GSM approach the `Credit Check` stage was replaced by the internal `Check Credit` task. This was the only modification made to the original example. The bottom of Figure 3.3 describes the information model that consists of data attributes and status attributes. The data attributes correspond to the BA (in this case the deal artifact), and can take any form as long as the data is supported by the database. The status attributes comprise a set of Booleans that describe the state of the different milestones and stages.

Table 3.1 describes the rules that open and close the stages and tasks in Figure 3.3. A rule is composed of a sentry and an internal action. The sentry is an expression that may be triggered by an event and/or a Boolean condition. The internal action can open (start) or close (terminate) a stage or task. Table 3.2 describes the rules for the milestones presented in Figure 3.3. The internal action for milestones concerns achieving (becoming *true*) or invalidating (becoming *false*) the milestone. The event expression ξ of the rule can use internal events generated by the GSM system, such as a state attribute changing state from *false* to *true* (indicated with a + sign in front of the corresponding milestone or stage), or from *true* to *false* (indicated with a – sign in front of the corresponding milestone or stage), or tasks completed indicted with a ‘C:’ in front of the corresponding task. For example rule 17 (see Table 3.2) of milestone `Credit Checked` is triggered when the generated event `C:Check Credit` happens. This event is a completed event from the task `Check Credit`. The event can also be an external event indicated by an ‘E:’ in front of the name of the external event.

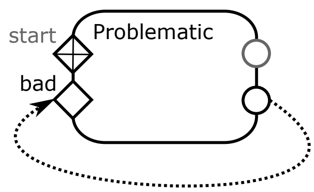


Figure 3.2: Violation of GSM toggle once principle

For example rule 24 (see Table 3.2) of milestone `ReDraftTermNeeded` is triggered when the external event `E:Regulation Change` happens.

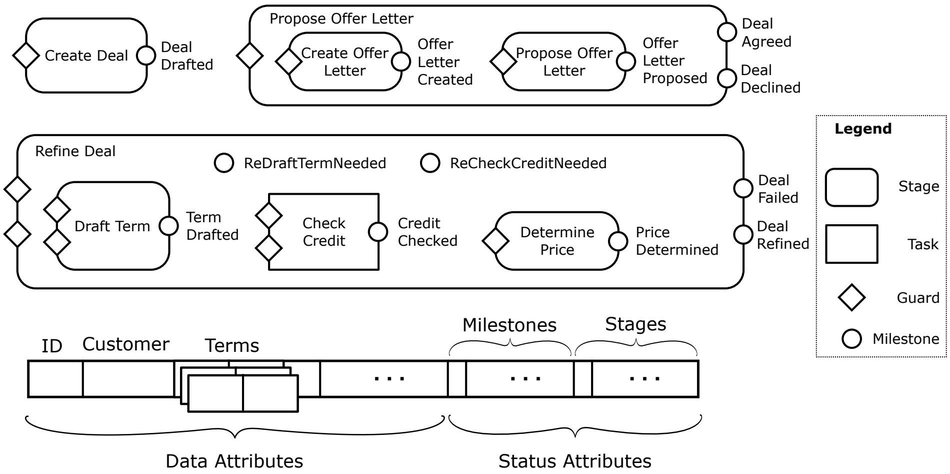


Figure 3.3: Artifact type for the deal artifact (from [Esh+13]) Example of a typical artifact type.

3.3 Case Management Model and Notation

CMMN [OMG14a] is a standard specification that was published in 2014 by the Object Management Group (OMG) [Gro17]. It is intended to complement other OMG specifications in the area of business processes. The goal of CMMN is to address case management as described by Reijers et al. [Rei+03], Swenson [Swe10a], van der Aalst and Berens [vB01], and van der Aalst et al. [van+05] using a data-centric approach based on BAs [Mar+13]. The development of CMMN was influenced by GSM, as indicated by the inclusion of two GSM papers [Hul+11a; Hul+11b] that were listed as non-normative references in the CMMN specification.

The CMMN relationship to BAs can be described using the four BALSA dimensions.

BA (Information Model). The *information model* in CMMN is based on the concept of a case file (also called a case folder or a case). The case file holds all of the information that is relevant to the case. The information in the case file can be structured (scalars, records of scalars, sets of records, etc.) or unstructured (documents, pictures, video, etc.) [MB15; OMG14a]. The case file allows for containment and reference relationships, which support different types of containers (sets, folders, directories, stacks, etc.).

Lifecycle. The *life cycle* of a case is modeled using an approach similar to that of GSM. Entry criteria similar to guards in GSM describe the condition to start a task or stage. Exit criteria describe the conditions to terminate a task or stage. The evolution of a case throughout the different stages and tasks describes the life cycle of the case.

Table 3.1: GSM sentries for the stages and tasks in the `Refine Deal` stage [Esh+13] in Figure 3.3

Stages and Tasks	Guards (Opening sentries)	Terminating sentries
`Draft Term`	r_1: [**on** `+Refine Deal`]	r_3: [**on** `+Term Drafted`]
	r_2: [**on** `+ReDraftTermNeeded` **if** `Refine Deal`]	r_4: [**on** `-Refine Deal`]
`Check Credit`	r_5: [**on** `+Refine Deal`]	r_7: [**on** `+Credit Checked`]
	r_6: [**on** `+ReCheckCreditNeeded` **if** `Refine Deal`]	r_8: [**on** `-Refine Deal`]
`Determine Price`	r_9: [**if** `Term Drafted` $\wedge$ `Credit Checked` $\wedge$ `Refine Deal`]	r_{10}: [**on** `+Price Determined`]
		r_{11}: [**on** `+Draft Term`]
		r_{12}: [**on** `+Check Credit`]
		r_{13}: [**on** `-Refine Deal`]

Table 3.2: GSM sentries for the milestones in the `Refine Deal` stage [Esh+13] in Figure 3.3

Milestones	Achieving sentries	Invalidating sentries
`Term Drafted`	r_{14}: [**on** `C:Draft Term` **if** `Refine Deal`]	r_{15}: [**on** `+Draft Term`]
		r_{16}: [**on** `+ReDraftTermNeeded`]
`Credit Checked`	r_{17}: [**on** `C:Check Credit` **if** `Refine Deal`]	r_{18}: [**on** `+Check Credit`]
		r_{19}: [**on** `+ReCheckCreditNeeded`]
`Price Determined`	r_{20}: [**on** `C:Determine Price` **if** `Refine Deal`]	r_{21}: [**on** `+Draft Term` **if** `ReDraftTermNeeded`]
		r_{22}: [**on** `+Check Credit` **if** `ReCheckCreditNeeded`]
		r_{23}: [**on** `+Determine Price`]
`ReDraftTermNeeded`	r_{24}: [**on** `E:Regulation Change` **if** `Refine Deal`]	r_{26}: [**on** `+Term Drafted`]
	r_{25}: [**if** `credit_level` $> 100,000$ $\wedge$ `Refine Deal`]	
`ReCheckCreditNeeded`	r_{27}: [**if** `risk_level` > 4 $\wedge$ `Refine Deal`]	r_{28}: [**on** `+Credit Checked`]

Services (Tasks). In CMMN *tasks* do the actual work, but they are not implemented as Web Services. However, *tasks* can invoke Web Services depending on the implementation.

Associations. The *associations* between the case file and its tasks, stages, and milestones are described by sentries. Sentries correspond to ECA rules and are defined in the same way as in GSM.

3.3.1 Case Type

This section formalizes the definition of a case type and its components. Without losing generality, the concepts of roles and work assignments in a case type can be ignored because they are irrelevant to the overall execution semantics of CMMN. Therefore, this formalization excludes roles, planning tables, and applicability rules (which specify when planning tables are available for roles to use them).

Definition 3.4. (Case type): In CMMN a case is composed of an information model and a behavioral model. The *Case Type* $\mathcal{T}$ is defined by its data $\mathcal{D}$, and its behavior $\mathcal{B}$, and is described as the tuple

$$\mathcal{T} = \langle \mathcal{D}, \mathcal{B} \rangle$$

where,

$\mathcal{D}$ is a set of *data attributes* that describe the case information model. The model starts with a *case file*, which is a container that holds all of the data belonging to a *case instance*. The schema of the *case file* allows for scalars, records of scalars, and containers such as sets, hierarchical structures, etc. It is also intended to hold structured (records of scalars, database tables, etc.) and unstructured data (documents, pictures, voice recordings, video, etc.). By default it does not contain status data. Elements $d \in \mathcal{D}$ have a predefined life cycle as described in Figure 5.6.

$$\mathcal{D} = \mathcal{D}_{casefile} \cup \mathcal{D}_{discrete} \cup \mathcal{D}_{container}$$

where,

$\mathcal{D}_{casefile}$ is a single element set containing the *case file*. Each case has a single *case file*, which acts as a container for all of the case data.

$\mathcal{D}_{discrete}$ is a set of all the *discrete data* in the case. *Discrete data* can be structured or unstructured.

$\mathcal{D}_{container}$ is a set of *data containers*, with its root being the case file.

$\mathcal{B}$ is a tuple describing the case's *behavioral model* (see Definition 3.5).

Definition 3.5. (Case behavior): The behavior $\mathcal{B}$ of a case type $\mathcal{T}$ is described by its stages $\mathcal{St}$, tasks $\mathcal{Ta}$, milestones $\mathcal{Mi}$, event listeners $\mathcal{Ev}$, hierarchy $\mathcal{H}$, and rules $\mathcal{R}$. The *case behavior* is described by the tuple

$$\mathcal{B} = \langle \mathcal{St}, \mathcal{Ta}, \mathcal{Mi}, \mathcal{Ev}, \mathcal{H}, \mathcal{R} \rangle$$

where,

$\mathcal{St}$ is a set of *stages*. It is partitioned into four types of stages (a single *case plan*, *planned stages*, *discretionary stages*, and *plan fragments*). All four types of stages have different characteristics, but all of them are behavioral containers.

$$\mathcal{St} = \mathcal{St}_{case} \cup \mathcal{St}_{planned} \cup \mathcal{St}_{discretionary} \cup \mathcal{St}_{fragment}$$

where,

$\mathcal{S}t_{case}$ is a single element set. The elements represent the *case plan*. The life cycle of the *case plan* is described in Figure 5.4.

$\mathcal{S}t_{planned}$ is a set of *planned stages*. These are stages contained in the execution plan, as opposed to *discretionary stages* that can be manually added to the execution plan during runtime.

$\mathcal{S}t_{discretionary}$ is a set of *discretionary stages*. The life cycle for elements of both *planned stages* $\mathcal{S}t_{planned}$ and *discretionary stages* $\mathcal{S}t_{discretionary}$ is described in Figure 5.8.

$\mathcal{S}t_{fragment}$ is a set of *plan fragments*. *Plan fragments* are groups of discretionary items. *Plan fragments* do not have any execution semantics.

$\mathcal{S}t_{case}$, $\mathcal{S}t_{planned}$, $\mathcal{S}t_{discretionary}$, and $\mathcal{S}t_{fragment}$ are pairwise disjoint.

$\mathcal{T}a$ is a set of *tasks*. The set of *tasks* $\mathcal{T}a$ is partitioned into two types of tasks (*planned tasks* and *discretionary tasks*).

$$\mathcal{T}a = \mathcal{T}a_{planned} \cup \mathcal{T}a_{discretionary}$$

where,

$\mathcal{T}a_{planned}$ is a set of *planned tasks*.

$\mathcal{T}a_{discretionary}$ is a set of *discretionary tasks*. The life cycle for elements of both *planned tasks* $\mathcal{T}a_{planned}$ and *discretionary tasks* $\mathcal{T}a_{discretionary}$ is described in Figure 5.8.

$\mathcal{T}a_{planned}$ and $\mathcal{T}a_{discretionary}$ are pairwise disjoint.

$\mathcal{M}i$ is a set of *milestones*. The life cycle of *milestones* is described in Figure 5.10.

$\mathcal{E}v$ is a set of *event listeners*. The life cycle of *event listeners* is described in Figure 5.10.

$\mathcal{H}$ is a binary relationship ($\mathcal{H} \subseteq \mathcal{S}t \times \{\mathcal{S}t \cup \mathcal{T}a \cup \mathcal{M}i \cup \mathcal{E}v\}$) describing a hierarchical organization of *stages* $\mathcal{S}t$, *tasks* $\mathcal{T}a$, *milestones* $\mathcal{M}i$, and *event listeners* $\mathcal{E}v$ in the case.

$\mathcal{R}$ is a tuple that describes the rules governing the behavior of the case type (see Definition 3.7).

Definition 3.6. (Case hierarchy): Stages $\mathcal{S}t$ in the case type can be nested within other stages. In addition, stages can contain tasks, milestones, and event listeners thereby creating a hierarchy in the case type. The *case hierarchy* $\mathcal{H}$ is a binary relationship ($\mathcal{H} \subseteq \mathcal{S}t \times \{\mathcal{S}t \cup \mathcal{T}a \cup \mathcal{M}i \cup \mathcal{E}v\}$) that describes the organization of the case type. The hierarchical relationship is defined as a set of ordered pairs

$$\begin{aligned}\mathcal{H} =&\{\langle x,y\rangle \mid x \in \mathcal{St} \ \wedge \ y \in \mathcal{St} \cup \mathcal{Ta} \cup \mathcal{Mi} \cup \mathcal{Ev} \ \wedge \ x \neq y \ \wedge \ y \text{ is reachable from } x \\ &\wedge \ \neg(\exists z)(z \in \mathcal{St} \ \wedge \ z \text{ is reachable from } x \ \wedge \ y \text{ is reachable from } z)\}\end{aligned}$$

The hierarchy of a case type implements a tree, with the root of the tree being the single element in $\mathcal{St}_{case}$ $(\neg(\exists\langle x,y\rangle)(\langle x,y\rangle \in \mathcal{H} \ \wedge \ y \in \mathcal{St}_{case}))$. The leaves of the tree correspond to tasks in $\mathcal{Ta}$, milestones in $\mathcal{Mi}$, and event listeners in $\mathcal{Ev}$. The non-leaf nodes of the tree are composite stages in $\mathcal{St}$. Elements in this tree are said to be "reachable" from x, if they are present in the sub-tree with root x.

Definition 3.7. (Case behavioral rules): The rules of a case type $\mathcal{R}$ are described by its conditions $\widehat{\varphi}$, sentries $\widehat{\mathscr{S}}$, entry criteria $\widehat{\mathscr{E}}$, exit criteria $\widehat{\mathscr{X}}$, manual activation $\widehat{\mathscr{M}}$, required $\widehat{\mathscr{R}}$, repetition $\widehat{\mathscr{N}}$ rules, and autocomplete $\widehat{\mathscr{A}}$. The rules are defined as a tuple

$$\mathcal{R} = \langle \widehat{\varphi}, \widehat{\mathscr{S}}, \widehat{\mathscr{E}}, \widehat{\mathscr{X}}, \widehat{\mathscr{M}}, \widehat{\mathscr{R}}, \widehat{\mathscr{N}}, \widehat{\mathscr{A}} \rangle$$

where,

$\widehat{\varphi}$ is a set of Boolean *expressions* defined in a similar way as φ in a sentry (see Definition 3.8), but is used for rules instead of for sentries.

$\widehat{\mathscr{S}}$ is a set of *sentries.* Sentries are defined in a similar way to the way they are in GSM. The main difference is that a sentry in CMMN may have multiple event expressions $\dot{\xi}$.

$\widehat{\mathscr{E}}$ is an *entry criteria* relation that associates a sentry with a stage, task, or milestone that will be activated when the sentry is satisfied. It is a binary relationship ($\widehat{\mathscr{E}} \subseteq \widehat{\mathscr{S}} \times \{\mathcal{St}_{planned} \cup \mathcal{St}_{discretionary} \cup \mathcal{Ta} \cup \mathcal{Mi}\}$) defined as follows:

$$\widehat{\mathscr{E}} = \{\langle x,y\rangle \mid x \in \widehat{\mathscr{S}} \wedge y \in \{\mathcal{St}_{planned} \cup \mathcal{St}_{discretionary} \cup \mathcal{Ta} \cup \mathcal{Mi}\} \wedge x \text{ entry criteria of } y\}$$

$\widehat{\mathscr{X}}$ is an *exit criteria* relation that associates a sentry with a stage or task that will terminate when the sentry is satisfied. It is a binary relationship ($\widehat{\mathscr{X}} \subseteq \widehat{\mathscr{S}} \times \{\mathcal{St}_{case} \cup \mathcal{St}_{planned} \cup \mathcal{St}_{discretionary} \cup \mathcal{Ta}\}$) defined as follows:

$$\widehat{\mathscr{X}} = \{\langle x,y\rangle \mid x \in \widehat{\mathscr{S}} \wedge y \in \{\mathcal{St}_{case} \cup \mathcal{St}_{planned} \cup \mathcal{St}_{discretionary} \cup \mathcal{Ta}\} \wedge x \text{ exit criteria of } y\}$$

$\widehat{\mathscr{M}}$ is a *manual activation* relation that associates a Boolean expression with a stage or task that will require manual activation. It is a binary relationship ($\widehat{\mathscr{M}} \subseteq \widehat{\varphi} \times \{\mathcal{St}_{planned} \cup \mathcal{St}_{discretionary} \cup \mathcal{Ta}\}$) defined as follows:

$$\widehat{\mathscr{M}} = \{\langle x,y\rangle \mid x \in \widehat{\varphi} \wedge y \in \{\mathcal{St}_{planned} \cup \mathcal{St}_{discretionary} \cup \mathcal{Ta}\} \wedge x \text{ manual activation rule of } y\}$$

$\widehat{\mathscr{R}}$ is a *required* relation that associates a Boolean expression with a stage, task, or milestone that indicates when these are required. It is a binary relationship ($\widehat{\mathscr{R}} \subseteq \widehat{\varphi} \times \{\mathcal{St}_{planned} \cup \mathcal{Ta}_{planned} \cup \mathcal{Mi}\}$) defined as follows:

$$\widehat{\mathscr{R}} = \{\langle x, y\rangle \mid x \in \widehat{\varphi} \ \wedge \ y \in \{\mathcal{St}_{planned} \cup \mathcal{Ta}_{planned} \cup \mathcal{Mi}\} \ \wedge \ x \text{ required rule of } y\}$$

$\widehat{\mathscr{N}}$ is a *repetition* relation that associates a Boolean expression with a stage, task, or milestone that indicates that these can repeat execution. It is a binary relationship ($\widehat{\mathscr{N}} \subseteq \widehat{\varphi} \times \{\mathcal{St}_{planned} \cup \mathcal{St}_{discretionary} \cup \mathcal{Ta} \cup \mathcal{Mi}\}$) defined as follows:

$$\widehat{\mathscr{N}} = \{\langle x, y\rangle \mid x \in \widehat{\varphi} \wedge y \in \{\mathcal{St}_{planned} \cup \mathcal{St}_{discretionary} \cup \mathcal{Ta} \cup \mathcal{Mi}\} \wedge x \text{ repetition rule of } y\}$$

$\widehat{\mathscr{A}}$ is a set of *stages* with an autocomplete flag.

$$\widehat{\mathscr{A}} = \{a \mid a \in \mathcal{St}_{case} \cup \mathcal{St}_{planned} \cup \mathcal{St}_{discretionary} \ \wedge \ \text{autoComplete}(a)\}$$

Definition 3.8. (CMMN Sentry): A *CMMN sentry* x in $\widehat{\mathscr{S}}$ for a case type $\mathcal{T}$ is an expression $\dot{\chi}$

$$\dot{\chi} = \begin{cases} [\textbf{on } \dot{\xi} \textbf{ if } \varphi] & \text{event expression } \dot{\xi} \text{ and condition } \varphi \\ [\textbf{on } \dot{\xi}] & \text{event expression } \dot{\xi} \text{ only} \\ [\textbf{ if } \varphi] & \text{condition } \varphi \text{ only} \end{cases}$$

where, $\dot{\xi}$ is an optional *event expression* comprising one or more events, and φ is an optional well-formed Boolean *expression* that may or may not use attributes from $\mathcal{D}$. A CMMN sentry $\dot{\chi}$ differs from a GSM sentry χ in the event expression. While the event expression ξ of a GSM sentry χ comprises a single event, the event expression $\dot{\xi}$ of a CMMN sentry $\dot{\chi}$ comprises one or more events. An event expression $\dot{\xi}$ having n events ($n > 0$) is expressed as an unordered tuple $[\![e_1, \ \ldots \ , e_n]\!]$, where each event is a CMMN standard event. We write the event e as `<state transition>:<element>`, where the element is one of the following:

- *Data element* ($x \in \mathcal{D}$) with the state transition described in Figure 5.6.
- *Case instance* ($\mathcal{St}_{case}$) with the state transition described in Figure 5.4.
- *Stage and task elements* ($x \in \mathcal{St}_{planned} \ \vee \ x \in \mathcal{St}_{discretionary} \ \vee \ x \in \mathcal{Ta}$) with the state transition described in Figure 5.8.
- *Event listener and milestone elements* ($x \in \mathcal{Ev} \ \vee \ x \in \mathcal{Mi}$) with the state transition described in Figure 5.10.

A condition φ can access all of the data in $\mathcal{D}$, and can be used to reason about the state of elements as defined in Figures 5.4, 5.6, 5.8 and 5.10.

3.3.2 Case Program

A CMMN program is a set of case types $\mathcal{T}$.

Definition 3.9. (Program): A CMMN *program* $\lceil P \rceil$ is a set of case types $\mathcal{T}_i$ for $i \leq n$.

$$\lceil P \rceil = \bigcup_{i=1}^{n} \mathcal{T}_i = \left\langle \bigcup_{i=1}^{n} \mathcal{D}_i \ , \ \bigcup_{i=1}^{n} \mathcal{B}_i \right\rangle$$

As a convenient notation, we use $\mathcal{D}^P$ and $\mathcal{B}^P$ to denote the data attributes $\mathcal{D}$ and the behavior $\mathcal{B}$ for a program $\lceil P \rceil$.

3.3.3 Case Model

A CMMN program can be represented by a CMMN model.

Definition 3.10. (Model): A CMMN [OMG14a] *model* $\mathcal{C}$ is a collection of model elements $\mathcal{E}$ with annotators $\mathcal{A}$ that are related by scope $\mathcal{U}$ and event $\mathcal{V}$ relationships. A CMMN *model* $\mathcal{C}$ provides a visual representation of a program $\lceil P \rceil$. A model is defined as a tuple

$$\mathcal{C} = \langle \mathcal{E}, \mathcal{U}, \mathcal{V}, \mathcal{A} \rangle$$

where,

$\mathcal{E}$ is a set of *modeling elements.* $\mathcal{E}$ is strongly typed and each element in $\mathcal{E}$ belongs to one of the following types: *scope*, *data*, *plan*, or *optional.* Table 3.3 shows the CMMN elements. We use $\wr x$ to indicate a visual representation of x. Therefore, for a program $\lceil P \rceil$ we have

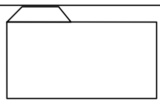
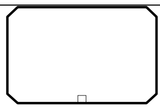
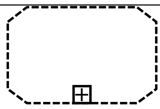
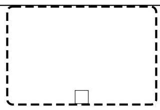
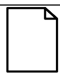
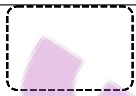
$$\mathcal{E} = \{x \mid (\exists y)(y \in \mathcal{St}^P \cup \mathcal{D}^P \cup \mathcal{Ta}^P \cup \mathcal{Mi}^P \cup \mathcal{Ev}^P \cup \widehat{\mathscr{E}}^P \cup \widehat{\mathscr{X}}^P \ \wedge \ x = \wr y)\}$$

Note that $|\mathcal{E}| \leq |\ \mathcal{St} \cup \mathcal{D} \cup \mathcal{Ta} \cup \mathcal{Mi} \cup \mathcal{Ev} \cup \widehat{\mathscr{E}}^P \cup \widehat{\mathscr{X}}^P\ |$, because not every element in the case type $\mathcal{T}$ needs to be modeled. In particular, some of the data in $\mathcal{D}$ may not be modeled, and since connectors ($\widehat{\mathscr{E}}^P \cup \widehat{\mathscr{X}}^P$) are optional these also might not be modeled.

$\mathcal{U}$ is a *scope relationship*, which is a binary relationship in which two elements x and y in $\mathcal{E}$ are related if and only if they are contained in the same scope. Note that $[\![x, y]\!] \in \mathcal{U}$ is an unordered pair.

$\mathcal{V}$ is an *event relationship*, which is a binary relationship in which two elements x and y in $\mathcal{E}$ are related if and only if an event from one (x) triggers the other (y). Note that $\langle x, y \rangle \in \mathcal{V}$ is an ordered pair.

Table 3.3: CMMN elements

Type	**Element** ($\mathcal{E}$)	**Name**
Scope		case
		stage
		discretionary stage
		plan fragment
Data		case file item
Plan		task
		discretionary task
		event listener
		milestone
Optional	....	connector (sentry)

$\mathcal{A}$ is a *set of annotators* used to indicate the characteristics of elements in $\mathcal{E}$. Each annotator 'a' in $\mathcal{A}$ is related to one and only one element x in $\mathcal{E}$. There are three types of annotators, namely *decorators*, *sentries*, and *markers*. Most elements, x in $\mathcal{E}$, can be associated with a single marker, one of each decorator (collapsed or expanded), and multiple sentries. Table 3.4 shows the annotators. Some of the annotators indicate the presence of rules in $\mathcal{R}^P$. Sentries correspond to entry criteria $\widehat{\mathscr{E}}^P$ and exit criteria $\widehat{\mathscr{X}}^P$ rules in $\mathcal{R}^P$. The autocomplete, manual activation, repetition, and required decorators correspond with autocomplete $\widehat{\mathscr{A}}^P$, manual activation $\widehat{\mathscr{M}}^P$, repetition $\widehat{\mathscr{N}}^P$, and required $\widehat{\mathscr{R}}^P$ rules in $\mathcal{R}^P$.

Definition 3.11. (Scope element set): A *scope element* is an element that can contain other elements. The set of *scope elements* $\mathcal{M}$ is a subset of $\mathcal{E}$ ($\mathcal{M} \subseteq \mathcal{E}$).

$$\mathcal{M} = \{z \mid \text{ type-of}(z) \in \text{ Scope}\}$$

where, the value of type-of(z) is described by the type column in Table 3.3.

Table 3.4: CMMN Annotators

Type	**Annotator** ($\mathcal{A}$)	**Name**
Decorator		collapsed planning table
		expanded planning table
	■	autocomplete
	⊞	collapsed
	⊟	expanded
	▷	manual activation
	#	repetition
	!	required
Sentry	◊	entry criterion
	◆	exit criterion
Marker		non-blocking human
		process
		case
		participant
		timer

As shown in Table 3.3, the *scope elements* are the *case*, *stage*, *discretionary stage*, and *plan fragment*. Note that for a program $\lceil P \rceil$ the set of scope elements $\mathcal{M}$ corresponds to $\mathcal{S}\boldsymbol{t}^P$ (we could write $\mathcal{M}$ as $\mathcal{M} = \{z \mid (\exists y)(y \in \mathcal{S}\boldsymbol{t}^P \ \wedge \ z = \wr y)\}$).

Definition 3.12. (Case element set): A *case element* is a special scope element $z \in \mathcal{M}$ that starts a case definition. A model $\mathcal{C}$ can contain multiple case definitions, each one with a corresponding *case element*. The set of *case elements* $\mathcal{L}$ is a subset of scope elements ($\mathcal{L} \subseteq \mathcal{M}$).

$$\mathcal{L} = \{z \mid \ \text{isa}(z) = \ \text{case}\}$$

where, the value of isa(z) is described in the name column in Table 3.3.

Note that each element z in $\mathcal{L}$ corresponds to a $\mathcal{S}\boldsymbol{t}_{case}$ in the program $\lceil P \rceil$ (we could write $\mathcal{L}$ as $\mathcal{L} = \{z \mid (\exists y)(y \in \mathcal{S}\boldsymbol{t}^P{}_{case} \ \wedge \ z = \wr y)\}$). Any non-empty model $\mathcal{C}$ must contain at least one *case element*. A *case element* can contain other elements, but it cannot be contained within any element.

Therefore, $\mathcal{C} \neq \varnothing \iff \mathcal{L} \neq \varnothing$.

Definition 3.13. (Module): A *module* of a model $\mathcal{C}$ is defined by a scope element $z \in \mathcal{M}$, as the following tuple

$$\ulcorner z \urcorner = \langle \mathcal{E}_z, \mathcal{U}_z, \mathcal{V}_z, \mathcal{A}_z \rangle$$

where,

$\mathcal{E}_z$ is a subset of $\mathcal{E}$ defined as

$$\mathcal{E}_z = \{z\} \cup \{x \mid x \text{ is reachable from } z\}.$$

$\mathcal{U}_z$ is a subset of $\mathcal{U}$ defined as

$$\mathcal{U}_z = \{[\![x, y]\!] \mid [\![x, y]\!] \in \mathcal{U} \ \wedge \ x \in \mathcal{E}_z \ \wedge \ y \in \mathcal{E}_z\}$$

$\mathcal{V}_z$ is a subset of $\mathcal{V}$ defined as

$$\mathcal{V}_z = \{\langle x, y\rangle \mid \langle x, y\rangle \in \mathcal{V} \ \wedge \ x \in \mathcal{E}_z \ \wedge \ y \in \mathcal{E}_z\}$$

$\mathcal{A}_z$ is a subset of $\mathcal{A}$ defined as

$$\mathcal{A}_z = \{a \mid a \in \mathcal{A} \ \wedge \ (\exists y)(y \in \mathcal{E}_z \ \wedge \ y = \text{annotated-by}(a))\}$$

where, annotated-by(a) describes the element in $\mathcal{E}$ that 'a' annotates.

Note that z may contain other scope elements, in which case all of the elements contained in these scope elements are also in $\ulcorner z \urcorner$. In other words $x \in \mathcal{E}_z \ \wedge \ x \in \mathcal{M} \implies \mathcal{E}_x \subset \mathcal{E}_z$. In addition, $\mathcal{E}$ is partitioned by $\mathcal{E}_x$ and $\mathcal{E}_y$

$$\mathcal{E} = \mathcal{E}_x \cup \mathcal{E}_y \ \wedge \ \mathcal{E}_x \cap \mathcal{E}_y = \varnothing \iff \mathcal{L} = \{x, y\} \tag{3.3.1}$$

By slight abuse of notation, we will use $\mathcal{E}_P$ with an upper case P to indicate the set $\mathcal{E}$ of model P, and $\mathcal{E}_z$ with a lower case z to indicate the set $\mathcal{E}$ of module $\ulcorner z \urcorner$.

Definition 3.14. (Scope relationship): A *scope relationship* relates two elements if and only if they are in the same scope $z \in \mathcal{M}$. The scope is a binary relationship $\mathcal{U}$ on $\mathcal{E}$ ($\mathcal{U} \subseteq \mathcal{E} \times \mathcal{E}$), and is defined as follows:

$$\begin{aligned} \mathcal{U} = \{[\![x, y]\!] \mid x \neq y \ \wedge \ (\exists z)(& z \in \mathcal{M} \ \wedge \ x \in \mathcal{E}_z \ \wedge \ y \in \mathcal{E}_z \\ & \wedge \ x \neq z \ \wedge \ y \neq z) \\ & \wedge \ \neg(\exists w)(w \in \mathcal{M}_z \ \wedge \ (x \in \mathcal{E}_w \ \vee \ y \in \mathcal{E}_w))\} \end{aligned}$$

Due to the declarative nature of CMMN, there is no sequence or control-flow relationship between its elements, however, scope plays an important role in the model by grouping elements.

Definition 3.15. (Event relationship): An *event relationship* relates two elements if and only if an event from one of them triggers an entry or exit criterion in the other element. The event is a binary relationship $\mathcal{V}$ on $\mathcal{E}$ ($\mathcal{V} \subseteq \mathcal{E} \times \mathcal{E}$), defined as follows:

$$\mathcal{V} = \{\langle x, y\rangle \mid [\![x, y]\!] \in \mathcal{U} \ \wedge \ x \xrightarrow{event} y\}$$

Events play an important role in CMMN because entry and exit criteria for elements can be triggered by events. Events cannot cross different scopes, therefore the two modeling elements must already be related by scope in order for the elements to be triggered.

3.3.3.1 Examples

Three examples of CMMN models have been described in this section. The example models have been used to illustrate CMMN concepts and grammar. The models are grammatically and semantically correct, but they are not intended to represent any real world processes. Annotators are not labeled in CMMN, but for illustrative purposes dotted lines have been used to associate annotator labels with the annotator icons in the model.

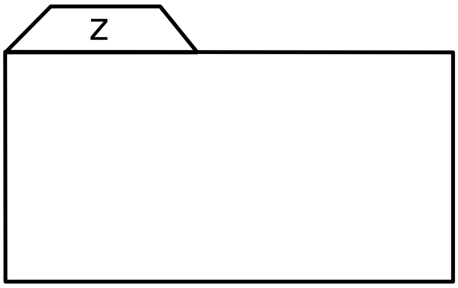


Figure 3.4: Model with one element (a case)

Example 3.1. Minimum case model. Every model must contain at least one case element (see Definition 3.12). Figure 3.4 shows the minimum valid CMMN model with a single case element. The model has one model element $\mathcal{E} = \{z\}$, one scope element $\mathcal{M} = \{z\}$, and one case element $\mathcal{L} = \{z\}$. The scope relationship is empty $\mathcal{U} = \varnothing$. The event relationship is empty $\mathcal{V} = \varnothing$. The annotator set is empty $\mathcal{A} = \varnothing$. This example contains a single module $\ulcorner z \urcorner$ with $\mathcal{E}_z = \{z\}$. Note that $\mathcal{E} = \mathcal{E}_z$.

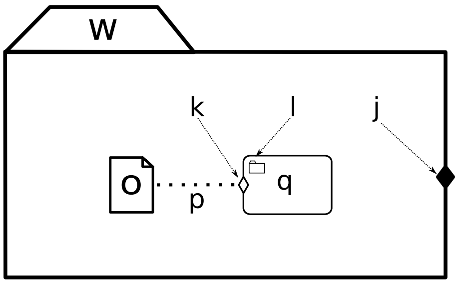


Figure 3.5: Model with four elements

Example 3.2. Simple case model. Figure 3.5 shows a model with a single case element w containing one case file item o, a connector p, and a task q. An event from case file item o triggers the entry criteria 'k' that will allow task q to execute. The optional connector p visualizes the event propagation between case file item o and the entry criteria 'k' of task q. The model has four model elements $\mathcal{E} = \{w, o, p, q\}$, one scope element $\mathcal{M} = \{w\}$, and one case element $\mathcal{L} = \{w\}$. The scope relationship is $\mathcal{U} = \{[\![o, p]\!], [\![o, q]\!], [\![p, q]\!]\}$. The event relationship is $\mathcal{V} = \{\langle o, q\rangle\}$. The annotator set is $\mathcal{A} = \{k, j, l\}$. This example contains one module $\ulcorner w \urcorner$ with $\mathcal{E}_w = \{w, o, p, q\}$. Note that $\mathcal{E} = \mathcal{E}_w$

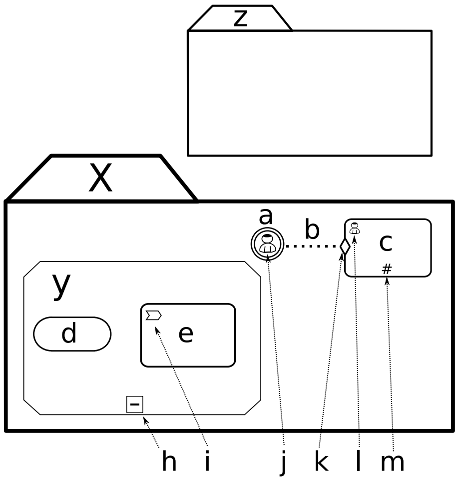


Figure 3.6: Model with eight elements

Example 3.3. Model with two cases. Figure 3.6 shows a model with two case elements x and z. This model contains three scope elements $\mathcal{M} = \{x, y, z\}$, and two case elements $\mathcal{L} = \{x, z\}$. The model has eight modeling elements $\mathcal{E} = \{a, b, c, d, e, x, y, z\}$. There are three scopes, x with $\{a, b, c, y\}$, y with $\{d, e\}$, and one empty scope z. The scope relationship is $\mathcal{U} = \{[\![a, c]\!], [\![a, b]\!], [\![a, y]\!], [\![b, c]\!], [\![b, y]\!], [\![c, y]\!], [\![d, e]\!]\}$. Note that $\{a, b, c\}$ and $\{d, e\}$ fall into different scopes, and so, $[\![a, d]\!] \notin \mathcal{U}$. The same is true for $[\![a, e]\!], [\![b, d]\!], [\![b, e]\!], [\![c, d]\!]$, and $[\![c, e]\!]$ that do not appear in $\mathcal{U}$. The event relationship is $\mathcal{V} = \{\langle a, c\rangle\}$. The annotator set is $\mathcal{A} = \{h, i, j, k, l, m\}$. This example contains three modules, $\ulcorner x \urcorner$ with $\mathcal{E}_x = \{x, a, b, c, y, d, e\}$, $\ulcorner y \urcorner$ with $\mathcal{E}_y = \{y, d, e\}$, and $\ulcorner z \urcorner$ with $\mathcal{E}_z = \{z\}$. Note that in terms of Equation (3.3.1) $\mathcal{L} = \{x, z\} \implies \mathcal{E} = \mathcal{E}_x \cup \mathcal{E}_z$. Figure 7.1 shows a tree view of the modeling elements $\mathcal{E}$ in this example. The tree view is for illustrative purposes and is not a CMMN diagram.

3.4 Differences between GSM and CMMN

Although GSM and CMMN contain similar concepts there are important differences between them. In particular, CMMN entities (e.g., stages, tasks, milestones, event listeners) are objects with a very specific life cycle and the ability to create multiple parallel instantiations. GSM entities (e.g., stages, tasks, milestones) on the other hand are structures with a name, a Boolean state, and a set of rules (sentries). The Boolean state of a GSM entity prevents multiple parallel executions of the structure. As previously discussed both GSM and CMMN can be categorized as BA based approaches.

Table 3.5: Comparing GSM and CMMN using BALSA

BALSA	**GSM**	**CMMN**
Business **A**rtifacts (Information Model)	Attributes are described by a relational database schema	Data is stored in the case file, and is not limited to attributes
Lifecycle	Modeled using guards, tasks, stages, and milestones	Modeled using tasks, stages, milestones, event listeners, and entry and exit criteria
Services (Tasks)	Tasks implemented by Web Services invocations based on events	Tasks are not required to be implemented as Web Services
Associations	Implemented by sentries using ECA rules	Implemented by sentries

The BALSA dimensions are useful for conducting comparisons between BA approaches. Kunchala et al. [Kun+15] have compared GSM to ArtiNets [KS11], the AXML Artifact model [Abi+09], Business Process Model and Notation (BPMN) [OMG13] with extensions [LN12], and the ACP-i model [Yon+11] using the BALSA framework. Table 3.5 shows a high level comparison of CMMN and GSM using the four BALSA dimensions. Table 3.6 presents a detailed comparison between GSM and CMMN.

CMMN can be considered an evolution of GSM. The CMMN approach incorporates most of the GSM approach, with the exception of event payload and the ability to invalidate milestones. However, CMMN adds important functionalities that are missing in GSM, including:

- The support of unstructured data. While GSM only supports structured data in a relational database CMMN is less constrained.
- The concept of discretionary stages and tasks introduced by CMMN is not present in GSM.
- External events are explicitly modeled using event listeners, while in GSM they are just embedded in the sentries.
- Event expression with multiple events is supported by CMMN, while in GSM a sentry can only contain a single event expression.
- CMMN defines a set of rules that enforce the behavior of the case type (required and repetition rules), and describe the case workers interaction with a case instance (manual activation, and autocomplete rules). Those rules describe functionality that is not present in GSM.
- The support of human interaction to affect the life cycle of the case, stages, tasks, event listeners, milestones, and data by using planning is introduced in CMMN, and is not present in GSM.

Table 3.6: Differences between GSM and CMMN

Concept	**GSM**	**CMMN**
autocomplete	Default behavior.	When false, a case worker must manually complete the case or stage.
case	N/A	The case is similar to a stage that encloses the complete BA behavior. The case implements the complete behavioral model.
case file	N/A	Contains the all of the case data. It implements the information model.
case file item	Corresponds to attributes.	In addition to attributes, it can include containers and documents.
case plan	N/A	Corresponds to the instance execution plan, but new (discretionary) stages and tasks can be added by the case workers.
containers	N/A	The case file can be organized as a file system with containers.
data	Attributes modeled in a relational database.	Data is organized in a case file. It can be stored in a content management system, a file system, or a database.
data events	N/A	Changes to data in the case file generate standard events.
discretionary	N/A	Stages and tasks can be discretionary, in which case they are only executed if a case worker adds them to the execution plan.
entity life cycle	Maintained in a Boolean attribute.	Based on a state machine.
events	Provides incoming and generated events.	Provides event listeners and standard events.
event expressions	Contains a single event.	Supports multiple events, in which case they form an 'AND' condition.
event listener	Implicit.	Explicitly defined for external events.
event payload	Events can have a payload that corresponds to a subset of attributes.	N/A
guards	Control when a stage opens (i.e., start executing).	Correspond to an entry criteria, which controls when a task or stage starts execution, or when a milestone occurs.
invalidation	Milestones can be invalidated	N/A

Continued on next page

Table 3.6 – *Continued from previous page*

Concept	**GSM**	**CMMN**
manual activation	N/A	After an entry criteria becomes true the case workers must decide if the task or stage should be executed.
milestones	Can be invalidated and the state is stored in a Boolean attribute.	Can only be achieved and have a life cycle.
repetition	Default behavior, but constrained to serial repetition.	Controlled by a Boolean expression. Allows for parallel and serial execution.
required	N/A	Controlled by a Boolean expression and indicates stages, tasks, and milestones that must be executed in order for the enclosing stage or case to complete. Needed because case workers can disable stages, tasks, and milestones.
runtime planning	N/A	Planning is an important characteristic and includes the runtime plan, planning tables, and plan fragments.
semantics	Based on business relevant steps (B-step) and snapshot.	Event driven, but not formally defined.
sentries	Expressions described in OCL.	Expressions can be described in any expression language. Can contain multiple event expressions.
stages	Have a name and a Boolean status.	An object that has a life cycle.
status	Stages and milestones have a Boolean attribute that describes their status.	Described by state machine and there are no status attributes.
tasks	Implemented as web services.	Implementation is not formalized.
task types	N/A	Includes blocking and non-blocking human tasks, process tasks, and case tasks.
terminators.	Control when a stage closes (i.e., ends execution)	Correspond to exit criteria, which control when a task, stage, or the case completes execution.

3.5 Summary

This chapter described and compared GSM [Hul+11b] and CMMN [OMG14a]. It provided formal definitions for the CMMN case type, case program, and case model that will be used in Chapters 5 and 7. The chapter also provided a clear and detailed comparison of GSM and CMMN. Some material from this chapter has been published in Marin et al. [Mar+15b; Mar+16].

Chapter 4

Case Management Model and Notation Method Complexity

This chapter evaluates the method complexity of the Case Management Model and Notation (CMMN) [OMG14a] version 1.0 specification and compares it to other popular process modeling notations. The understanding of CMMN method complexity presented here will provide background information for the definition of CMMN metrics in Section 7.2. Material from this chapter has been published in Marin et al. [Mar+14b].

This exploratory research is among the first contributions towards providing an understanding of CMMN's complexity in the context of other process modeling methods. The research analyzed the method complexity of CMMN version 1.0 using the approach proposed by Rossi and Brinkkemper [RB96]. In order to compare the results to other popular process modeling methods, this research adjusted Rossi and Brinkkemper's approach by adjusting it to Unified Modeling Language (UML) [OMG09c] to match the work done by Indulska et al. [Ind+09b]. It then compared the CMMN version 1.0 results with the results that were obtained by Indulska et al. [Ind+09b], Recker et al. [Rec+09], and Siau and Cao [SC02]. Based on the findings CMMN method complexity compares favorably to other methods including Business Process Model and Notation (BPMN) [OMG13], Unified Modeling Language Activity Diagram (UML AD) [OMG09c], and Event-driven Process Chain (EPC) [van99].

This chapter is organized as follows. Section 4.1 introduces method complexity and its importance. Section 4.2 describes the methodology used in this chapter to evaluate CMMN method complexity and to compare it against other process modeling notations. Section 4.3 describes the method complexity calculation for CMMN. Section 4.4 compares CMMN method complexity with other process modeling notations and presents the findings.

4.1 Method Complexity

Method complexity allows us to compare modeling notations and is important because it is expected to affect the learnability, ease of use, and overall use of a method [Rec+09; RB96]. CMMN promotes a data-centric and declarative perspective to process modeling [Mar+13], which makes it important to understand how it compares with other methods used for modeling business processes, such as BPMN, UML AD, EPC amongst others. BPMN, UML ADs, and EPC are well-known process methods with modeling notations. This chapter compares specific versions of their specifications to CMMN version 1.0.

This research is based on the CMMN version 1.0 formal meta-model described in the specification. Meta-models are important because they describe the expressive power of a method by representing its vocabulary (i.e., concepts and properties) and valid constructs (i.e., relationships and roles) [RB96]. Most current process modeling notations are described in their formal specifications using UML meta-models: this is the case for CMMN version 1.0 and BPMN version 2.0 [OMG13]. For illustrative purposes, a small subset of the CMMN formal meta-model is presented in Figure 4.1, which shows a portion of the case class diagram in UML. Other modeling notations like UML version 2.4 [OMG11] are also described using UML meta-models. As described by UML [OMG11] a model is an instance from a meta-model, and there may be multiple levels of meta-modeling. For example, standards organizations such as the Object Management Group (OMG) [Gro17] rely on multiple levels of UML models to describe their formal specifications. The meta-models for CMMN, BPMN, and UML ADs are described using UML models. In turn, models described in CMMN, BPMN, or UML ADs conform to the corresponding specifications of the UML meta-model. This research focuses on CMMN's version 1.0 formal meta-model as described in the OMG specification using UML.

The evaluation of CMMN's method complexity and its comparison to BPMN are important topics because CMMN is designed to complement BPMN [OMG09b]. Although BPMN is widely used it has its fair share of critics. Some researchers have criticized the BPMN standard for being too complicated for users to utilize. In 2007, zur Muehlen et al. [zur+07] concluded that BPMN has a complex modeling vocabulary. Recker et al. [Rec+09] found BPMN version 1.2 [OMG09a] to be more complex than UML version 2.2. In order to deal with the complexity of BPMN some researchers have identified common subsets of the BPMN notation that are in use today. In particular, zur Muehlen and Recker [zR08] identified three subsets of the BPMN notation that are in common use. These include a subset used by the U.S. department of defense [Ind+09b; U.S09], a subset of commonly used BPMN constructs based on the analysis of 120 models [Ind+09b; zR08], and a use case analysis for a truck dealership in the U.S. [Ind+09b; zH08]. These subsets were analyzed by Indulska et al.

[Ind+09b]. The same subsets have been used in this study in order to compare BPMN with CMMN.

Siau and Rossi [SR98] noted that there are a large number of modeling methods, and that new methods continue to be created by practitioners and researchers. They felt that this could be problematic not as a result of the large number of modeling methods, but rather because there is a lack of available techniques for evaluating and comparing these methods. In response to this they developed a comprehensive review of the approaches used for evaluating modeling methods. Siau and Rossi provided four reasons as to why they believed it is important to compare methods. First, researchers need to understand the nature of the methods in order to classify, study, and improve them. Secondly, practitioners can use comparisons as a way to select between methods. Thirdly, method developers need to know the strengths and weaknesses of the various methods in order to design better methods. Fourthly, since no method is suitable for all situations, comparison can assist with selecting the correct method for a particular situation. Siau and Rossi's assessment of modeling methods is applicable to process modeling methods.

There are at least three widely used process modeling methods: EPC, UML AD, and BPMN. These modeling methods have overlapping functionalities, and so most imperative processes can be modeled using any of the three methods. It was only in 2014 that CMMN was added to the list of process modeling methods by the OMG. EPC was introduced in 1992 by Keller et al. [Kel+92] and became popular in the 1990s as a conceptual business process modeling language [Men08]. UML AD appeared in 2001 in UML version 1.4 [OMG01] which according to Dumas and Ter Hofstede [DT01], was intended to model organizational processes (i.e., workflows) . In 2004, BPMN version 1.0 was published as a graphical notation to model business processes [BPM04]. The creators of BPMN evaluated several process modeling methods, including EPC and UML ADs, and decided to consolidate some of the ideas present in these methods into BPMN version 1.0 [BPM04]. Although CMMN addresses the specific case management use case described by Le Clair et al. [Le +09], Reijers et al. [Rei+03], Swenson [Swe10a], and van der Aalst and Berens [vB01] it is useful to compare these modeling methods because of their overlapping functionality.

Siau and Rossi [SR98] categorized the evaluation techniques into empirical and non-empirical techniques. Empirical evaluation techniques include surveys, laboratory experiments, field experiments, case studies, and action research. Non-empirical evaluation techniques include feature comparison, meta-model analysis, metrics analysis, paradigmatic analysis, contingency identification, ontological evaluation, and approaches based on cognitive psychology. Using Siau and Rossi's categorization, Rossi and Brinkkemper's [RB96] meta-model-based method complexity is categorized as metrics analysis, because it uses the method meta-model to compute metrics that are used for the purposes of comparison.

The method complexity comparison used in this research was based on the meta-model method complexity metrics introduced by Rossi and Brinkkemper in 1996 [RB96], which have been used to compare several business process methods. In 2002, this method complexity comparison was used by Siau and Cao [SC02] to evaluate UML version 1.4 [Boo+99] and its techniques, including UML ADs. The same method complexity comparison was used by Indulska et al. in 2009 [Ind+09b] to compare subsets of BPMN version 1.2. In 2009, Recker et al. [Rec+09] utilized Siau and Cao's work on UML version 1.4 to compare UML AD version 1.4 to BPMN version 1.2. This produced a corpus of evaluations that were compared to the CMMN version 1.0 notation that has been used in this thesis.

Using Rossi and Brinkkemper's [RB96] terminology, a process modeling notation like CMMN is considered a *method*. Methods can have multiple *techniques*, which correspond to the multiple diagrams that can be created. For example, UML is a method with multiple techniques including UML ADs, UML Class diagrams, UML State-Chart diagrams, etc. Both CMMN version 1.0 and BPMN version 1.2 have a single technique: the case plan model for CMMN, and the business process diagram for BPMN. However, the latest BPMN version 2.0, has multiple techniques, including process diagrams, collaboration diagrams, conversation diagrams, and choreography diagrams. This study uses Rossi and Brinkkemper's method complexity to compare specific versions of the CMMN, BPMN, EPC methods, and the UML AD technique.

4.2 Methodology

This study uses Rossi and Brinkkemper's [RB96] meta-model-based method complexity that counts meta-model objects and properties. The cumulative method complexity $C'(\mathcal{M})$ used in this approach is a vector in three-dimensional space, where the axes represent the count of objects $n(O_{\mathcal{M}})$, the count of relationships $n(R_{\mathcal{M}})$, and the count of properties $n(P_{\mathcal{M}})$. This allows for the comparison of the different vectors that correspond to the complexity $C'(\mathcal{M})$ of the methods being analyzed.

It is important to note the version number of the different specifications being analyzed because method complexity is likely to change from one version of a specification to the next. This research made use of the evaluations done by Recker et al. [Rec+09] and Siau and Cao [SC02] of BPMN version 1.2 and UML ADs version 1.4 as described above. The EPC evaluation done by Indulska et al. [Ind+09b] has also been used in this research. Their evaluation used the version of EPC that was defined by Nüttgens and Rump [NR02] and their calculations were based on the meta-model created by Becker et al. [Bec+03].

4.2.1 Meta-Model-Based Method Complexity

This section makes use of definitions used in the work done by Rossi and Brinkkemper [RB96]. Although they defined 17 method complexity metrics 12 of them were for techniques and the rest for the method itself. This research uses a subset of these method complexity metrics to compare the CMMNmethod complexity with the work done by Indulska et al. and Recker et al. Rossi and Brinkkemper formally defined the model of a technique in the Object, Property, Relationship, Role (OPRR) [RB96] modeling language as follows:

Definition 4.1. (Model of a technique): The model of a technique in the OPRR modeling language is a six-tuple

$$M_T = \langle O_T, P_T, R_T, X_T, r_T, p_T \rangle$$

where,

- O_T is a finite set of object types.
- P_T is a finite set of property types.
- R_T is a finite set of relationship types.
- X_T is a finite set of role types.
- r_T is a mapping $r_T : R_T \rightarrow \{x \mid x \in \mathfrak{P}(X_T \times (\mathfrak{P}(O_T) - \{O_T\})) \wedge n(x) \leq 2\}$, where $n(x)$ is the cardinality of x, and $\mathfrak{P}(O_T)$ is the power set of set O_T.
- p_T is a partial mapping $p_T : NP \rightarrow \mathfrak{P}(O_T)$, where $NP = \{O_T \cup R_T \cup X_T\}$ is the set of non-property types.

Definition 4.2. (Object types per method): $n(O_{\mathcal{M}})$ is the count of objects in the method, which corresponds to the count of objects in all of the techniques

$$n(O_{\mathcal{M}}) = \sum_{T \in \mathcal{M}} n(O_T)$$

Definition 4.3. (Relationship types per method): $n(R_{\mathcal{M}})$ is the count of relationships in the method, which corresponds to the count of objects in all of the techniques

$$n(R_{\mathcal{M}}) = \sum_{T \in \mathcal{M}} n(R_T)$$

Definition 4.4. (Property types per method): $n(P_{\mathcal{M}})$ is the count of properties in the method, which corresponds to the count of properties in all of the techniques

$$n(P_{\mathcal{M}}) = \sum_{T \in \mathcal{M}} n(P_T)$$

Definition 4.5. (Complexity of a technique): The complexity of the technique is defined as

$$C'(\mathcal{M}_T) = \sqrt{n(O_T)^2 + n(R_T)^2 + n(P_T)^2}$$

Definition 4.6. (Model of a method): A method is a set of techniques. Therefore, for Rossi and Brinkkemper [RB96] the model of a method $\mathcal{M}$ is

$$M_{\mathcal{M}} = \bigcup_{T \in \mathcal{M}} M_T$$

Definition 4.7. (Cumulative complexity of the method): The cumulative complexity of the method is defined as

$$C'(\mathcal{M}) = \sqrt{n(O_{\mathcal{M}})^2 + n(R_{\mathcal{M}})^2 + n(P_{\mathcal{M}})^2}$$

Note that by definition the complexity of a technique $C'(\mathcal{M}_T)$ can be compared to the cumulative complexity of a method $C'(\mathcal{M})$. This research compared the UML technique for Activity Diagrams with the methods for EPC, BPMN, and CMMN. UML AD was treated as a method. CMMN contains a single technique (i.e., case plan model), therefore $M_{\mathcal{M}} = M_T$, $n(O_{\mathcal{M}}) = n(O_T)$, $n(R_{\mathcal{M}}) = n(R_T)$, and $n(P_{\mathcal{M}}) = n(P_T)$.

4.2.1.1 Adjustments

In their work, Rossi and Brinkkemper [RB96] used the OPRR modeling language as implemented in MetaEdit [Smo+91] to model the methods to be analyzed. However, later studies conducted by Indulska et al. [Ind+09b] and Recker et al. [Rec+09] were based on UML meta-models instead of OPRR meta-models. Therefore, in order to produce results comparable with previous evaluations [Ind+09b; Rec+09; SC02] this study used Rossi and Brinkkemper's meta-model method complexity metrics, but explicitly identified and followed the approach used by Indulska et al. This allowed the researcher to use the normative CMMN meta-model described using UML class diagrams in the CMMN specification. Additionally, to avoid confusion and to be consistent with Rossi and Brinkkemper's approach, for the purpose of this study UML meta-model classes are referred to as objects, and attributes are referred to as properties.

There are some differences between the meta-model-based method complexity as defined by Rossi and Brinkkemper [RB96] and how it was applied by Indulska et al. [Ind+09b]. First, as noted above, Rossi and Brinkkemper used OPRR and Indulska et al. used UML. Indulska et al. developed a UML meta-model of BPMN version 1.2 for their research because the BPMN version 1.2 specification did not describe a normative UML meta-model. Secondly, Rossi and Brinkkemper described a set of 17 complexity metrics. Because BPMN version 1.2 contains a single technique (i.e., business process diagrams) Indulska et al. used a smaller subset that focused on the total cumulative method complexity of a method $C'(\mathcal{M})$. Thirdly, Indulska et al. introduced the concept of full and concrete notation for BPMN version 1.2. The full notation consists of the objects, relationships, and properties from the notation meta-model, and the concrete notation consists of the objects, relationships, and properties derived from the graphical notation. In accordance with Rossi and Brinkkemper, who used a simple, conceptual complexity to compare methods based on the meta-model, the researcher focused on the full notation in this study.

4.2.1.2 Counting Principles

After careful analysis of the meta-model and the approach used by Indulska et al. [Ind+09b], the following counting principles were identified and used for specifications described in UML meta-models:

1. The count of objects includes all of the abstract classes.
2. The count of properties excludes references to other classes.
3. The count of properties includes all objects and relationship properties.
4. The count of properties excludes tool-generated properties. The meta-model used by Indulska et al. was developed for their research, hence it did not include tool-generated properties.
5. Enumerations are not counted.

4.3 CMMN Analysis

Table 4.1 was created using the counting principles described above along with the resulting objects and their properties for CMMN version 1.0. The data was extracted from the CMMN version 1.0 specification. Figure 4.1 illustrates part of the CMMN class diagram using the counting principles. It includes `CMMNElement`, `Case`, `Role`, `Stage`, and `CaseParameter` as objects. It also includes properties `name` for `Case`, `name` for `Role`, and `description` for `CMMNElement`. The `Id` in `CMMNElement` is a tool-specific property because it should be generated by the implementing tool. Therefore, it does not appear in Table 4.1. In CMMN 11 tool-generated properties were identified and removed from the count. Examples of tool-generated properties in CMMN include `Id`, `exporter`, `exporterVersion`, and `expressionLanguage`, which are expected to be populated by the tool implementing the specification.

Table 4.2 provides the resulting relationships and their properties for CMMN version 1.0. The data was extracted from the CMMN version 1.0 specification. As described by the counting principles all of the CMMN meta-model classes are included, but tool-generated properties are excluded from the count.

Table 4.1: CMMN version 1.0 objects and properties

Objects $O_{\mathcal{M}}$	Properties $P_{\mathcal{M}}$
CMMNElement	description
Definitions	name
Import	location
CaseFileItemDefinition	definitionType
CaseFileItemDefinition	name
CaseFileItemDefinition	structureRef
Property	name
Property	type
Case	name
Role	name
CaseFile	
CaseFileItem	multiplicity
CaseFileItem	name
EventListener	
Milestone	
PlanItemDefinition	name
TimerEventListener	timerExpression
StartTrigger	
CaseFileItemStartTrigger	standardEvent
PlanItemStartTrigger	standardEvent
UserEventListener	
PlanFragment	
PlanItem	name
Sentry	name
IfPart	
Expression	body
Stage	autoComplete
TableItem	
DiscretionaryItem	
ApplicabilityRule	name
Task	isBlocking
ProcessParameter	
Parameter	name
ParameterMapping	
CaseParameter	
HumanTask	
ProcessTask	
process	
CaseTask	
PlanItemControl	
ManualActivationRule	name
RequiredRule	name
RepetitionRule	name

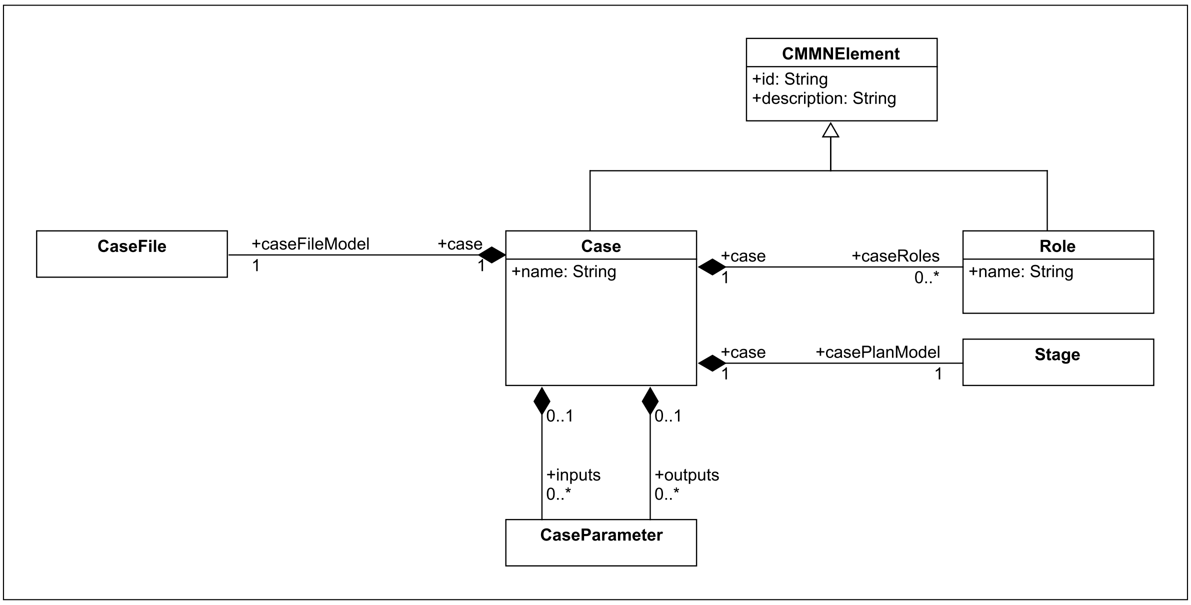


Figure 4.1: Case class diagram from CMMN version 1.0 specification [OMG14a]

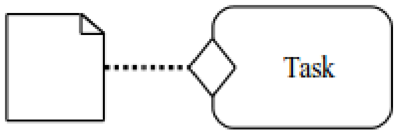


Figure 4.2: Event propagation notation between a case file item and a task

Relationships in CMMN are challenging in terms of this analysis because there is a single relationship connector (dashed line) in the CMMN notation, and its use is optional. In CMMN, the connector is only used in two situations. First, it is used optionally to indicate event propagation represented in the meta-model by the `OnPart` of which there are three classes: one abstract class and two concrete classes. Figure 4.2 shows an example of the event propagation notation between a case file item and a task. Secondly, the same connector (dashed line) is used for an expanded planning table in a human task. In this situation the connector is used to connect the human task to the discretionary items contained in the planning table. Figure 4.3 provides an example of an expanded planning table in a human task containing two discretionary tasks. However, planning tables can also be used in stages in which case the connector is never used. CMMN does not have an object in the meta-model to indicate the second situation. There is no object that represents the connection of an expanded table in a human task to its discretionary items. Table 4.2 counts the planning table as a relationship to account for this situation.

Having identified the appropriate set of objects, relationships, and properties using an approach similar to that used by Indulska et al. [Ind+09b] and by using the derived counting principles, Rossi and Brinkkemper's [RB96] method complexity calculations were applied by counting the cells in the tables. Based on Table 4.1, there are 39 non-duplicated object

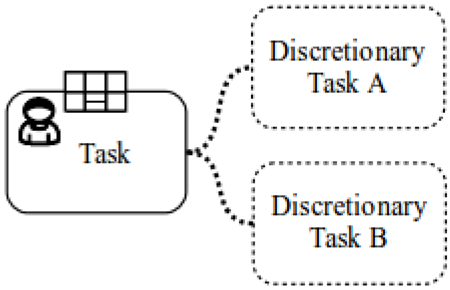


Figure 4.3: Planning table in a human task containing two discretionary tasks

Table 4.2: CMMN version 1.0 relationships and properties

Relationships $R_{\mathcal{M}}$	Properties $P_{\mathcal{M}}$
OnPart	
CaseFileItemOnPart	standardEvent
PlanItemOnPart	standardEvent
PlanningTable	

types $n(O_{\mathcal{M}})$ in the CMMN method. Based on Table 4.2, there are four non-duplicated relationship types $n(R_{\mathcal{M}})$. Based on Tables 4.1 and 4.2, there are 28 properties $n(P_{\mathcal{M}})$. Therefore, the calculated cumulative method complexity $C'(\mathcal{M})$ for CMMN version 1.0 is $C'(\mathcal{M}) = \sqrt{n(O_{\mathcal{M}})^2 + n(R_{\mathcal{M}})^2 + n(P_{\mathcal{M}})^2} = \sqrt{39^2 + 4^2 + 28^2} = 48.18$.

4.4 Findings

Table 4.3 shows the CMMN version 1.0 method complexity in the context of other popular process notations. The table is organized based on the cumulative method complexity $C'(\mathcal{M})$. The methods included were reported by Siau and Cao [SC02], and Indulska et al. [Ind+09b] using the BPMN version 1.2 subsets identified by zur Muehlen and Ho [zH08], zur Muehlen and Recker [zR08], and zur Muehlen et al. [zur+07]. For illustrative purposes, Figure 2.3 shows a simple insurance claim process described by Korherr [Kor08] using the four process modeling notations evaluated during this study (see Section 2.2.1 for more details).

4.4.1 Implications

A calculated, cumulative complexity of 48.18 for CMMN version 1.0 indicates that it is more complex than EPC, which has a cumulative complexity of 19.26. However, UML version 1.4 Activity Diagrams, which has a cumulative complexity of 11.18 is less complex than BPMN version 1.2, which has a cumulative complexity of 169.07. Table 4.3 clearly shows how BPMN version 1.2 makes extensive use of properties, relationships, and objects: more so than all of the other methods. As stated by Rossi and Brinkkemper [RB96], this may also indicate that

Table 4.3: Method complexity comparison

Method	Objects $n(O_{\mathcal{M}})$	Relationships $n(R_{\mathcal{M}})$	Properties $n(P_{\mathcal{M}})$	Cumulative Complexity $C'(\mathcal{M})$
BPMN 1.2^{FULL} [Ind+09b]	90	6	143	169.07
BPMN 1.2 DoD^{FULL} [Ind+09b; U.S09]	59	4	112	126.65
BPMN 1.2 Case Study^{FULL} [Ind+09b; zH08]	36	5	81	88.78
BPMN 1.2 Frequent Use^{FULL} [Ind+09b; zR08]	21	4	59	62.75
CMMN 1.0 [OMG14a]	39	4	28	48.18
EPC^{FULL} [Ind+09b]	15	5	11	19.26
UML 1.4 Activity Diagrams [SC02]	8	5	6	11.18

BPMN version 1.2 is more expressive than CMMN version 1.0, which in turn may be more expressive than EPC and UML version 1.4 Activity Diagrams.

The results are encouraging as they may indicate that CMMN should be easier to learn than BPMN. As suggested by Rossi and Brinkkemper [RB96], these results should be validated via empirical studies. Although empirical validation of the results is needed, practitioners who find BPMN difficult to use may want to explore CMMN as an alternative for Knowledge Intensive Processes (KiP) that follow the use cases identified by di Ciccio et al. [di +14], Işik et al. [I+13], Le Clair et al. [Le +09], Reijers et al. [Rei+03], Swenson [Swe10a], and van der Aalst and Berens [vB01].

The reliability and validity of the comparisons may be compromised by the mix of the meta-models and counting principles involved. The researcher was careful to follow the original approach described by Rossi and Brinkkemper [RB96], and adjusted it to compare the results with the work done by Indulska et al. [Ind+09b], Recker et al. [Rec+09], and Siau and Cao [SC02]. In the process, it was noted that Rossi and Brinkkemper [RB96] and Siau and Cao [SC02] used OPRR meta-models to compare their results, while Indulska et al. [Ind+09b] used an UML meta-model; and Recker et al. [Rec+09] used the two meta-models. This study used the normative UML meta-model from the specification was used for CMMN version 1.0.

4.5 Summary

This chapter has provided one of the first studies of method complexity of CMMN version 1.0 using the approach proposed by Rossi and Brinkkemper [RB96]. It compared the results against the results obtained by Siau and Cao [SC02], Indulska et al. [Ind+09b], and Recker et al. [Rec+09]. Based on the findings, CMMN compares favorably to other methods. Material from this chapter has been published in Marin et al. [Mar+14b].

Chapter 5

Transformations Between GSM and CMMN

This chapter contributes formal transformations between Case Management Model and Notation (CMMN) [OMG14a] case types and Guard-Stage-Milestone (GSM) [Hul+11b] artifact types that allow for theoretical results derived from GSM to be applied to CMMN. The transformation from a CMMN case type to a GSM artifact type may also provide CMMN with formal execution semantics. Not surprisingly, the transformation of a GSM artifact type into a CMMN case type is easier to achieve because CMMN is based on GSM [Mar+13; OMG14a]. The transformation from a GSM artifact type into a CMMN case type is relatively straight forward and simple to describe. The resulting case type modeled using CMMN is visually similar to the original artifact type, making it easier for a human being to understand. However, the transformation from a case type into an artifact type is more complex because CMMN has extended GSM by introducing new constructs and defining a life cycle with a set of standard events for those constructs. This transformation allows CMMN to use the formal operational semantics of GSM, and this means that the formal verification work developed for GSM can also be applied to CMMN.

This chapter is organized as follows. Section 5.1 describes the motivation for the transformations. Section 5.2 develops and describes the GSM to CMMN transformation. Section 5.3 develops and describes the CMMN to GSM transformation based on patterns. Two appendices complement the material in this chapter. Appendix D, file 30 (FSM-2-GSM.pdf) describes and formalizes the transformation of a Deterministic Finite State Machine (DFSM) into GSM types that are required to transform CMMN into GSM. Material from Appendix D file 30 (FSM-2-GSM.pdf) has been published in Marin et al. [Mar+16]. Appendix A describes the syntax directed translation grammar [Aho+07] used to transform a CMMN case type into a GSM artifact type.

Figure 5.1: Transformations

5.1 Motivation

The motivation for performing the transformations between CMMN and GSM centers on the possibility of using the theoretical results from GSM [Bel+12; Dam+11; Gon+15; Sol+13b] to provide formal execution semantics for CMMN. This thesis uses an approach similar to that used by Solomakhin et al. [Sol+12] to map GSM to Data-Centric Dynamic Systems (DCDS) [Bag+13; Cal+15]. Both GSM and DCDS are implementations of BAs. Figure 5.1 shows the transformations. The transformation of GSM artifact types to DCDS is described in Solomakhin et al. [Sol+12].

Several researchers of Business Artifact (BA) have done work on transformations. Solomakhin et al. [Sol+13b] have done a transformation of GSM into DCDS which is another BA framework [Bag+13]. Meyer and Weske [MW13] sketched algorithms to transform artifact-centric process models into activity-centric process models. They identified four types of process models, namely artifact-centric, synchronized object life cycles, activity-centric, and activity-centric with attribute definitions. They also sketched five algorithms that allow any of the four types of process models to be transformed into one another. Popova and Dumas [PD13] and Popova et al. [Pop+15] worked on the automatic transformation of Petri Nets into GSM artifact types.

Transformations between process modeling notations particularly between Business Process Model and Notation (BPMN) [OMG13] and Business Process Execution Language (BPEL) [OAS07] have been described in the work of several researchers [MH11; Shi+16]. Transformations between BPEL and BPMN are important because BPEL does not have a graphical notation, and early BPMN versions did not have clear execution semantics. A transformation can be used to address these two issues. The CMMN specification [OMG14a] and some authors, such as Bruno [Bru16], Eshuis et al. [Esh+16], and Jansen [Jan15], have described CMMN as being based on the GSM type of BA. However, the researcher is not aware of any publication that describes the formal relationship between CMMN and GSM. This chapter addresses this gap by proposing two transformations, one from GSM to CMMN, and the other from CMMN to GSM. These transformations will help clarify the relationship between CMMN and GSM, and provide a way to apply the theoretical work that has been done on GSM to CMMN models.

5.2 Transforming an Artifact Type into a Case Type

This section presents a conversion from an artifact type Γ (see Definition 3.1) into a Case type $\mathcal{T}$ (see Definition 3.4). While the syntactical transformation between an artifact type Γ into a Case type $\mathcal{T}$ is achievable CMMN does not have the concepts of snapshot and Business step (B-step) that formally describe the GSM operational semantics. CMMN execution semantics are event-driven, described by rules (i.e., stage autocomplete, manual activation, required, repetition, and applicability rules) and by the entities' life cycle (see Figures 5.4, 5.6, 5.8 and 5.10) [OMG14a]. Implementation details are left to the vendors, in particular the order in which events are processed is not defined. Therefore, different vendors may implement CMMN's execution semantics slightly differently, and so there is no guarantee that a converted artifact type Γ will execute in CMMN in the same way that it did in GSM.

The transformation of an artifact type into a case type assumes that the target CMMN engine supports Object Constraint Language (OCL) [OMG14b] as an expression language, and allows expressions to access the state of the CMMN objects' life cycle. The CMMN specification allows implementations to support multiple expression languages. It is therefore natural to continue using OCL during the transformation, which is the only expression language supported by GSM.

Definition 5.1. (From an Artifact type Γ to a CMMN case type $\mathcal{T}$): Given an artifact type $\Gamma = \langle\ Att,\ EType,\ Stg,\ Mst,\ Stry,\ Lcyc\ \rangle$ with $Lcyc = \langle\ Substages,\ Tasks,\ Submilestones,$ $Guards,\ Terminators,\ Ach,\ Inv\ \rangle$. The corresponding case type $\mathcal{T}$ will have the following form,

$$\mathcal{T}_\Gamma = \langle \mathcal{D}_\Gamma, \mathcal{B}_\Gamma \rangle$$

Where,

$$\mathcal{D}_\Gamma\ = \{d \mid d \in Att_{data}\} \cup \{m_d \mid mExpr(m)\}$$

$$\mathcal{B}_\Gamma\ = \langle\ \mathcal{S}\boldsymbol{t}_\Gamma,\ \mathcal{T}\boldsymbol{a}_\Gamma,\ \mathcal{M}\boldsymbol{i}_\Gamma,\ \mathcal{E}\boldsymbol{v}_\Gamma,\ \mathcal{H}_\Gamma,\ \mathcal{R}_\Gamma \rangle$$

$$\mathcal{S}\boldsymbol{t}_\Gamma\ = \{s \mid s \in Stg\}$$

$$\begin{aligned}\mathcal{T}\boldsymbol{a}_\Gamma\ = &\{t \mid (\exists S)(S \in Stg\ \wedge\ t = Tasks(S))\}\\ &\cup\{m_i \mid mExpr(m)\}\\ &\cup\{m_a \mid mExpr(m)\}\\ &\cup\{e_i \mid e \in EType_{inc}\ \wedge\ Payload(e)\}\end{aligned}$$

$$\mathcal{M}\boldsymbol{i}_\Gamma\ = \{m \mid m \in Mst\}$$

$$\mathcal{Ev}_\Gamma = \{e \mid e \in EType_{inc} \ \wedge \ noPayload(e)\}$$

$$\mathcal{H}_\Gamma = \{\langle x, y\rangle \mid x \in Stg \ \wedge \ (y = Substages(x) \ \vee \ y = Submilestones(x) \ \vee \ y = Tasks(x))\}$$

$$\cup\{\langle case, y\rangle \mid topLevelStage(y)\}$$

$$\cup\{\langle case, e_i\rangle \mid e \in EType_{inc} \ \wedge \ Payload(e)\}$$

$$\cup\{\langle x, m_i\rangle \mid x \in Stg \ \wedge \ mExpr(m) \ \wedge \ m = Submilestones(x)\}$$

$$\cup\{\langle x, m_a\rangle \mid x \in Stg \ \wedge \ mExpr(m) \ \wedge \ m = Submilestones(x)\}$$

$$\mathcal{R}_\Gamma = \langle \widehat{\varphi}, \widehat{\mathscr{S}}, \widehat{\mathscr{E}}, \widehat{\mathscr{X}}, \widehat{\mathscr{M}}, \widehat{\mathscr{R}}, \widehat{\mathscr{N}}, \widehat{\mathscr{A}}, \rangle$$

$$\widehat{\varphi} = \{c \mid (\exists y)(y \in Stry \ \wedge \ c = ifconvert(y))\}$$

The Boolean condition in the artifact type sentries is converted to case type Boolean conditions.

$$\widehat{\mathscr{S}} = \{s \mid (\exists y)(y \in Stry \ \wedge \ s = convert(y))\}$$

The set of artifact type sentries is converted to case type sentries.

$$\widehat{\mathscr{E}} = \{\langle x, y\rangle \mid y \in Stg \ \wedge \ x = convert(Guards(y))$$

$$\vee \ y \in Tasks \ \wedge \ x = convert(Guards(y))$$

$$\vee \ y \in Mst \ \wedge \ mExpr(m) \ \wedge \ x = convert(Ach(y))\}$$

$$\cup \ \{\langle x, m_i\rangle \mid mExpr(m) \ \wedge \ x = convert(Inv(m))\}$$

$$\cup \ \{\langle x, m_a\rangle \mid mExpr(m) \ \wedge \ x = convert(Ach(m))\}$$

The case type entry criteria relationship is populated with the guards for stages, tasks, and milestones. In addition, the generated tasks used to update the milestone status get the corresponding achieved and invalidated guards.

$$\widehat{\mathscr{X}} = \{\langle x, y\rangle \mid y \in Stg \ \wedge \ x = convert(Terminators(y))\}$$

The exit criteria relationship is populated with the artifact type sentries for terminators.

$$\widehat{\mathscr{M}} = \varnothing$$

GSM does not support the concept of manual activation.

$$\widehat{\mathscr{R}} = \varnothing$$

GSM does not support the concept of required.

$$\widehat{\mathscr{N}} = \{\langle true, y\rangle \mid y \in \mathcal{St} \cup \mathcal{Ta} \cup \mathcal{Mi}\}$$

In GSM all of the stages, tasks and milestones are repeatable.

$$\widehat{\mathscr{A}} = \{a \mid a \in \mathcal{St}\} \cup \{case\}$$

In GSM all of the stages are automatically terminated when the terminator is achieved, and no manual intervention is needed.

m_d is a Boolean property for milestone $m \in Mst$. This property is only created for milestones that can be invalidated in the artifact type, and allows the milestone to maintain its status.

m_i is a task that sets the value of m_d to false, when the milestone is invalidated.

m_a is a task that sets the value of m_d to true, when the milestone is achieved.

mExpr(x) is a predicate that is true if x is a milestone ($x \in Mst$) and has both an achieving and invalidating sentry $((\exists g, j)(g = Inv(m) \wedge j = Ach(m)))$, and the value of x is used in a sentry condition expression φ.

Payload(x) is a predicate that is true if the event x has a payload.

noPayload(x) is a predicate that is true if the event x does not have a payload.

topLevelStage(x) is a predicate that is true for $x \in Stg$ when x is a *top level stage* $(\nexists y)(x = Substages(y))$.

convert(x) is a function that uses *onconvert*(x) and *ifconvert*(x) to convert a GSM sentry into a CMMN sentry.

ifconvert(x) is a function that converts the *condition expression* of a GSM sentry φ into a valid CMMN *condition expression* by replacing all of the references to Att_{stages} and to $Att_{milestones}$ with a reference to the correct state as shown in Figures 5.8 and 5.10. In the case of milestones that test for $\neg m$, the generated m_d property is used.

replaces references to S ($S \in Att_{stages}$) with $S.Active$.

replaces references to $\neg S$ ($S \in Att_{stages}$) with $\neg S.Active$.

replaces references to m ($m \in Att_{milestones}$) with $m.Completed$.

replaces references to $\neg m$ ($m \in Att_{milestones}$) with $\neg m_d$.

add $\neg m_d$ for milestones that test for $-m$ in the ξ expression.

onconvert(x) is a function that converts the *event expression* of a GSM sentry ξ into a valid CMMN *event expression* $\dot{\xi}$ by replacing all of the generated events with state transitions as shown in Figures 5.8 and 5.10. In the case of milestones that test for $-m$, the generated m_d property is used.

replaces $+m$ (milestone m is triggered) with <*occur*>:m.

replaces $-m$ (milestone m is triggered) with <*update*>:m_d.

replaces $+S$ (stage S open) with <*start*>:S.

replaces $-S$ (stage S closes) with <*complete*>:S.

replaces I:t (task t is invoked) with <*start*>:t.

replaces C:t (task t completes) with <*complete*>:t.

replaces E:e (incoming event e with no payload arrives) with <*occur*>:e.

replaces E:e (incoming event e with payload arrives) with <*complete*>:e_i.

The notion of invalidating a milestone does not exist in CMMN. Therefore, to deal with milestones for which there is an invalidation sentry in the artifact type ($m \in Mst$ such that $(\exists g)(g = Inv(m))$ a Boolean property (m_d) is created to maintain the state of the milestone, and two tasks (m_a and m_i) are used to set the Boolean property to true when the milestone is achieved, and to false when it is invalidated. The entry criteria for these two tasks are the same sentries used by the milestone for achieving ($Ach(m)$) and invalidating ($Inv(m)$) it. The $onconvert(x)$ function makes use of the milestone Boolean property as required to convert an expression that makes use of the milestone status ($Att_{milestone}$).

CMMN does not have the notion of an incoming event with a payload. Therefore, a map of incoming events without payloads to event listeners is created. In CMMN incoming events with a payload need to be processed by a task. Therefore, a map of incoming events with payloads to tasks e_i without entry criteria is created to listen to the incoming events. The $onconvert(x)$ function maps incoming event expressions to a CMMN standard event on an event listener for events without payloads, or to standard events of the tasks listening for the incoming message.

5.2.1 Example

Using the example taken from Eshuis et al. [Esh+13], described in Section 3.2.2 and presented in Figure 3.3, this researcher converted the GSM artifact type into a CMMN case type. The resulting case type is depicted in Figure 5.2.

All five milestones present in the original artifact type have invalidating sentries (see Table 3.2) and these are used in other sentries' condition expressions φ. Therefore, new m_a and m_i tasks were generated for each milestone, as well as new Boolean attributes m_d (see rule 6 in Table 3.1, and rules 21 and 22 in Table 3.2). There is one incoming event (`Regulation Change` in rule 24 in Table 3.2) without a payload, which is converted into an event listener. All of the resulting entry and exit criteria for the deal case type presented in Figure 5.2 are shown in Table 5.1.

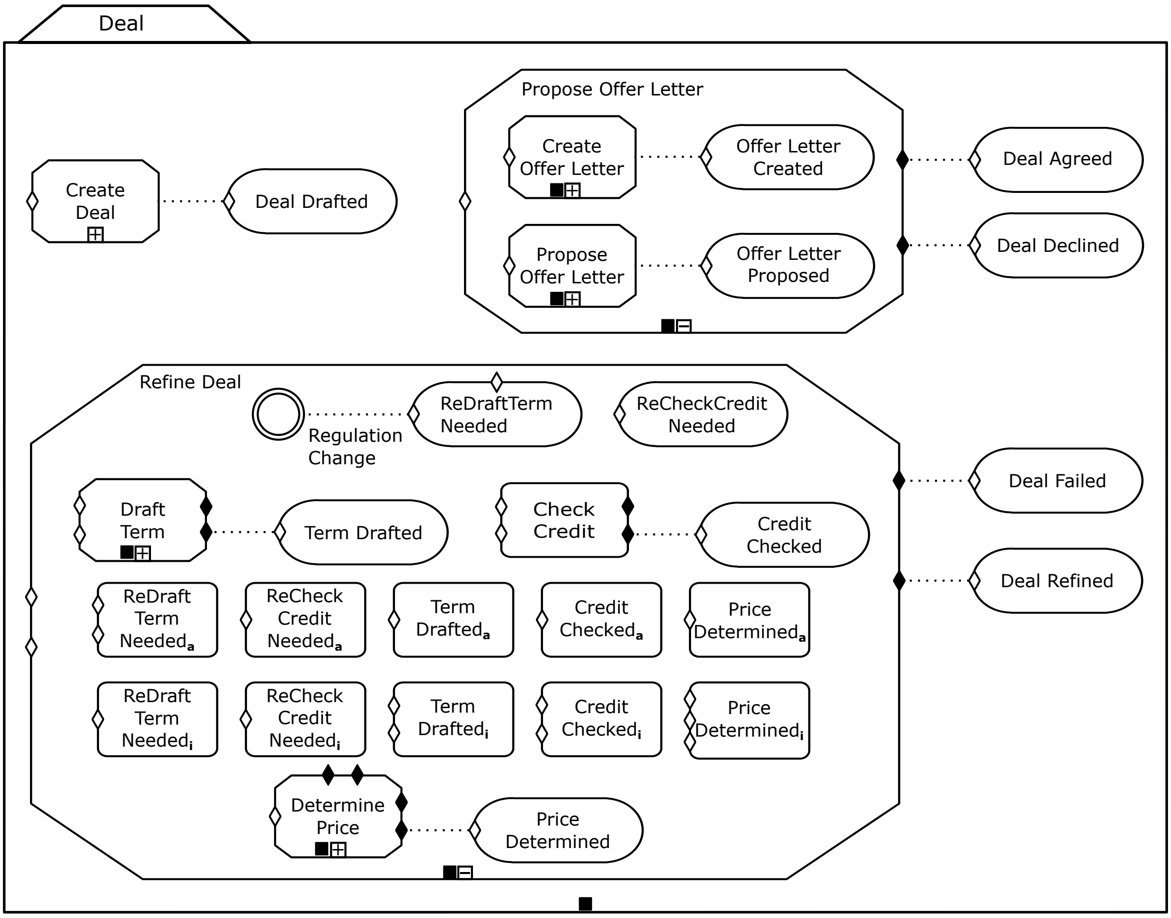


Figure 5.2: The deal case model in CMMN
The IBM Global Financing (IGF) example from Figure 3.3 transformed into CMMN.

Table 5.1: Entry and exit criteria for the Refine Deal stage in Figure 5.2

	Entry criteria	**Exit criteria**
Stages		
Draft Term	r_1: [**on** *start*:Refine Deal]	r_3: [**on** *occur*:Term Drafted]
	r_2: [**on** *occur*:ReDraftTermNeeded **if** Refine Deal·*Active*]	r_4: [**on** *complete*:Refine Deal]
Determine Price	r_5: [**if** Term Drafted$_d$ ∧ Credit Checked$_d$ ∧ Refine Deal·*Active*]	r_6: [**on** *occur*:Price Determined]
		r_7: [**on** *start*:Draft Term]
		r_8: [**on** *start*:Check Credit]
		r_9: [**on** *complete*:Refine Deal]
Tasks		
Check Credit	r_{10}: [**on** *start*:Refine Deal]	r_{12}: [**on** *occur*:Credit Checked]
	r_{11}: [**on** *occur*:ReCheckCreditNeeded **if** Refine Deal·*Active*]	r_{13}: [**on** *complete*:Refine Deal]
Term Drafted$_a$	r_{14}: [**on** *complete*:Draft Term **if** Refine Deal·*Active*]	

Continued on next page

Table 5.1 – *Continued from previous page*

	Entry criteria	**Exit criteria**
Credit Checked$_a$	r_{15}: [**on** *complete*:Check Credit **if** Refine Deal·*Active*]	
Price Determined$_a$	r_{16}: [**on** *complete*:Determine Price **if** Refine Deal·*Active*]	
ReDraftTermNeeded$_a$	r_{17}: [**on** *occur*:Regulation Change **if** Refine Deal·*Active*]	
	r_{18}: [**if** credit_level $> 100,000$ $\wedge$ Refine Deal·*Active*]	
ReCheckCreditNeeded$_a$	r_{19}: [**if** risk_level > 4 $\wedge$ Refine Deal·*Active*]	
Term Drafted$_i$	r_{20}: [**on** *start*:Draft Term]	
	r_{21}: [**on** *occur*:ReDraftTermNeeded]	
Credit Checked$_i$	r_{22}: [**on** *start*:Check Credit]	
	r_{23}: [**on** *occur*:ReCheckCreditNeeded]	
Price Determined$_i$	r_{24}: [**on** *start*:Draft Term **if** ReDraftTermNeeded$_d$]	
	r_{25}: [**on** *start*:Check Credit **if** ReCheckCreditNeeded$_d$]	
	r_{26}: [**on** *start*:Determine Price]	
ReDraftTermNeeded$_i$	r_{27}: [**on** *occur*:Term Drafted]	
ReCheckCreditNeeded$_i$	r_{28}: [**on** *occur*:Credit Checked]	
Milestones		
Term Drafted	r_{14}: [**on** *complete*:Draft Term **if** Refine Deal·*Active*]	
Credit Checked	r_{15}: [**on** *complete*:Check Credit **if** Refine Deal·*Active*]	
Price Determined	r_{16}: [**on** *complete*:Determine Price **if** Refine Deal·*Active*]	
ReDraftTermNeeded	r_{17}: [**on** *occur*:Regulation Change **if** Refine Deal·*Active*]	
	r_{18}: [**if** credit_level $> 100,000$ $\wedge$ Refine Deal·*Active*]	
ReCheckCreditNeeded	r_{19}: [**if** risk_level > 4 $\wedge$ Refine Deal·*Active*]	

5.3 Transforming a Case Type into an Artifact Type

Transforming a Case type $\mathcal{T}$ into an artifact type Γ is more difficult because, as discussed in Section 3.4, CMMN is more complex than GSM. CMMN entities are objects with a DFSM based life cycle, which is quite different from the simpler Boolean status in GSM. The strategy used to transform a case type $\mathcal{T}$ into an artifact type Γ involves identifying patterns in GSM in order to implement case type functionality that is not readily translated into an artifact type. The mechanics of the transformation are as follows:

1. Start by defining a basic constrained transformation for very simple case types. This provides an initial approximation and helps with identifying the areas that require further work.

2. Identify patterns to lift the constraints:

   (a) Patterns to implement the CMMN object life cycles.

   (b) Patterns based on the case type rules that will alter the object life cycle patterns.

   (c) Miscellaneous patterns and functions that transform expressions and sentries.

3. Develop an algorithm that applies the patterns in the right order.

Definition 5.2. (From a constrained case type $\mathcal{T}$ to an artifact type Γ): Given a case type $\mathcal{T}$ $= \langle\ \mathcal{D},\ \mathcal{B}\ \rangle$, with $\mathcal{B} = \langle\ \mathcal{St},\ \mathcal{Ta},\ \mathcal{Mi},\ \mathcal{Ev},\ \mathcal{H},\ \mathcal{R}\ \rangle$ and $\mathcal{R} = \langle\ \widehat{\varphi},\ \widehat{\mathscr{S}},\ \widehat{\mathscr{E}},\ \widehat{\mathscr{X}},\ \widehat{\mathscr{M}},\ \widehat{\mathscr{R}},\ \widehat{\mathscr{N}}\ \rangle$, with the following constraints:

- No data containers ($\mathcal{D}_{container} = \varnothing$).
- No expressions (event expressions or conditions) that refer to the case file ($\mathcal{D}_{casefile}$).
- No event expressions or conditions referring to the data life cycle (Figure 5.6).
- No event listeners ($\mathcal{Ev} = \varnothing$).
- No discretionary items ($\mathcal{St}_{discretionary} = \mathcal{Ta}_{discretionary} = \mathcal{St}_{fragment} = \varnothing$).
- Event expressions limited to a single event. The events must be restricted to $<\!occur\!>:m$ (milestone m is triggered), $<\!start\!>:S$ (stage S starts), $<\!complete\!>:S$ or $<\!terminate\!>:S$ (stage S completed), $<\!start\!>:t$ (task t starts), and $<\!complete\!>:t$ or $<\!terminate\!>:t$ (task t completes).
- No manual activation rules ($\widehat{\mathscr{M}} = \varnothing$), every stage must have autocomplete rules ($\widehat{\mathscr{A}} = \{a \mid a \in \mathcal{St}_{case} \cup \mathcal{St}_{planned}\}$), no required rules ($\widehat{\mathscr{R}} = \varnothing$), and no repetition rules ($\widehat{\mathscr{N}} = \varnothing$).
- No human interaction is expected in the resulting case.
- Sets $\mathcal{D}$, $\mathcal{St}$, $\mathcal{Ta}$, and $\mathcal{Mi}$ must be pairwise disjoint.

The corresponding artifact type $\Gamma_{\mathcal{T}}$ will have the following form,

$$\Gamma_{\mathcal{T}} = \langle Att_{\mathcal{T}}, EType_{\mathcal{T}}, Stg_{\mathcal{T}}, Tsk_{\mathcal{T}}, Mst_{\mathcal{T}}, Stry_{\mathcal{T}}, Lcyc_{\mathcal{T}} \rangle$$

Where,

$\boldsymbol{Att}_{\mathfrak{J}} = Att_{data} \cup Att_{stages} \cup Att_{milestones}$

$\boldsymbol{Att}_{data} = \{a \mid a \in \mathcal{D}_{discrete}\}$

$\boldsymbol{Att}_{stages} = \{s \mid s \text{ is Boolean } \wedge (\exists w)(s = \text{name}(w) \wedge w \in \mathcal{S}t \cup \mathcal{T}a)\}$

$\boldsymbol{Att}_{milestones} = \{m \mid m \text{ is Boolean } \wedge (\exists w)(m = \text{name}(w) \wedge w \in \mathcal{M}i)\}$

$\boldsymbol{EType}_{\mathfrak{J}} = EType_{inc} \cup EType_{gen}$

$\boldsymbol{EType}_{inc} = \varnothing$

$\boldsymbol{EType}_{gen} = \{e \mid (\exists y)(y \in \mathcal{S}t \cup \mathcal{T}a \cup \mathcal{M}i \wedge (e = open(y) \vee e = close(y)$
$\vee\, e = terminate(y)))\}$

$\boldsymbol{Stg}_{\mathfrak{J}} = \{s \mid (\exists z)(s = \text{name}(z) \wedge z \in \mathcal{S}t_{case} \vee z \in \mathcal{S}t_{planned})\}$

$\boldsymbol{Tsk}_{\mathfrak{J}} = \{t \mid t \in \mathcal{T}a\}$

$\boldsymbol{Mst}_{\mathfrak{J}} = \{m \mid (\exists z)(m = \text{name}(z) \wedge z \in \mathcal{M}i)\}$

$\boldsymbol{Stry}_{\mathfrak{J}} = \{s \mid s \in \widehat{\mathscr{S}}\}$

$\boldsymbol{Lcyc}_{\mathfrak{J}} = \langle Substages, Tasks, Submilestones, Guards, Terminators, Ach, Inv\rangle$

Substages is defined by $Substages^R = \{\langle x,y\rangle \mid \langle x,y\rangle \in \mathcal{H} \wedge y \in \mathcal{S}t \cup \mathcal{T}a\}$

Tasks is defined by $Tasks^R = \{\langle x,y\rangle \mid \langle x,y\rangle \in \mathcal{H} \wedge y \in \mathcal{T}a\}$

Submilestones is defined by $Submilestones^R = \{\langle x,y\rangle \mid \langle x,y\rangle \in \mathcal{H} \wedge y \in \mathcal{M}i\}$

Guards is defined by $Guards^R = \{\langle x,y\rangle \mid \langle x,y\rangle \in \widehat{\mathscr{E}} \wedge y \in \mathcal{S}t \cup \mathcal{T}a\}$

Terminators is defined by $Terminators^R = \{\langle x,y\rangle \mid \langle x,y\rangle \in \widehat{\mathscr{X}}\}$

Ach is defined by $Ach^R = \{\langle x,y\rangle \mid \langle x,y\rangle \in \widehat{\mathscr{E}} \wedge y \in \mathcal{M}i\}$

Inv is defined by $Inv^R = \varnothing$

$name(x)$ is a function that returns the name of the element.

$open(x)$ is a function that converts the following CMMN events into GSM events.

Replaces <*occur*>:m with $+m$ (milestone m is triggered).

Replaces <*start*>:S with $+S$ (stage S opened).

Replaces <*start*>:t with I:t (task t is invoked).

$close(x)$ is a function that converts the following CMMN events into GSM events.

Replaces <*complete*>:S with $-S$ (stage S closes).

Replaces <*complete*>:*t* with C:t (task *t* completes).

terminate(x) is a function that converts the following CMMN events into GSM events.

Replaces <*terminate*>:*S* with $-S$ (stage *S* closes).

Replaces <*terminate*>:*t* with C:t (task *t* completes).

There are three reasons for these constraints in this section:

1. CMMN uses DFSM to describe the life cycle of its entities and data. This introduces a mismatch between the generated event types in GSM and the CMMN DFSM's generated events.
2. There is native functionality in CMMN that is not present in GSM.
3. CMMN allows for the same stage or task to execute in parallel when using the repetition rule. GSM cannot handle parallel executions because of the Boolean status associated with the stage.

5.3.1 Patterns

This section shows how to remove the case type constraints listed in the previous section (Definition 5.2). This section presents patterns as a mechanism to transform a case type into an artifact type. A pattern in this context is a GSM implementation of CMMN functionality done in a generic way that identifies points in the implementation where the original CMMN entity or expression should be inserted. A pattern in this sense is similar to a macro in a programming language that can be expanded to include components or expressions.

Figure 5.3 illustrates the legend used to describe the patterns. The traditional GSM notation is used with only a few exceptions shown in the last column of the figure. The exceptions include the use of a dot ($\cdot$) to indicate manual input, the use of an asterisk ($*$) to indicate the place in the pattern where the CMMN logic should be added, and three dots placed in a horizontal or a vertical line to indicate repetition ($\ldots$).

Manual input is used to indicate events that will be raised by case workers. The introduction of the manual input is needed because GSM does not target knowledge workers, and so it does not provide constructs for knowledge workers to interact with and thereby affect the behavior of the BA. GSM leaves the implementation of human interaction to the developer. However, CMMN accounts for case workers' interactions with the case type by providing event transitions in the life cycle of entities that are generated through the manual intervention of a case worker.

The first set of patterns is based on the entities' life cycle as these are described in the CMMN specification. Although the life cycle of CMMN entities is a DFSM, the CMMN specification uses statecharts [Har87] to describe them. Here, DFSM is used when referring to the life cycle of CMMN entities and statecharts when referring to the depiction of the life cycle in the CMMN specification. Appendix D, file 30 (FSM-2-GSM.pdf) demonstrates the transformation of a DFSM into a GSM type, which will be used for the patterns presented in this section.

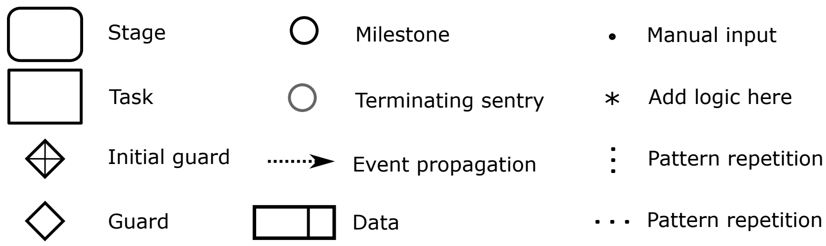


Figure 5.3: Legend used in GSM diagrams

5.3.1.1 Case Pattern

In CMMN the case itself is a stage with a particular life cycle. The case life cycle is used to represent the state of the case and to allow case workers to manipulate that state. The case life cycle is implemented as an artifact type pattern that encapsulates the rest of the case. In GSM the stages *Stg* are organized into $Substages^R$ to create a forest with the potential for multiple roots or top-level stages. However, in a case type there is a single root stage $\mathcal{S}t_{case}$, therefore the resulting artifact type Γ will have a single root stage.

Figure 5.4 presents the statechart demonstrating the life cycle of the case and Figure 5.5 illustrates the resulting pattern. The $\Gamma^{case}_{pattern}(c)$ function takes the single case element $c \in \mathcal{S}t_{case}$ and applies the pattern. The expanded case type will be placed inside the Active stage replacing the asterisk ($*$).

5.3.1.2 Data Pattern

Data in CMMN has a predefined life cycle, as depicted in Figure 5.6, that guarantees a set of events from each data attribute and describes the status of the data attribute. The status of data attributes can be used in condition expressions, and the generated events can be used in event expressions. GSM does not have the concept of data attributes generating events or carrying a status. Additionally, CMMN data can be unstructured (e.g., documents, spreadsheets, pictures, video clips, voice recordings, etc.), while GSM only supports relational data attributes.

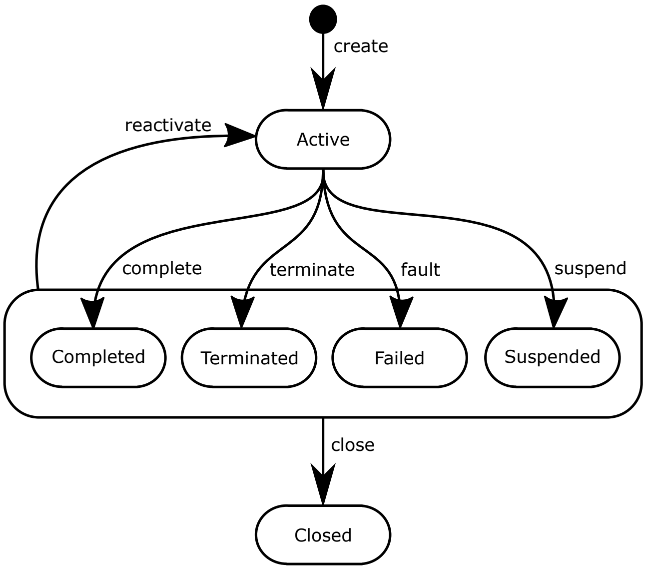


Figure 5.4: Statechart [Har87] depicting the life cycle of a case instance [OMG14a] ($\mathcal{S}t_{case}$)

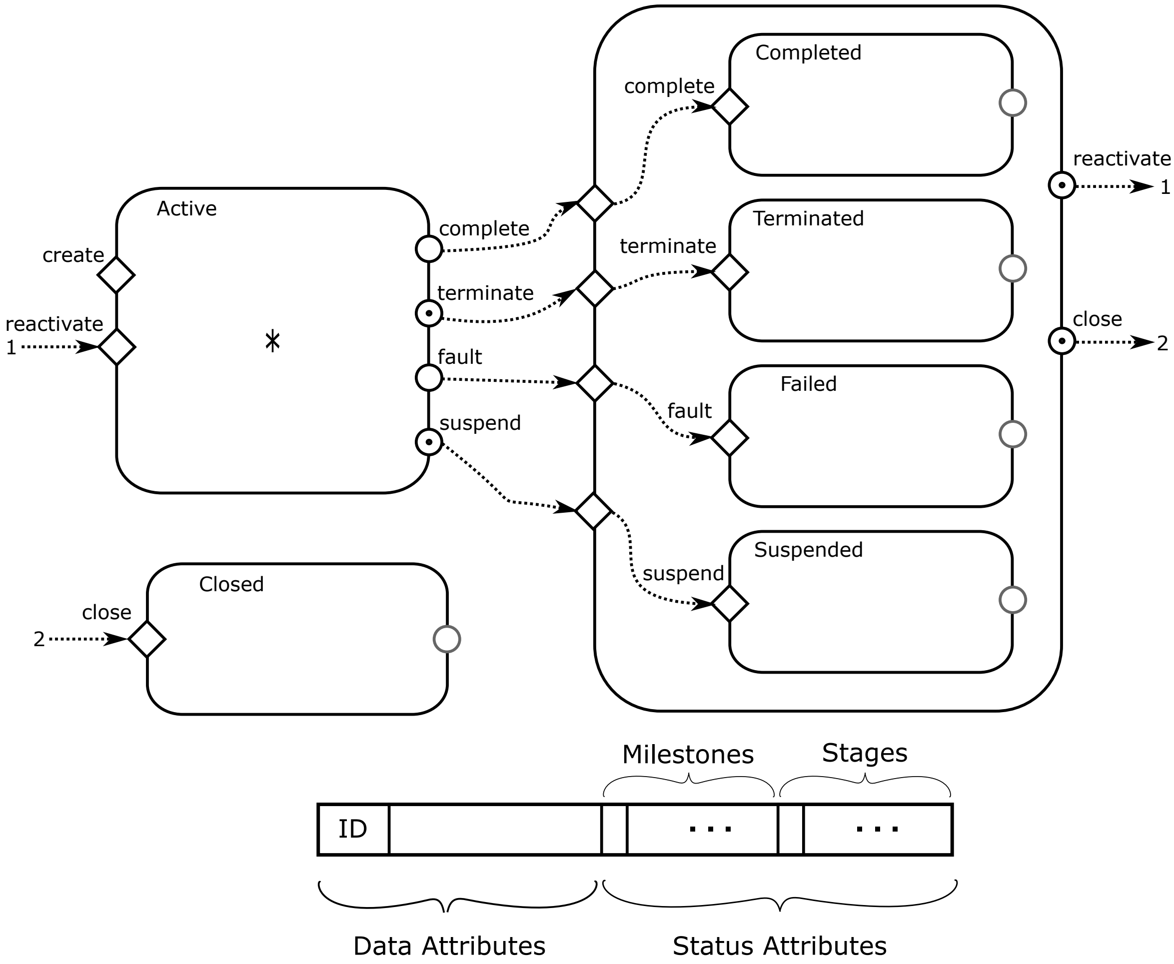


Figure 5.5: GSM pattern for the life cycle of a case instance ($\mathcal{S}t_{case}$)
GSM version of the statechart depicted in Figure 5.4 implementing $\Gamma^{case}_{pattern}$

In order to preserve the CMMN data behavior and types, the pattern implements the data life cycle of the DFSM depicted in Figure 5.6 with an artifact type that enforces all data manipulation via GSM events. This approach of isolating the underlying data storage by enforcing a messaging interface, supports both structured data in a relational database, and unstructured data in a content management system.

Figure 5.6 depicts the statechart describing the life cycle of data in CMMN, while Figure 5.7 presents the resulting pattern. The $\Gamma^{data}_{pattern}(d)$ function takes a data attribute $d \in \mathcal{D}$ and creates an artifact type for it.

The resulting pattern presented in Figure 5.7 has two available states (i.e., `Available` and `Available'`) with each having an almost identical set of tasks. The tasks with an apostrophe in their names implement exactly the same functionality as the tasks without the apostrophe. The reason for the duplication is that state `Available`, shown in Figure 5.6, has state transitions from it to itself. This is problematic in GSM because the operation's semantics are based on B-steps following the Prerequisite-Antecedent-Consequent (PAC) rules, and a transition into itself will violate the toggle once principle as described in Section 3.2.1.1. Therefore, it is invalid in GSM for the stage `Available` to close (terminate) and immediately open (execute) again inside the same B-step. To comply with the toggle once principle the pattern duplicates the `Available` state and, to avoid name collisions, appends apostrophes to the duplicated names.

This pattern also includes tasks that implement each operation in the data attribute. Each incoming event must contain the data attribute in the payload, and the task corresponding to the transition should implement the correct behavior using the data attribute.

For each data attribute in the case type a copy of this pattern will be created resulting in a new artifact type. This forces the transformation to fix all of the expressions that make use of the data element.

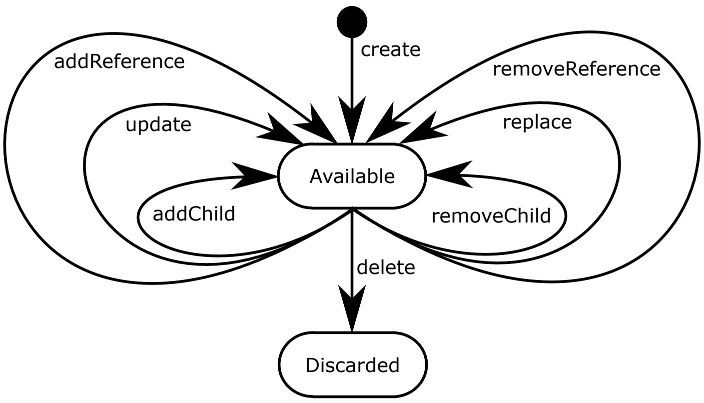


Figure 5.6: Statechart depicting the life cycle of a CMMN data element [OMG14a] ($x \in \mathcal{D}$)

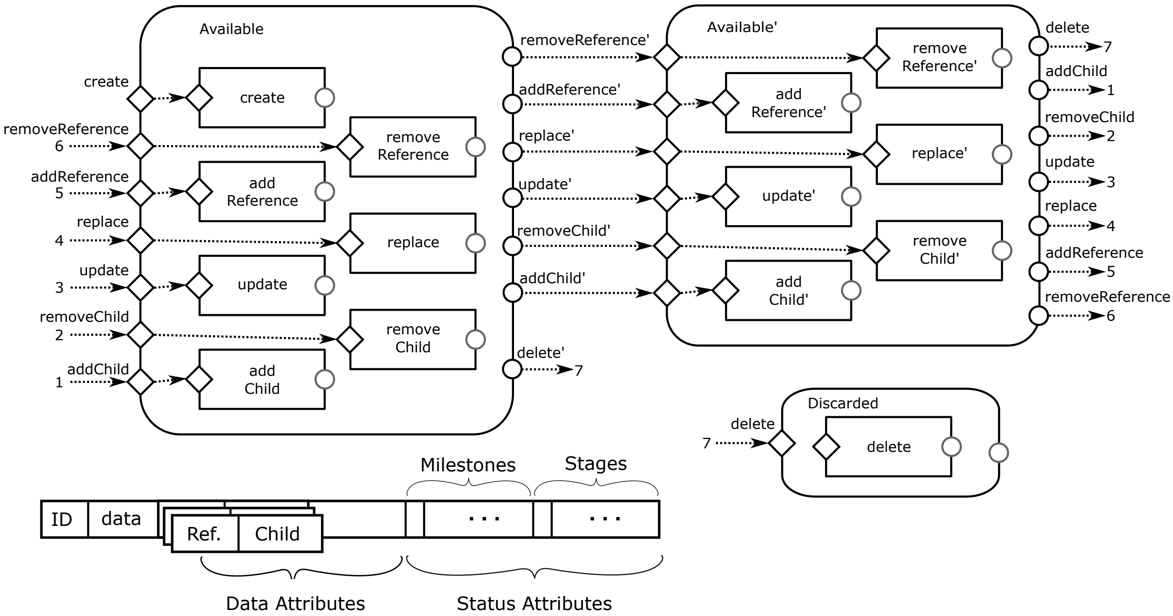


Figure 5.7: GSM pattern for data life cycle ($\mathcal{D}$) in a case instance
GSM version of the statechart depicted in Figure 5.6 implementing $\Gamma^{data}_{pattern}(d)$

5.3.1.3 Stages and Tasks Patterns

Stages and tasks share the same life cycle in CMMN. Figure 5.8 illustrates the life cycle of stages and tasks in CMMN and Figure 5.9 depicts the resulting GSM pattern. There are two functions that assist this pattern with dealing with idiosyncrasies in the transformation. The $\Gamma^{stage}_{pattern}(S)$ function takes $S \in \mathcal{St}_{planned} \cup \mathcal{St}_{discretionary}$ and applies the pattern. The $\Gamma^{task}_{pattern}(t))$ function takes $t \in \mathcal{Ta}$ and applies the pattern.

5.3.1.4 Milestone and Event Listener Patterns

Event listeners and milestones in CMMN have their own life cycles, which in turn generate events that can be used in sentries. The CMMN events themselves are very simple in comparison to GSM events because they lack a payload. Milestones in CMMN are similar to GSM milestone, with the following exceptions:

- they have a life cycle describing their state and emitting transition events.
- they can have multiple associated sentries (entry criteria).
- they can be marked as repeatable or not.
- they cannot be invalidated.

Figure 5.10 illustrates the life cycle of milestones and event listeners in CMMN. Figure 5.11 depicts the corresponding GSM pattern in which the event listener or milestone is implemented as a GSM milestone. There are two functions that assist this pattern with dealing with

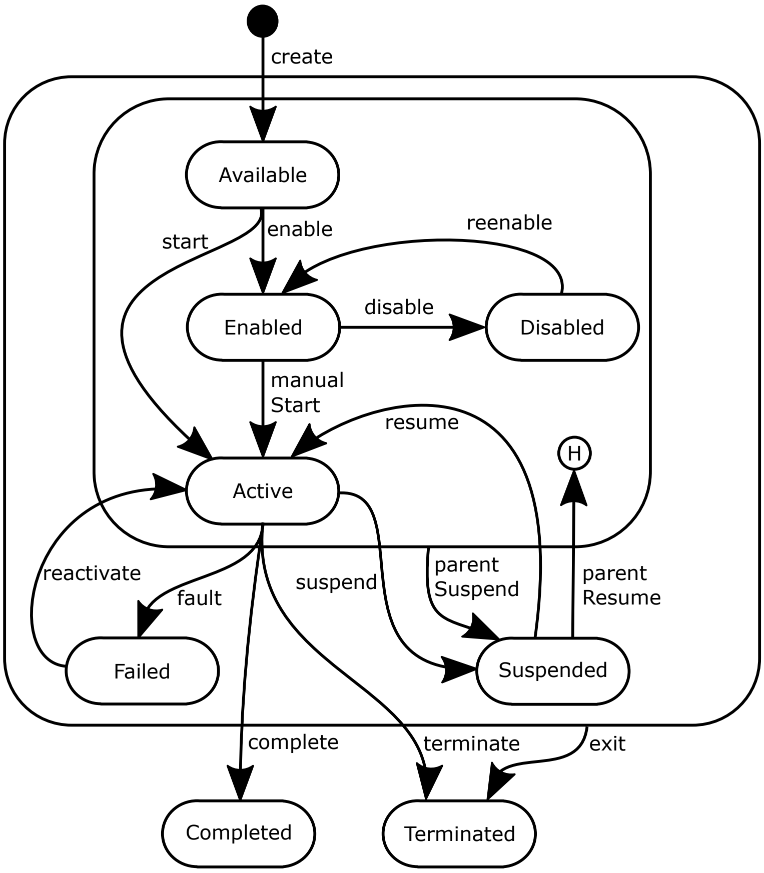


Figure 5.8: Statechart depicting the life cycle of stages and tasks [OMG14a] ($x \in \mathcal{St}_{planned} \vee x \in \mathcal{St}_{discretionary} \vee x \in \mathcal{Ta}$)

idiosyncrasies in the transformation. The $\Gamma^{milestone}_{pattern}(m)$ function takes a $m \in \mathcal{Mi}$ and generates the pattern by converting the milestone entry criteria into GSM milestones. The $\Gamma^{eventlistener}_{pattern}(e)$ function takes $e \in \mathcal{Ev}$ and creates a GSM milestone for the pattern.

5.3.1.5 Rule Patterns

In addition to the life cycle of entities, a case type has four rules that alter the behavior of the life cycle of the case, stages, tasks, and milestones. This section describes how these patterns alter the life cycle patterns described earlier. In CMMN, each rule is described by a Boolean expression and the rule is said to be present if the Boolean expression exists. It is possible for a rule to be present but evaluate to false and so it does not have any effect on the entity (case, stage, task, or milestone).

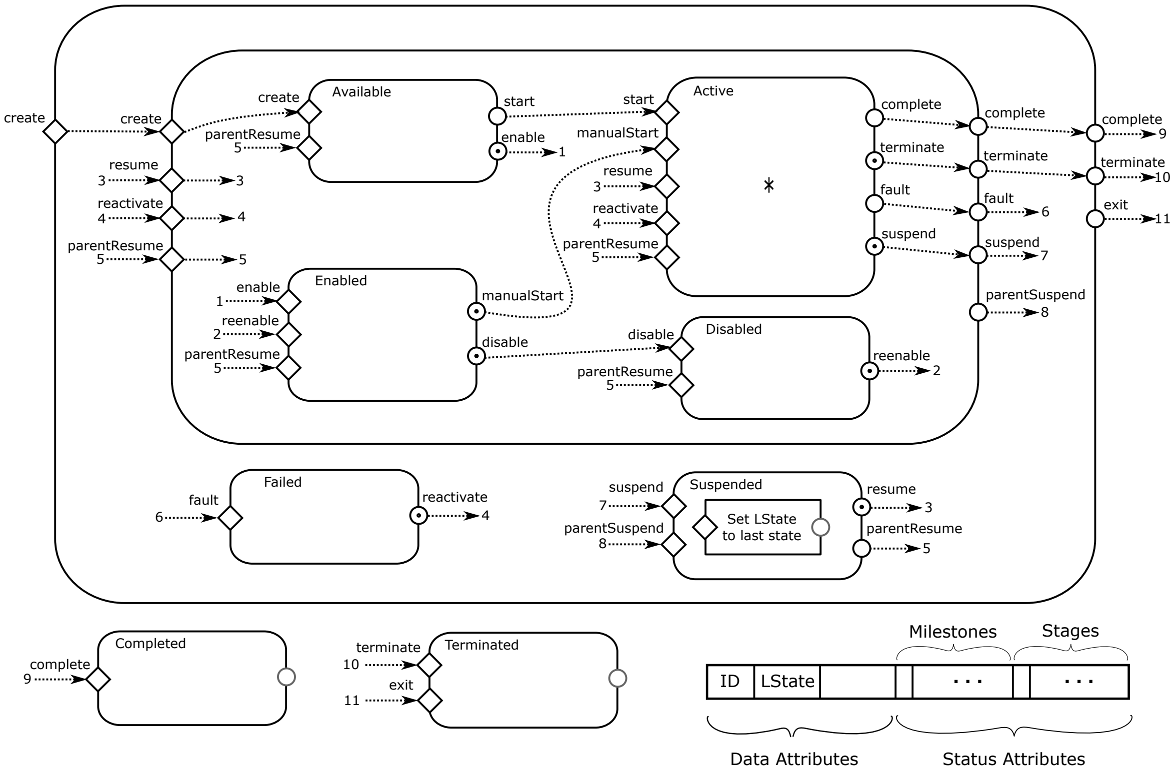


Figure 5.9: GSM pattern for the life cycle of stages and tasks
GSM version of the statechart depicted in Figure 5.8 implementing $\Gamma^{stage}_{pattern}$ and $\Gamma^{task}_{pattern}$ for $x \in \mathcal{St}_{case} \cup \mathcal{St}_{planned} \cup \mathcal{St}_{discretionary} \cup \mathcal{Ta}$

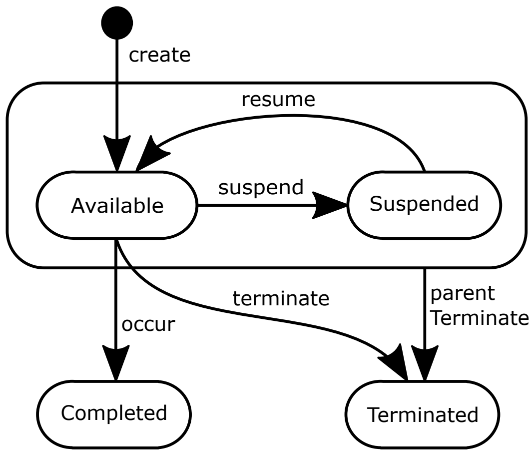


Figure 5.10: Statechart depicting the life cycle of event listeners and milestones [OMG14a] ($x \in \mathcal{Ev} \ \vee \ x \in \mathcal{Mi}$)

Manual activation rules

A manual activation rule is described by the binary relationship $\widehat{\mathscr{M}}$, and it affects the life cycle pattern of the case, stages and tasks (Figure 5.9). In CMMN, manual activation functionality allows a case worker to manually start a case, stage, or task after its entry criteria have been satisfied. This functionality gives control to the case worker over which stages or tasks must be executed.

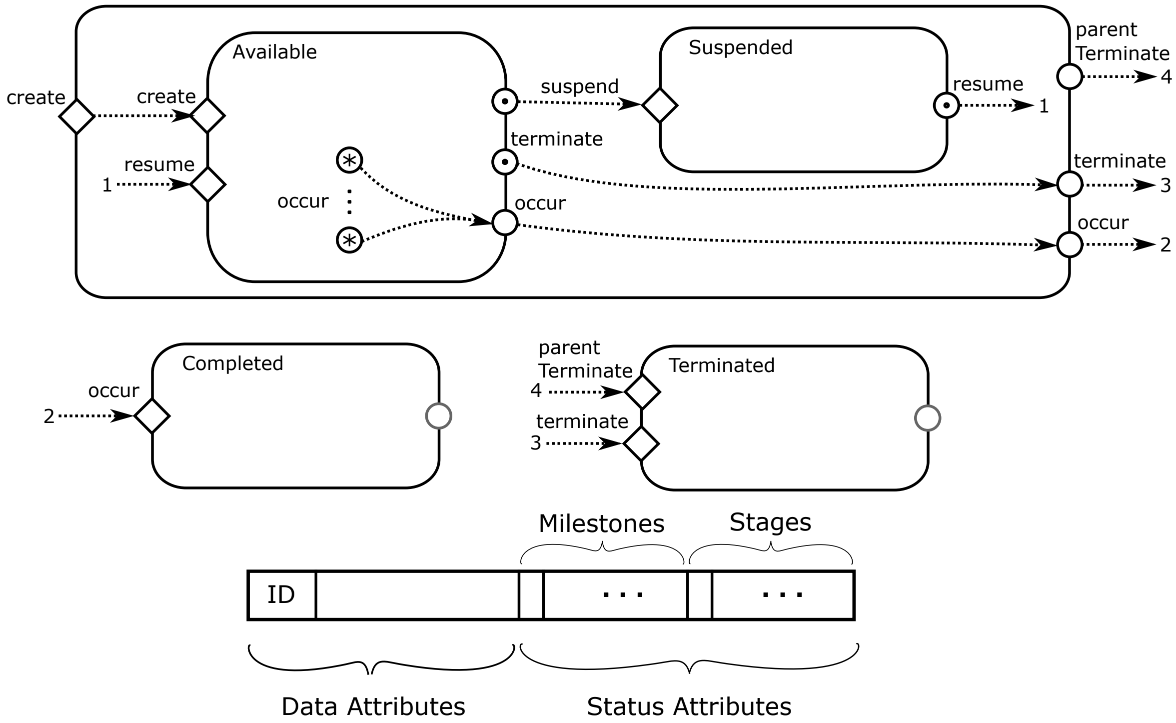


Figure 5.11: GSM pattern for the life cycle of event listeners and milestones ($x \in \mathcal{Ev} \vee x \in \mathcal{Mi}$) GSM version of the statechart depicted in Figure 5.10 implementing $\Gamma^{milestone}_{pattern}$ and $\Gamma^{eventlistener}_{pattern}$

This pattern encloses the original case, stage or task in two embedded stages. The outer stage has all of the guards and terminators of the original entity. The inner stage contains two guards. One has a sentry containing the manual activation event in the event expression ξ, as well as the manual activation Boolean expression in the condition expression φ. The other has a sentry containing the negation of the manual activation Boolean expression in the condition expression φ, and no event expression ξ. In addition, the terminators of the original entity will be duplicated in this stage. Figure 5.12 illustrates the pattern. The $\Gamma^{manual}_{pattern}(x, y)$ function takes a $\langle x, y \rangle \in \widehat{\mathscr{M}}$ and implements the pattern by updating the life cycle of the case, stage or task.

Autocomplete rules

Autocomplete rules are described by the set $\widehat{\mathscr{A}}$, and these affect the life cycle pattern of the case (Figure 5.5) and stages (Figure 5.9). Autocomplete indicates that the stage should complete normally when all of its enclosed entities (i.e., stages, tasks, event listeners, or milestones) have completed. Non-autocomplete stages require manual intervention by a case worker in order for the stage to complete. Autocomplete is the default behavior in GSM.

The $\Gamma^{auto}_{pattern}(a)$ function takes $a \in \widehat{\mathscr{A}}$ and updates the life cycle pattern by enhancing the terminators so that they can be used by the case workers to complete a stage or a case. In

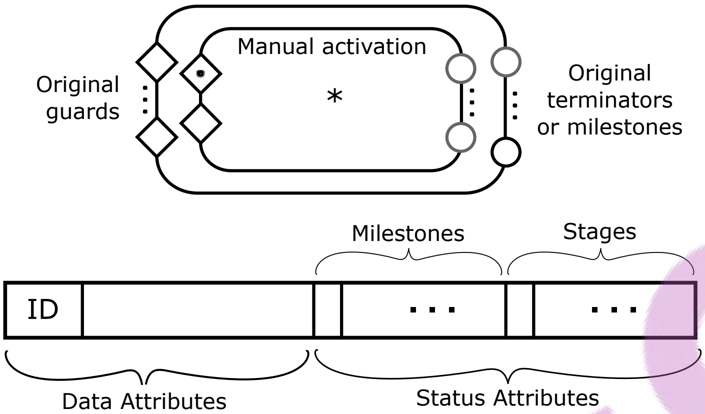


Figure 5.12: GSM pattern for manual activation ($\langle x, y\rangle \in \widehat{\mathscr{M}}$)
Implements $\Gamma^{manual}_{pattern}$

addition, this function adds a new terminator that can be triggered exclusively by the case workers.

Required rules

Required rules are described by the binary relationship $\widehat{\mathscr{R}}$. In CMMN stages, tasks, and milestones can be defined as "required", but the life cycle that is affected is the enclosing scope (either a stage or the case itself). "Required" in CMMN indicates that the entity (i.e., stages, tasks, or milestones) must complete in order for the enclosing scope (i.e., case or stage) to complete.

This pattern enhances all of the terminators of the enclosing scope by adding a negation of the required rule's Boolean expression in the ξ expression. The $\Gamma^{required}_{pattern}(x, y)$ function takes $\langle x, y\rangle \in \widehat{\mathscr{R}}$ and updates the enclosing scope pattern accordingly.

Repetition rules

Repetition rules are described by the binary relationship $\widehat{\mathscr{N}}$, and affect the life cycle pattern of stages, tasks (Figure 5.9) and milestones (Figure 5.11). The repeated execution of stages, tasks and milestones is controlled in CMMN by the repetition rule. Repeating entities may result in the same entity executing in parallel. For example a task that should be executed every time a picture arrives in the case may have several instances executing in parallel when multiple pictures arrive around the same time. In GSM everything is repeatable, but only serial repetition is allowed because the status is held in a single Boolean variable.

This pattern is split into two patterns. First, there is a pattern that controls the ability to repeat an execution of a stage, task or milestone. The function $\Gamma^{repetition}_{pattern}(x, y)$ takes a $\langle x, y\rangle \in \widehat{\mathscr{N}}$ and adjusts the pattern associated with y by negating the repetition rule's Boolean expression, but includes it in the guard of the stages, tasks, or milestones. Secondly, there is a pattern that deals with parallel execution which is described in further detail in Section 5.3.1.7. The parallel execution pattern is applicable to the repetition rule, discretionary items, and plan fragments patterns.

5.3.1.6 Discretionary Item Pattern

Stages and tasks can be discretionary in CMMN. In addition, CMMN contains the concept of a plan fragment that is simply a group of discretionary items (i.e., discretionary stages, discretionary tasks, or plan fragments). A discretionary item is only executed if a case worker manually adds it to the execution plan.

The pattern for a discretionary item starts by first expanding the task or stage pattern, and then wrapping that expansion with the discretionary pattern as shown in Figure 5.13. The $\Gamma^{discretionary}_{pattern}(d)$ function takes $d \in \mathcal{St}_{discretionary} \cup \mathcal{Ta}_{discretionary} \cup \mathcal{St}_{fragment}$ and applies the pattern.

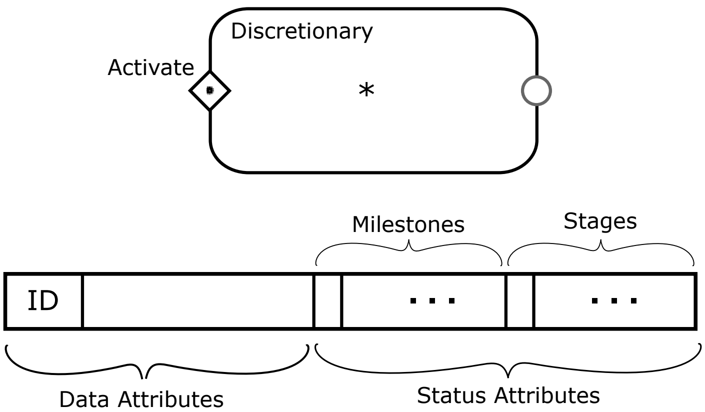


Figure 5.13: GSM pattern for discretionary items
Implements $\Gamma^{discretionary}_{pattern}$ for $x \in \mathcal{St}_{discretionary} \cup \mathcal{Ta}_{discretionary} \cup \mathcal{St}_{fragment}$

5.3.1.7 Parallel Execution Pattern

GSM cannot execute stages in parallel because each stage has a Boolean status property. In order to maintain the integrity of the Boolean status property, GSM can only support sequential repetitions. A pattern that allows for the parallel execution of CMMN entities requires the definition of as many copies of the entity as the maximum number of expected parallel executions required. In addition, all expressions that refer to the standard events or state of the entity must be fixed as follows:

1. condition expression must incorporate the states of all the copies into an AND expression.

2. event expressions cannot contain multiple events, therefore they need to be changed into AND condition expressions.

This is done by creating user defined functions in OCL for each entity that requires repetition, and using that function in sentries that refer to the entity. Figure 5.14 illustrates this pattern. The pattern is implemented in two steps. The first step duplicates the stages, tasks, or milestones n times. The second step changes all of the sentries that refer to the state or events so that they use the functions instead of the original expression. The $\Gamma^{parallel}_{pattern}(x)$ function takes a pattern x and makes n copies, creates the functions, and searches the Γ for sentries that need to be fixed. Note that n is a constant indicating the number of parallel executions that will be supported by the resulting artifact type.

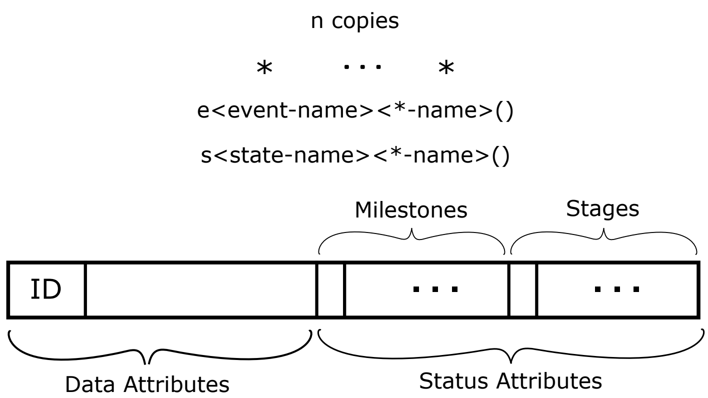


Figure 5.14: GSM pattern for parallel execution
Implements $\Gamma^{parallel}_{pattern}$

5.3.1.8 Sentry Related Patterns and Transformation

There are a few important differences between sentries in CMMN and sentries in GSM. CMMN allows for event expressions $\dot{\xi}$ with multiple events, while in GSM this is restricted to a single event. The condition expression φ can be written in any expression language that is supported by the CMMN engine. The default expression language in CMMN is XPath [OMG14a], while GSM can only support OCL.

Condition expressions

Condition expressions φ have to be transformed into OCL. The transformation is language dependent and outside the scope of this thesis. This thesis indicates where this transformation is required with the $exprConvert(s)$ function.

Event expressions

This transformation takes a CMMN event expression and converts it into a GSM event expression as follows:

open events are converted as follows,

Replaces <*occur*>:m with $+m$ (milestone m is triggered).

Replaces <*start*>:S with $+S$ (stage S opened).

Replaces <*start*>:t with I:t (task t is invoked).

close events are converted as follows,

Replaces <*complete*>:S with $-S$ (stage S closes).

Replaces <*complete*>:t with C:t (task t completes).

terminate events are converted as follows,

Replaces <*terminate*>:S with $-S$ (stage S closes).

Replaces <*terminate*>:t with C:t (task t completes).

This thesis describes this transformation using the $eventConvert(s)$ function.

Split entry criteria

Entry criteria containing event expressions with more than one event must be split into multiple guards with each guard containing a single event expression. This pattern takes an entry criterion in a stage, task, or milestone and does the following:

1. counts the number of events in the event expression (let us say n).
2. creates n embedded stages.
3. splits the entry criteria into n guards, each one with a single event in the event expression, and the same condition expression.
4. each of the n stages gets all the n guards, and any other guards that the original stage, task, or milestone had.
5. the n stages get all of the terminators from the original entity.

Figure 5.15 depicts this pattern. This thesis describes this pattern using the $\Gamma_{pattern}^{splitEntry}(x, y)$ function.

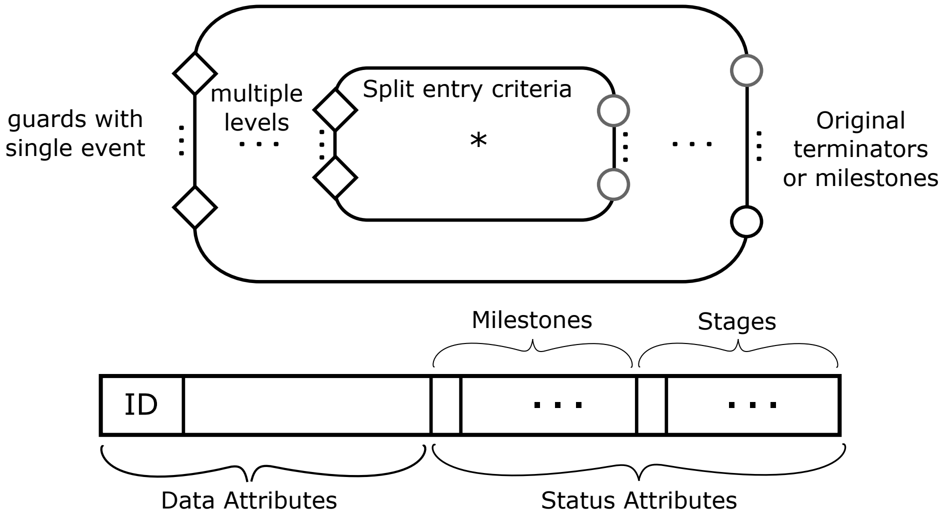


Figure 5.15: GSM transformation to split entry criteria ($\langle x, y\rangle \in \widehat{\mathscr{E}}$)
Implements $\Gamma^{splitEntry}_{pattern}$

Split exit criteria

Exit criteria containing event expressions with more than one event must be split into multiple terminators each containing a single event expression. This pattern takes an exit criterion in a case, stage, or task and does the following:

1. counts the number of events in the event expression (let us say n).
2. if $n > 1$ then create a stage and do
   (a) create n tasks and n Booleans.
   (b) split the exit criteria into n guards, each one with a single event in the event expression, and the original condition expression.
   (c) each of the n tasks gets one guard and a terminator that tests if the Boolean is true (this is required just to terminate the task).
   (d) each task turns its Boolean to one which indicates that the event expression did occur.
   (e) the new stage gets a single terminator without an event expression, but receives a condition expression similar to the original condition expression and an AND-expression that concatenates all of the Booleans.

This thesis describes this pattern using the $\Gamma^{splitExit}_{pattern}(x, y)$ function.

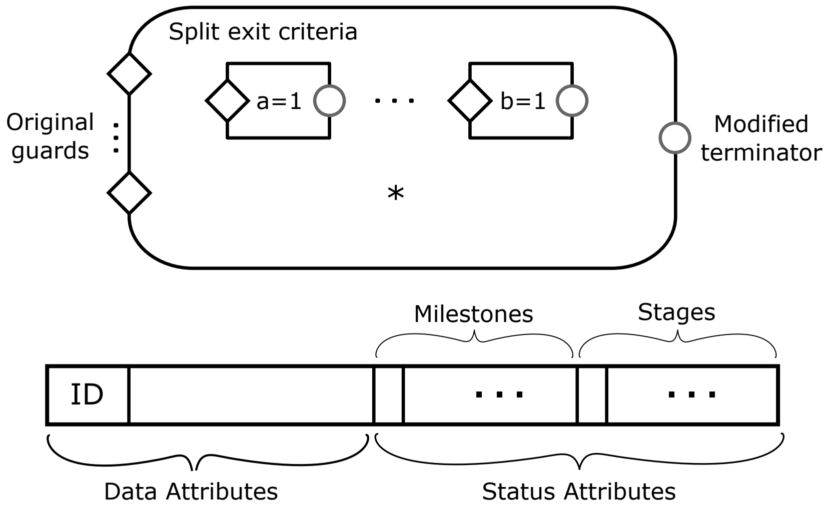


Figure 5.16: GSM transformation to split exit criteria ($\langle x, y\rangle \in \widehat{\mathscr{X}}$)
Implements $\Gamma_{pattern}^{splitExit}$

Standard events

To complete the transformation to GSM, all of the CMMN standard events used by the case type should be added to the GSM generated events set $EType_{gen}$. This thesis describes this transformation using the *standardEvents*(x) function.

5.3.2 Transformation Algorithm

A syntax directed translation [Aho+07] grammar was used to describe the algorithm. A syntax directed translation works well with translating the first-order logic definition of a case type into an artifact type definition. The simple context-free grammar is derived from the definitions for Case type (Definition 3.4), Case behavior (Definition 3.5), Case hierarchy (Definition 3.6), Case behavioral rules (Definition 3.7), and CMMN Sentry (Definition 3.8). Fortunately, the recognition of the resulting grammar preserves the ordering required for the transformation.

A set of rewriting rules was developed based on the patterns described in Section 5.3.1. The rewriting rules are presented in Table 5.2. The semantic rules in the grammar are used to apply the rewriting rules. The resulting grammar is described in Appendix A.

Note that a case type $\mathcal{T}$ is transformed into a set of artifact types Γ. The main artifact type corresponds to the behavior $\mathcal{B}$ of the case type $\mathcal{T}$, and to a set of data attribute artifact types that correspond to each data attribute in $\mathcal{D}$. The number of artifact types Γ is derived by using the equation $1 + |\mathcal{D}|$.

Table 5.2: Rewriting rules

Order	Concept	Condition	Transformation	$\mapsto$ GSM
1	Data	$\forall d \in \mathcal{D}$	$\Gamma^{data}_{pattern}(d)$	$\Gamma \leftarrow \Gamma \cup \Gamma^{data}_{pattern}(d)$
2	Case stages	$c \in \mathcal{St}_{case}$	$\Gamma^{case}_{pattern}(c)$	$\Gamma \leftarrow \Gamma \cup \Gamma^{case}_{pattern}(c)$
3	Stages	$\forall S \in \mathcal{St}_{planned}$	$\Gamma^{stage}_{pattern}(S)$	$\Gamma \leftarrow \Gamma \cup \Gamma^{stage}_{pattern}(S)$
4	Discretionary stages	$\forall dS \in \mathcal{St}_{discretionary}$	$\Gamma^{parallel}_{pattern}(\Gamma^{discretionary}_{pattern}(\Gamma^{stage}_{pattern}(dS)))$	$\Gamma \leftarrow \Gamma \cup \Gamma^{parallel}_{pattern}(\Gamma^{discretionary}_{pattern}(\Gamma^{stage}_{pattern}(dS)))$
5	plan fragment	$\forall f \in \mathcal{St}_{fragment}$	$\Gamma^{parallel}_{pattern}(\Gamma^{discretionary}_{pattern}(f))$	$\Gamma \leftarrow \Gamma \cup \Gamma^{parallel}_{pattern}(\Gamma^{discretionary}_{pattern}(f))$
6	Tasks	$\forall t \in \mathcal{Ta}_{planned}$	$\Gamma^{task}_{pattern}(t)$	$\Gamma \leftarrow \Gamma \cup \Gamma^{task}_{pattern}(t)$
7	Discretionary tasks	$\forall dt\mathcal{Ta}_{discretionary}$	$\Gamma^{parallel}_{pattern}(\Gamma^{discretionary}_{pattern}(\Gamma^{task}_{pattern}(dt)))$	$\Gamma \leftarrow \Gamma \cup \Gamma^{parallel}_{pattern}(\Gamma^{discretionary}_{pattern}(\Gamma^{task}_{pattern}(dt)))$
8	Milestones	$\forall m \in \mathcal{Mi}$	$\Gamma^{milestone}_{pattern}(m)$	$\Gamma \leftarrow \Gamma \cup \Gamma^{milestone}_{pattern}(m)$
9	Event listeners	$\forall e \in \mathcal{Ev}$	$\Gamma^{eventlistener}_{pattern}(e)$	$\Gamma \leftarrow \Gamma \cup \Gamma^{eventlistener}_{pattern}(e)$
10	Hierarchy	$\forall \langle x, y \rangle \in \mathcal{H}$	Force patterns at the right level	$Substages^R \hookleftarrow Substages^R \sqcup \{\langle x, findPattern(y) \rangle \mid \langle x, y \rangle \in \mathcal{H}\}$
11	rule expressions	$\forall r \in \widehat{\varphi}$	Create new sentries	$Stry \leftarrow Stry \sqcup \{[\textbf{if } toOCL(r)]\}$
12	Sentries	$\forall s \in \widehat{\mathscr{S}}$	Convert sentries	$Stry \leftarrow Stry \sqcup \{s \mid s = [\textbf{on } eventConvert(\dot{\xi}) \textbf{ if } exprConvert(\varphi)] \vee \ldots\}$
13	Entry criteria	$\forall \langle x, y \rangle \in \widehat{\mathscr{E}}$	Attach guard to the right stage	$Guards^R \hookleftarrow Guards^R \sqcup \{\langle \Gamma^{splitEntry}_{pattern}(x, findPattern(y)) \rangle \mid \langle x, y \rangle \in \widehat{\mathscr{E}}\}$
14	Exit criteria	$\forall \langle x, y \rangle \in \widehat{\mathscr{X}}$	Attach terminator to the right stage	$Terminators^R \hookleftarrow Terminators^R \sqcup \{\langle \Gamma^{splitExit}_{pattern}(x, findPattern(y)) \rangle \mid \langle x, y \rangle \in \widehat{\mathscr{X}}\}$
15	Manual activation	$\forall \langle x, y \rangle \in \widehat{\mathscr{M}}$	$\Gamma^{manual}_{pattern}(x, y)$	$\Gamma \leftarrow \Gamma \sqcup \Gamma^{manual}_{pattern}(x, findPattern(y))$
16	Required	$\forall \langle x, y \rangle \in \widehat{\mathscr{R}}$	$\Gamma^{required}_{pattern}(x, y)$	$\Gamma \leftarrow \Gamma \sqcup \Gamma^{required}_{pattern}(x, findPattern(y))$
17	Repetition	$\forall \langle x, y \rangle \in \widehat{\mathscr{N}}$	$\Gamma^{repetition}_{pattern}(x, y)$	$\Gamma \leftarrow \Gamma \sqcup \Gamma^{parallel}_{pattern}(\Gamma^{repetition}_{pattern}(x, findPattern(y)))$
18	Autocomplete	$\forall a \in \widehat{\mathscr{A}}$	$\Gamma^{auto}_{pattern}(x)$	$\Gamma \leftarrow \Gamma \sqcup \Gamma^{auto}_{pattern}(findPattern(y))$
19	Standard events	$\exists e \in \widehat{\mathscr{S}} \wedge e' = \dot{\chi}$	$standardEvents(x)$	$EType_{gen} \leftarrow EType_{gen} \cup standardEvents(\mathcal{T})$

5.4 Summary

This chapter contributes formal transformations between CMMN case types and GSM artifact types to the broader BA knowledge base. The transformation from a GSM artifact type to a CMMN case type is relatively straightforward and simple to describe. The resulting case type modeled using CMMN is visually similar to the original artifact type, allowing for it to be easily understood by human beings. The transformation of a case type into an artifact type is more complex because CMMN extends GSM by introducing new constructs and defining a life cycle with a set of standard events for those constructs. However, it does allow CMMN to use the formal operational semantics of GSM. It also allows for the formal verification work developed for GSM to be applied to CMMN. Material from Appendix D file 30 (FSM-2-GSM.pdf) has been published in Marin et al. [Mar+16].

Chapter 6

Systematic Literature Review of Process Modeling Complexity Metrics

This chapter contributes a Systematic Literature Review (SLR) [Bio+05; Kit04; Woh+12] of complexity metrics for process models to the broader body of knowledge associated with complexity metrics for process models. The review was designed to identify complexity metrics for process models that have been proposed in the last 20 years (from January 1996 to June 2016 inclusive), and how these were validated. The goal of this review was to identify complexity metrics for process models with the specific purpose of identifying metrics that could be relevant to Case Management Model and Notation (CMMN) [OMG14a]. In addition, this review was conducted in order to identify the research methods, present in the literature about process models, used to validate complexity metrics that could be adapted to validate CMMN complexity metrics. This chapter follows the recommendations put forward for conducting systematic reviews of software engineering literature by Biolchini et al. [Bio+05], Kitchenham [Kit04], Kitchenham and Charters [KC07], Kitchenham et al. [Kit+04], Sjoberg et al. [Sjo+07], and Wohlin et al. [Woh+12] and utilizes the definitions provided in Section 2.2. The complexity metrics for process models identified in this chapter and the research methods used to validate them form the basis for Chapters 7 and 8.

This chapter is organized as follows. Section 6.1 provides the justification and background information for the review. Section 6.2 describes the research questions used for this review. Section 6.3 describes the research protocol used, including the data sources and search strategy. Section 6.3.6 describes the inclusion and exclusion criteria used for the papers that were identified during the initial search. Section 6.4 describes the results of the review. Section 6.5 describes the principal findings of the review. In addition, Appendix B provides all of the papers and metrics identified in this SLR. Appendix D.3 contains all of the supplementary material including the raw data for the review. File 61 (SLR-analysis.Rmd) in Appendix D contains the R source code used to perform the quantitative analysis of the SLR data. Appendix D, file 28 (SLR-analysis.pdf) contains the resulting SLR analysis document.

6.1 Background

This review was the first step in the process of identifying and validating complexity metrics for CMMN. The complexity metrics for process models that were identified in the review were analyzed in order to identify metrics and research methods used for validation that could be applicable to CMMN. The review was conducted using the State of the Art through Systematic Reviews (StArt) tool [Fab+13; Fab+16], which was selected based on the evaluations done by Hernandes et al. [Her+12] and Marshall et al. [Mar+14c].

6.1.1 Related Work

Four other SLRs, relevant to this study, were identified in the area of metrics for process models. Three of these reviews [Muk+10a; PC16; Ś+10a] have identified process metrics that had undergone validation. Although the four review [Mor+15] did not identify metrics however the content was still related to this SLR. None of these four reviews examined the research methods or the quality of the validations performed on the metrics.

Polančič and Cegnar [PC16] conducted an SLR of the complexity metrics of process models to provide an overview of the metrics used to assess process model complexity. They conducted a search over a period of ten years (from 2005 to 1 February 2015) and identified 43 relevant process modeling papers with 66 complexity metrics. Their primary objectives were to identify the available complexity metrics and assess their usefulness for evaluating the complexity of a process model. Their main question was: "What are the existing metrics for measuring process diagrams complexity?" [PC16]. In addition to the main question, they had four sub-questions which are outlined and answered below:

- What are the underlying theories and domains used for deriving metrics? They found that the domain for most metrics was Business Process Management (BPM) (53%) followed by various domains (24%) and software engineering (23%). They classified the underlying theories according to the authors' definitions of the metrics, and found that most metrics were based on the graphical notation definition (38%), followed by graph theory (27%), software complexity (12%), Petri Net (9%), Cognitive load theory (7%), Shannon's information entropy (4%), and number theory (3%).

- In what way do metrics measure complexity? They identified 12 diverse metrics that were used as foundation metrics.
- How good are the metrics at measuring the complexity of the process models as a whole? They looked at the type of constructs (Activities, control-flow, data-flow, and resources) used to calculate the metrics, and found that only 3% of the metrics considered all of the constructs. Most metrics used control-flow (39%), and the rest used a combination of the constructs.
- Are metrics validated and useful in practice? They found that only two of the metrics had been theoretically and empirically validated (3%), while the majority of the metrics had either only been empirically validated (56%) or had not been validated at all (41%).

Sánchez-González et al. [S+10a] conducted an SLR to identify the state of the art in business process measurements and trends. Their main research objective was to identify the most current and useful initiatives used to measure business processes. There was no time limit on their search criteria, which yielded 49 relevant papers. With respect to the measurement concepts, they found that for modeling measurements the most used measurement concept was complexity (44%), followed by understandability (21%). All of the other concepts, which accounted for 35% of the measurement concepts, were equally represented at 7% (quality, entropy, density, cohesion, and coupling). They categorized the types of measurements as modeling measurements, which had the most papers (77%), and execution measurements (23%). They found that most papers had not validated their metrics (59%), but that some had used empirical validation (35%) and others theoretical validation (6%). They concluded that there was a significant tendency to create metrics without using any validation or empirical support. They also found that the focus was on process design complexity metrics, with very little research being done on process execution metrics. Their SLR looked at metric validation by creating three categories: not validated (59%), theoretically validated (6%), and empirically validated (35%).

Muketha et al.'s [Muk+10a] survey of business process complexity metrics was designed to identify gaps in the literature. While their literature review covered five years, from 2005 to 2010, it appears not to have followed SLR protocol. They found 6 relevant papers and identified 26 complexity metrics of which only 1 had been both theoretically and empirically validated. Echoing Sánchez-González et al.'s [S+10a] findings, they also established that very few of the metrics had been validated either theoretically or empirically. Additionally, like Polančič and Cegnar [PC16], they found that when validated, empirical validation was the preferred method of validation.

Moreno-Montes de Oca et al. [Mor+15] conducted an SLR on business process modeling quality to identify the state of the art in business process modeling quality and gaps in the literature. They found 72 relevant papers and their SLR covered 14 years (from 2000 to August

2013). Their research questions centered on the maturity of the business process modeling research field, which types of quality issues were being researched, and how the studies were conducted to address these issues. Their conclusion was that there is no agreement on what constitutes business process modeling quality.

Most of the papers in this review included a short survey of BPM metrics in addition to defining new metrics [Abr+10; Çoş14; Klu15; Por10; Ś+11; Ś+12; Sol+13a].

6.1.2 Context

Business process modeling is critical if enterprises are to understand and redesign the activities used to achieve their business goals [Mor+15]. Metrics are an important aspect of process modeling quality as they help guide process improvement [Sán+15]. Sánchez-González et al. [Ś+10a] found that complexity is one of the most researched concepts in process modeling metrics with 44% of the papers that they reviewed being dedicated to this topic. Therefore, it is important to understand which complexity metrics have been created for process models and which research methods have been used to validate them.

Being a declarative process modeling approach, CMMN could benefit from complexity metrics for declarative process models or data-centric approaches. It could also benefit from other complexity metrics for process models that can be adapted to declarative process models. Any potentially new metric developed for CMMN will need to be validated, and so an understanding of the research methods used in the process modeling literature will inform the validation of CMMN metrics. For the purposes of this thesis, it was important to understand how the design time process model complexity metrics in the literature could inform the creation and validation of complexity metrics for CMMN.

6.1.3 Validation of Process Metrics

There seems to be consensus amongst researchers that there are three steps that are required in order to define and validate a software metric [Cal+01; Muk+10a; PC16; Ser+02; Son09]. These steps include the definition of the metrics, a theoretical validation of the metrics, and an empirical validation of the metrics. In some cases researchers have added additional steps. For example, Muketha et al. [Muk+10a] include the implementation of the metric in a tool as an optional step, and Calero et al. [Cal+01] include a psychological explanation of the metrics as an additional step. This SLR follows the general consensus in its approach to identifying the different types of theoretical and empirical validations used in the literature of complexity metrics for process models. Figure 6.1 shows the consensus steps for metric creation and validation.

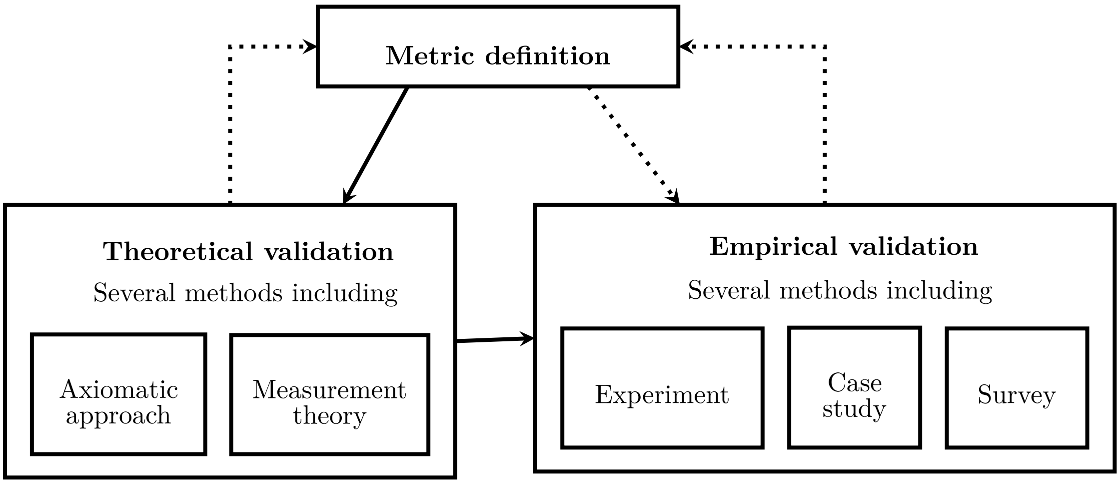


Figure 6.1: Metric definition and validation steps adapted from Calero et al. [Cal+01], Muketha et al. [Muk+10a], and Soni [Son09] and Serrano et al. [Ser+02]

Theoretical validation can be conducted using measurement theory such as the DISTANCE framework by Poels and Dedene [PD00], axiomatic approaches like the one proposed by Weyuker [Wey88] or by Briand et al. [Bri+96], or more qualitative approaches such as those used by Kaner and Bondapproach [KB04]. The main goal of theoretical validation is to ensure that the new metrics do not violate measurement theory [Muk+10a].

Empirical validation can be conducted using multiple experimental designs. Some authors list surveys, case studies, and experiments as appropriate and common methods for the empirical validation of metrics [Muk+10a; PC16; Ser+02; Son09]. Metrics by definition are quantitative, and therefore quantitative methods seem appropriate to validate them. In most cases the empirical validation of metrics must be done using an appropriate statistical technique based on the scale of the metric being analyzed [Bri+95].

6.2 Review Questions

The following research questions were identified for this review in order to support the goal of identifying metrics and research methods that could assist with the definition and validation of CMMN metrics:

RQ1: Which static complexity metrics for process models have been identified in the literature? The aim was to create an inventory of complexity metrics for process models that would serve as the starting point for identifying design time complexity metrics for CMMN. For this reason this SLR focused on statically calculated complexity metrics, i.e. metrics that do not require runtime information. This question led to the following sub-question:

RQ1.1: Which notations have been used in the literature to identify complexity metrics for process models? The goal was to see if declarative process model notations or data-centric process model notations had been used because metrics for these notations can easily be adapted to CMMN.

RQ2: Which types of research methods, present in the literature, have been used to validate complexity metrics for process models? The aim was to create an inventory of research methods that could be used to design validation methods for CMMN complexity metrics.

RQ3: How have the complexity metrics for process models described in the literature been validated? The motivation was to understand the established validation strategies that could be used to validate potential CMMN complexity metrics. This question led to the following sub-questions:

RQ3.1: Which of the complexity metrics for process models, identified in the literature, have been validated? The objective was to understand how often and how many validations complexity metrics for process models underwent.

RQ3.2: How many human subjects are used during the validation of complexity metrics for process models? The motivation was to understand the number of subjects required to validate complexity metrics for process models.

6.3 Review Methods

In accordance with suggestions made by Kitchenham and Charters [KC07], the SLR consisted of three phases, namely planning, execution, and reporting. This section describes the planning phase, and provides an overview of the methods to be applied in this study. The planning phase consisted of identifying the need, scope, data sources, search criteria, and the steps required for the execution phase of the review. During this phase, the StArt tool was selected for its ability to support most phases of the SLR [Her+12; Mar+14c]. Because the StArt tool was selected, it required a research protocol that would accommodate it.

The execution phase of the SLR was planned in order to include the following activities:

Search and identification of studies based on the research databases and search strings identified during the planning phase.

Selection of studies based on inclusion and exclusion criteria. Each paper in the review has to be matched to a selection criteria, and categorized as accepted, rejected, or duplicated.

Data extraction from accepted studies. Each accepted paper has to be read in order to extract the information required to answer the review questions. In addition, new studies will be identified using backward snowballing based on citations in the papers included in the review. During this activity, the accepted papers will be classified as primary, secondary, duplicated, survey, or uses.

Data analysis activities will be conducted using R [R C16], in order to analyze the collected data and to tabulate it.

Although the activities performed during the execution phase are presented in sequential order, with the only exception being the research database queries, all of the activities were planned so as to be conducted in an iterative fashion.

The reporting phase produced a report containing the findings. In this thesis the reporting phase is presented in Section 6.4, and the full report is provided in Appendix D file 28 (SLR-analysis.pdf).

6.3.1 Data Sources and Search Strategy

The research databases used in this review include the Association of Computing Machinery (ACM) Digital Library, the Institute of Electrical and Electronics Engineers (IEEE) Xplore Digital Library, Springer's SpringerLink, Elsevier's Scopus, Elsevier's ScienceDirect, Thomson Reuters' Web of Science, and Google Scholar. These databases were selected because they are well known and contain references to BPM workflow and process modeling. These research databases also index the main conferences in the area of BPM, and are available to researchers at the University of South Africa (UNISA).

The planned query for this review was:

Dates: 1996 to 2016;

Title: (Complexity OR Metric OR Metrics) AND (Process OR Processes OR Workflow OR EPC OR BPEL OR BPMN OR BPM OR YAWL);

Abstract: (complexity OR complex) AND (process OR processes OR workflow) AND (measure OR measures OR measuring OR metric OR metrics OR measurement);

Keywords: BPM; process; Workflow; complexity measurement; complexity metric; metric; define; calculate.

In addition, backward snowballing [Woh14; Woh16] was used to identify papers that were not found in the research databases. Backward snowballing is done by using the bibliography in the accepted papers to identifying additional papers for inclusion in the review.

6.3.2 Study Selection

Papers that were included in this study were first identified via the search string used in the research databases and the time span of the search. In particular, the selection criteria included papers related to BPM or workflow modeling that described modeling metrics and contained some indication from the author that those metrics could be used to measure complexity metrics for process models.

Each paper that was identified through the research databases as a result of the search string was classified as:

Accepted were papers that met the inclusion criteria (see Figure 6.2).

Rejected were papers that did not meet the inclusion criteria.

Duplicated were papers identified by more than one research database. They corresponded to identical papers returned by the different research databases.

Each accepted paper was further classified during the data extraction activity as follows:

Primary were papers that described new complexity metrics at the process level.

Secondary were papers that described metrics already presented in primary papers, usually with the purpose of validating them.

Duplicated were papers that did not provide any new material, because they presented the same text, metrics or results included in other papers classified as primary or secondary.

Survey were papers that surveyed complexity metrics.

Uses were papers that made use of complexity metrics for process models.

Rejected were papers that did not satisfy the review's inclusion criteria and were not filtered during the selection phase. These were papers that may have defined complexity metrics for models, but those metrics may not have been at the process level.

Only primary and secondary papers were used in the data analysis activities reported in Section 6.4.

6.3.3 Data Extraction

Data extraction was performed by reading the document multiple times. In each instance, specific data was extracted. The data extracted from the papers was recorded in the StArt tool and in a Microsoft Excel spreadsheet (see Appendix D file 32 (Analysis.xlsx)). The variables that were extracted are documented in Appendix D file 49 (SLR-data(variable-description).pdf).

Some papers claimed that their metrics were applicable to generic workflow and processes, or to several modeling notations, but used a particular notation to define, describe or to test the metric. For the sake of this review, only the notation that corresponded to the notation used when a metric was defined or tested was recorded. If other notations were included in the paper, these were provided in the description of the article (see Appendix B.1).

Some papers may have included experiments where the focus was not to validate the metrics. Those experiments were excluded from this review. Other papers may have contained experiments designed to test several aspects of a metric, in such cases only the metric validation information was extracted. However, concepts related to complexity such as error prediction were included.

The data extraction was conducted at three levels:

Paper level. Information from all of the accepted papers was extracted and recorded. For papers categorized as duplicate, survey, or uses, minimal information was extracted. Primary and secondary papers were the focus of the data extraction.

Validation level. Primary and secondary papers that contained validation information were used for that information. Some papers contained information for multiple validations or experiments and that information was recorded as individual validations. Validations were categorized into theoretical and empirical validations.

Metric level. Primary papers were those that proposed new metrics. Each one of those proposed metrics was individually recorded. Duplicate metric definitions were identified and labeled. Validation information for each metric was extracted at the metric level, this included a record for each validation conducted for a metric.

6.3.4 Study Quality Assessment

The quality of the experiments included in this review was evaluated using rigor, and relevance based on the work of Bin Ali et al. [Bin+14], Dybå and Dingsøyr [DD08], and Vasconcellos et al. [Vas+17]. Rigor is defined by Dybå and Dingsøyr [DD08] as the use of a thorough and appropriate research method in the study. Relevance is defined by Dybå and Dingsøyr [DD08] as how useful the findings are to the software industry or research community. The rigor and relevance categories used criteria from the work done by Bin Ali et al. [Bin+14] and

Table 6.1: Assessment criteria for experiments' rigor, credibility, and relevance

Rubric	1	0.5	0
	Rigor = design + validity + hypothesis		
Design	Good description of the study including the information needed to understand what was done.	A deficient description of the study.	No description of the study.
Validity Threats	Discusses all threats to validity (internal, external, conclusion and construct) [Woh+12].	Describes at least two of the threats to validity.	Otherwise.
Hypothesis	Formal hypotheses or clearly stated research questions.	Informal description of hypotheses or research questions.	No hypothesis or research questions.
	Relevance = subjects + scale + sample size		
Subjects	Professional practitioners for human validation or real industrial models for software validation.	Academics or experts for human validation or artificial models designed to test the hypothesis for software validation.	Other subjects.
Scale	More than 20 subjects for human validation or more than 20 models for software validation.	More than 10 (but less that 20) subjects for human validation or more than 10 (but less than 20) models for software validation.	Less than ten subjects for human validation or ten models for software validation.
Sample Size	Sample size justified using statistical analysis (e.g., power calculation).	Author discusses the selection of sample size.	No justification for the sample size used.

Vasconcellos et al. [Vas+17], which was adapted for this SLR. The categories and criteria were modified for this review as follows:

- Context and research methods were not included in this review because these give high ratings to studies conducted in industry environments and low ratings to lab experiments. However, it is legitimate to validate software metrics in a lab setting. Context was considered important by Bin Ali et al. [Bin+14] because they were looking at industrial processes. Both context and research methods were considered important

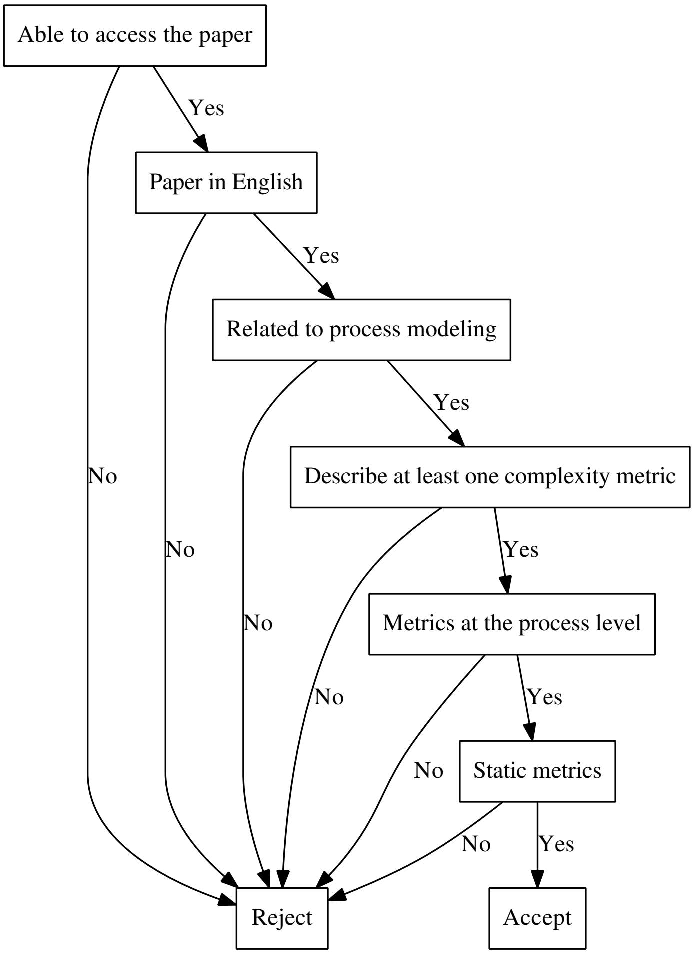


Figure 6.2: Selection criteria

by Vasconcellos et al. [Vas+17] because of their focus on the strategic alignment of software processes in an organization.

- New criteria (hypothesis and sample size) were included in this review because complexity metrics, which are quantities, are more likely to be validated using a quantitative method.

Table 6.1 shows the rubric used to evaluate the rigor and relevance categories.

6.3.5 Data Synthesis

Data was to be recorded using the StArt tool during the selection activity, and in an Excel Spreadsheet during the data extraction activity. The data recorded in the StArt tool was exported to an Excel Spreadsheet. The two Excel Spreadsheets were to be saved in a file format suitable for the R [R C16] tool to read. R scripts were to be used to process the extracted data and produce a report along with the graphs and tables that would be included in Section 6.4, and in Appendix B.

6.3.6 Included and Excluded Studies

The inclusion and exclusion criteria for this SLR have been summarized in Figure 6.2 and can be described as follows:

- Papers that describe at least one new complexity metric. In the context of this systematic review a complexity metric is any metric that the author believes can be used to measure complexity, understandability, or how error prone a model is.
- Only complexity metrics at the process model level are to be collected. Metrics for other aspects of a model, for example, activities or resources, should not be collected in the systematic review, unless they are required to calculate a complexity process level metric.
- Only complexity metrics that can be statically calculated are to be collected. Metrics that required runtime information for their calculations will not be collected in this review.
- Only complexity metrics that are fully and formally described are to be collected. For papers with complexity metrics, size metrics are also collected.
- The paper must be related to Workflow or BPM modeling in order to be included in this review.
- The paper must be written in English and must be available at the UNISA library, by inter-library loan, or online.

6.4 Results

The first activity in the SLR execution phase was to conduct the search in the research databases. The query statement described in Section 6.3.1 was adapted for each of the research databases used, as shown in Table 6.2. The query statement and results of the queries were recorded in the StArt tool and the bibliography files that were returned from the research databases were tagged with the date of the search (see Appendix D.3). The results from the different searches in the research databases are summarized in Table 6.3 and Figure 6.3. In total, the search returned 374 papers. After this activity was completed, only papers uncovered by backward snowballing were added to the SLR.

Table 6.2: Research database queries

Database	Date	Results	Query string
ACM	2016/07/01	18	{acmdlTitle:(Complexity Metric Metrics) AND acmdlTitle:(Process Processes Workflow EPC BPEL BPMN BPM YAWL) AND recordAbstract:(complexity complex) AND recordAbstract:(process processes workflow) AND recordAbstract:(measure measures measuring metric metrics measurement)} filter: {“publicationYear”:{“gte”:1996, “lte”:2016}}

Continued on next page

Table 6.2 – *Continued from previous page*

Database	**Date**	**Results**	**Query string**
Scopus	2016/07/02	97	((TITLE ((complexity OR metric) AND (process OR workflow OR epc OR bpel OR bpmn OR bpm OR yawl)) AND ABS ((complexity OR complex) AND (process OR workflow) AND (measure OR measuring OR metric OR measurement)) AND LANGUAGE (english)) AND PUBYEAR > 1995 AND PUBYEAR < 2017) AND (complexity) AND (EXCLUDE (SUBJAREA , "ENVI") OR EXCLUDE (SUBJAREA , "PHYS") OR EXCLUDE (SUBJAREA , "EART") OR EXCLUDE (SUBJAREA , "BIOC") OR EXCLUDE (SUBJAREA , "CENG")) AND (EXCLUDE (SUBJAREA , "CHEM") OR EXCLUDE (SUBJAREA , "AGRI") OR EXCLUDE (SUBJAREA , "MEDI") OR EXCLUDE (SUBJAREA , "PSYC"))
Web of Science	2016/07/02	36	TITLE: ((Complexity OR Metric*) AND (Process* OR Workflow OR EPC OR BPEL OR BPMN OR BPM OR YAWL)) AND TOPIC: (complex* AND (process* OR workflow) AND (measur* OR metric*)) Refined by: PUBLICATION YEARS: (2012 OR 2004 OR 2010 OR 1997 OR 2011 OR 2006 OR 2005 OR 2008 OR 2000 OR 2003 OR 2002 OR 2016 OR 2014 OR 2001 OR 2015 OR 2007 OR 1998 OR 2013 OR 1999 OR 2009) AND WEB OF SCIENCE CATEGORIES: (COMPUTER SCIENCE SOFTWARE ENGINEERING OR MULTIDISCIPLINARY SCIENCES OR COMPUTER SCIENCE INFORMATION SYSTEMS OR COMPUTER SCIENCE THEORY METHODS OR MATHEMATICS APPLIED OR COMPUTER SCIENCE ARTIFICIAL INTELLIGENCE OR OPERATIONS RESEARCH MANAGEMENT SCIENCE OR ENGINEERING MULTIDISCIPLINARY OR COMPUTER SCIENCE CYBERNETICS OR ENGINEERING INDUSTRIAL OR COMPUTER SCIENCE INTERDISCIPLINARY APPLICATIONS OR MATHEMATICS INTERDISCIPLINARY APPLICATIONS OR SOCIAL SCIENCES MATHEMATICAL METHODS)
IEEE	2016/07/03	35	(("Document Title":Complexity OR "Document Title":Metric*) AND (p_Title:Process OR "Document Title":Processes OR "Document Title":Workflow OR "Document Title":EPC OR "Document Title":BPEL OR "Document Title":BPMN OR "Document Title":BPM OR "Document Title":YAWL)) metric* complex* and refined by Year: 1996-2016
IEEE	2016/07/03	10	(("Document Title":Complexity OR "Document Title":Metric*) AND (p_Title:Processes OR "Document Title":Process OR "Document Title":Workflow OR "Document Title":EPC OR "Document Title":BPEL OR "Document Title":BPMN OR "Document Title":BPM OR "Document Title":YAWL)) complex* measur* and refined by Year: 1996-2016
Springer	2016/07/03	9	(Process | Workflow | EPC | BPEL | BPMN | BPM | YAWL) & (measur* | metric) & complexity & define & calculate within English Business & Management Computer 1996 - 2016. NOTE: Springer does not provide facilities to search title or abstract. So title search was done manually. It also does not provide a way to download a set of citations, requiring this to be done manually as well.

Continued on next page

Table 6.2 – *Continued from previous page*

Database	**Date**	**Results**	**Query string**
Science Direct	2016/07/03	11	pub-date > 1995 and TITLE((Complexity OR Metric) AND (Process OR Workflow OR EPC OR BPEL OR BPMN OR BPM OR YAWL)) and ABSTRACT(complex* AND (process OR workflow) AND (measur* OR metric))[All Sources(Business, Management and Accounting,Computer Science,Decision Sciences,Engineering,Mathematics)].
Google Scholar	2016/07/03	116	(intitle:Complexity OR intitle:Metric*) (intitle:Processes OR intitle:Process OR intitle:Workflow OR intitle:EPC OR intitle:BPEL OR intitle:BPMN OR intitle:BPM OR intitle:YAWL) complexity (metric OR measure) Custom range: 1996 to 2016. Note: Google Scholar does not allow abstract searches and does not download all of the citations. As a result this was done manually.
Backward snowballing	various	40	Papers cited by other papers, and identified during the review.

Table 6.3: Search results

Engine	**Discovered**	**Relevant**
ACM	18	1
Google Scholar	116	16
IEEE	45	2
Science Direct	11	0
Scopus	97	25
Snowballing	42	29
Springer	9	1
Web of Science	36	8
TOTALS	374	82

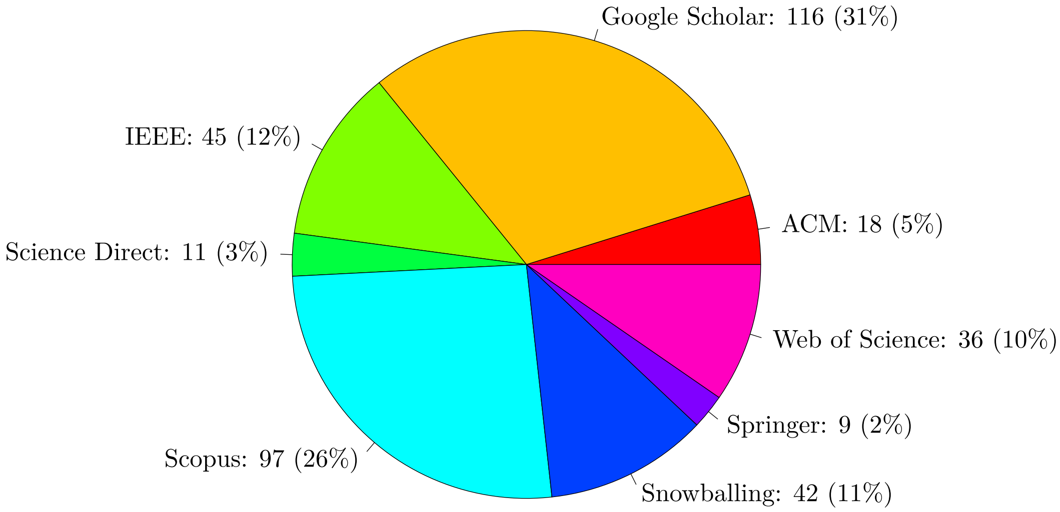


Figure 6.3: Distribution of search results per research database

The selection activity was conducted after having performed the search activity, which consisted of applying the inclusion and exclusion criteria described in Figure 6.2. During this activity the abstract of each paper was evaluated against the inclusion and exclusion criteria. If the abstract was not detailed enough to apply the criteria, then the full paper was retrieved and reviewed. The language criteria affected four documents written in a different language (one in Spanish, one in Portuguese, two in Chinese). Fortunately some of the authors had similar papers written in English. The availability criteria affected three theses from two Malaysian Universities. An inter-library loan was requested in order to gain access to these theses, but UNISA's librarians were unable to secure the documents. One of the two universities explained that "based on university policy all the theses are restricted for our institutional members only" (see Appendix D file 29 (SLR-Response from University Putra Malaysia.pdf)). The outcome of this activity was to divide the papers into three categories, namely accepted, duplicated, and rejected. Only accepted papers were used in the next activity in this review. The outcome of this activity was recorded in the StArt tool. Figure 6.4 shows the results of the selection activity, where 82 relevant papers were accepted. This activity was conducted for each new paper that was discovered via backwards snowballing.

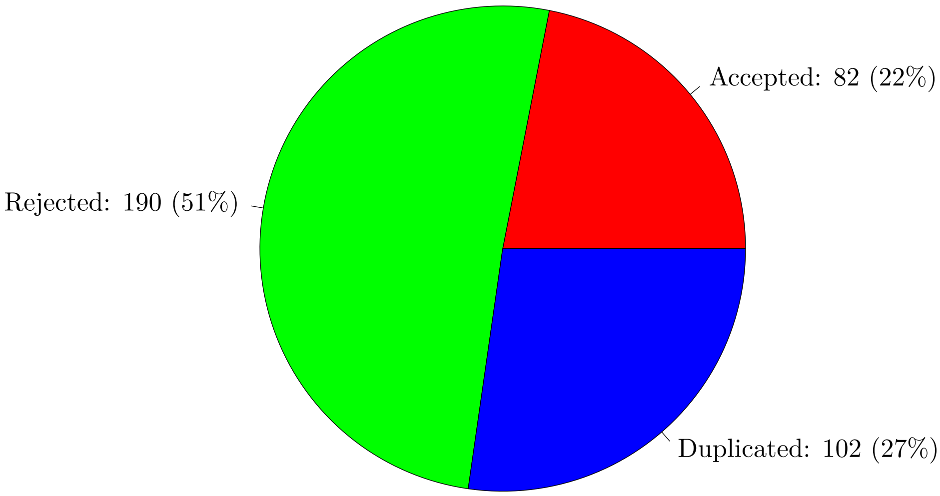


Figure 6.4: Results of selection phase

The data extraction activity used the accepted papers as input. The papers were read and data was extracted. During the first step in this activity the data was recorded in the StArt tool and the papers were classified as primary, secondary, duplicated, survey, and uses. Figure 6.5 shows the classification. The second step in this activity was to extract further information from the primary and secondary papers. During this step the data was recorded in a Microsoft Spreadsheet because the StArt tool was not able to maintain the three levels of granularity that were required (i.e., papers, validations, and metrics). In this step, data was recorded at the paper level, validation level, and metric level. Most papers contained more than one metric, and some papers contained more than one validation procedure.

Table 6.4: Number of items per level

Levels	**Items**	**Count**	
Papers	Primary papers	40	
	Secondary papers	17	
	Totals	57	
Validation Studies	Theoretical validations	14	in 10 papers
	Empirical validations	45	in 32 papers
	(Human validation 27)		
	(Software validation 18)		
	Totals	59	
Metrics	non-duplicated metrics	206	
	duplicated metrics	75	
	Totals	281	

The primary and secondary papers were used for the data analysis activity. During this activity the extracted data was analyzed and processed in order to synthesize it into a set of tables and graphs. For this purpose R [R C16] was used and a data analysis report was produced (see Appendix D file 61 (SLR-analysis.Rmd)). Table 6.4 offers a summary of the outcome of this activity. There were 57 papers that were analyzed during the activity, with 59 validation studies described in the analyzed papers and 206 non-duplicated metrics being described in the primary papers. Of these 27 studies used human validation, and the remaining 18 used software validation. Each validation study tried to validate a set of metrics. Theoretical validations were always successful, while some metrics may have failed the empirical validation.

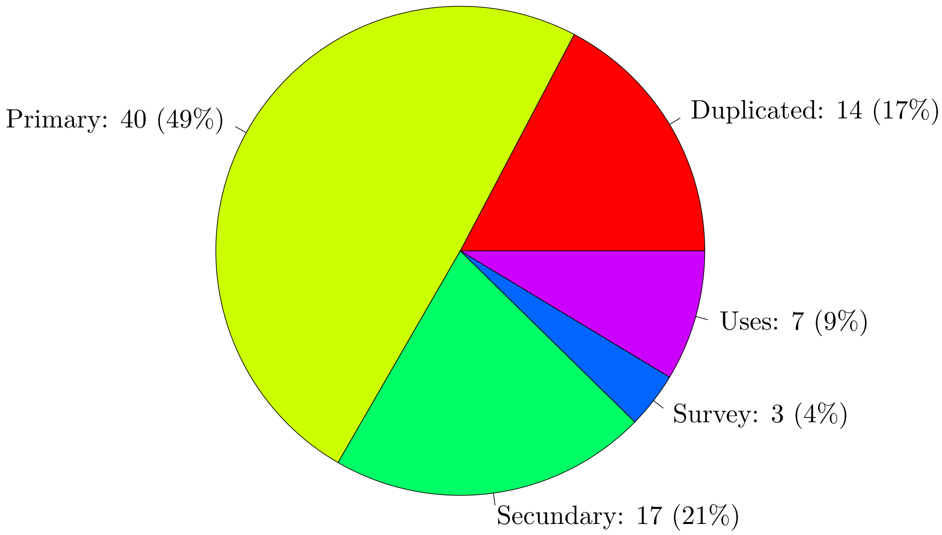


Figure 6.5: Classification during extraction phase

6.4.1 Findings

This section evaluates the information that was extracted during the data extraction activity in order to answer the research questions posed in Section 6.2. In this SLR, the researcher tabulated the data in a quantitative way. However, this SLR did not collect quantitative statistical results from metric validations. It was limited to extracting predetermined information about the metric validations including the authors' validation claims. Therefore, no sensitivity analysis of statistical results was conducted.

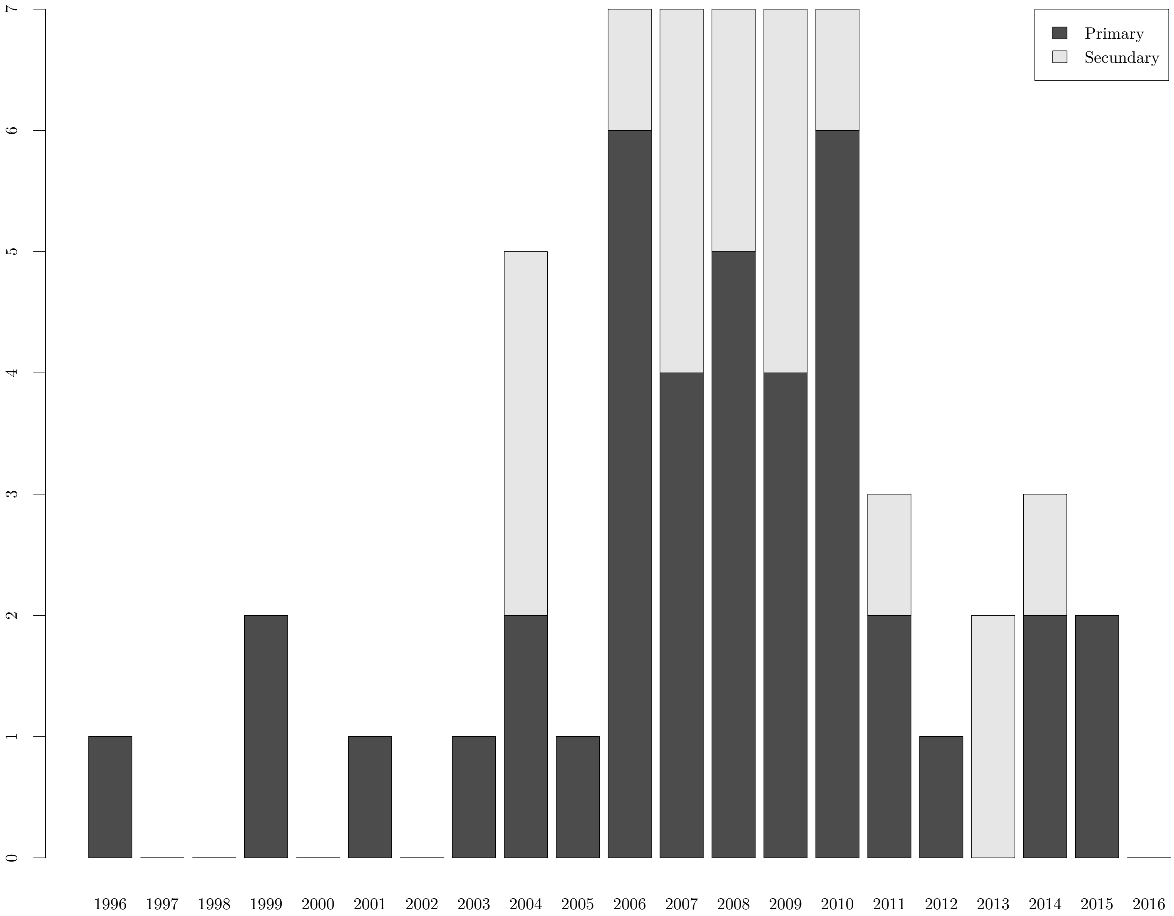


Figure 6.6: Distribution of primary and secondary papers produced during the years examined in the SLR

The distribution of primary and secondary papers, produced during the 20 years covered in the SLR, is shown in Figure 6.6. It shows that there was a focus on complexity metrics for process models during 2006 and 2010, with seven papers being published during each of those years. Figure 6.7 plots the rigor and relevance of the empirical validations in the papers in this review. Both rigor and relevance were measured on a scale of zero to three. Six papers reached three for rigor, but none of the papers reached three for relevance. Relevance included a 'sample size justification' rating. During the review it was found that none of the papers had justified their sample size. Figure 6.7 shows a worrying cluster of low rigor and low relevance studies in the lower left corner, and only three validations in the upper right corner having rigor equal to 3 and relevance greater than or equal to $1^1/_2$. These were produced by García et al. [Gar+04b; Gar+04d].

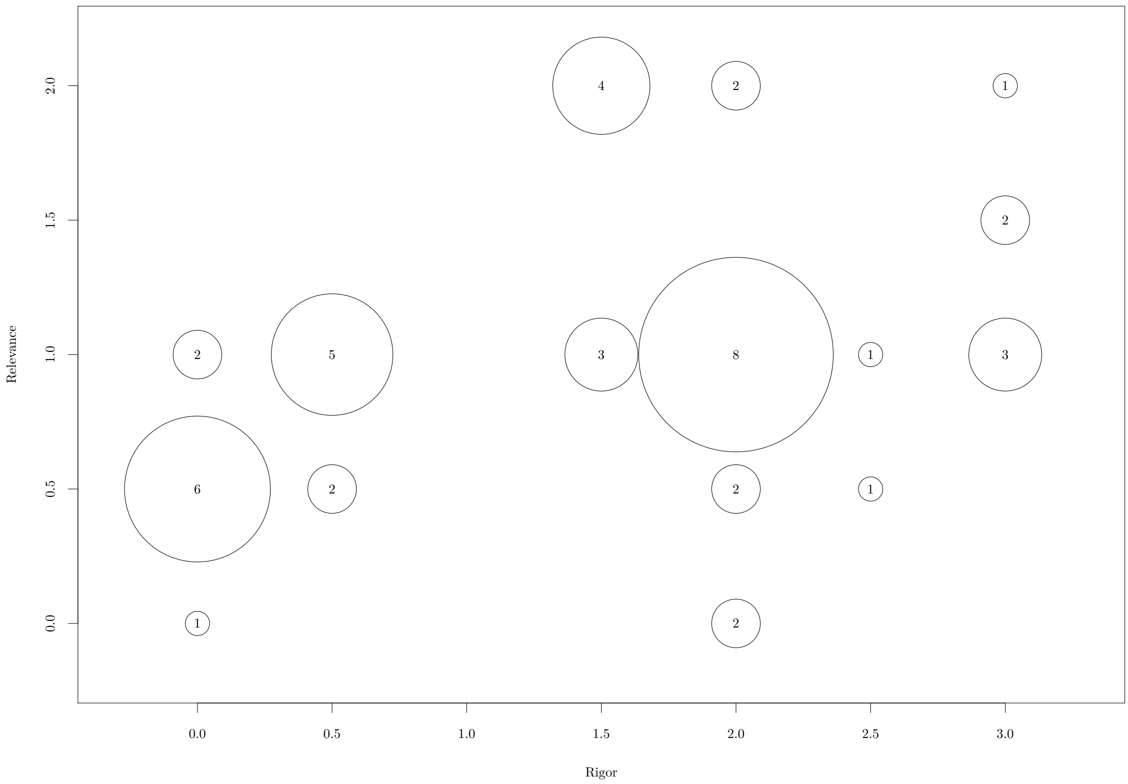


Figure 6.7: Rigor versus Relevance

6.4.1.1 Complexity Metrics Identified in the Literature (RQ1)

Research question **RQ1** was: which static complexity metrics for process models have been identified in the literature? There were 81 papers that were identified, of which 40 were primary papers containing the definitions for new metrics. In total, there were 281 metrics identified during the review of which 206 were non-duplicated metrics. Table B.42 (in Appendix B.2) provides a full inventory of all the metrics identified in this review. During this review the metrics were categorized as shown in Table 6.5, and Figure 6.8 illustrates the distribution of metrics across the different types.

Table 6.5: Type of metric calculations used in complexity metrics for process models

Category	**Type of metric calculation**	**Sample metrics**
Counter	These metrics are calculated by counting elements or types of icons in a process model.	`NA`, `MaxND`, $\texttt{AND}_j$
Ratio	These metrics are calculated by dividing one metric by another.	`MM`, $\texttt{CNC}_P$, `CA`

Continued on next page

Table 6.5 – *Continued from previous page*

Category	Type of metric calculation	Sample metrics
Weighted	These metrics are calculated by assigning weights to certain element types of a process model, and by multiplying the weights by the number of elements of that type.	$\texttt{CC}_{\texttt{YAWL}}$, CCBP, CADAC
Algorithm	The calculation of the metric is defined by an algorithm that given a process model returns the value of the metric.	Λ, SM, CI
Percentage	These metrics are calculated as a percentage of some of the elements in the model.	Flexibility, BAD, EAC
Average	These metrics are calculated as an average of some of the elements in the model.	MeanND, $\overline{\mathtt{d}_C}$, ADP
Calculated	These metrics are calculated using an algebraic formula that cannot be categorized into any of the other type of metrics described in here.	c, k, CFC

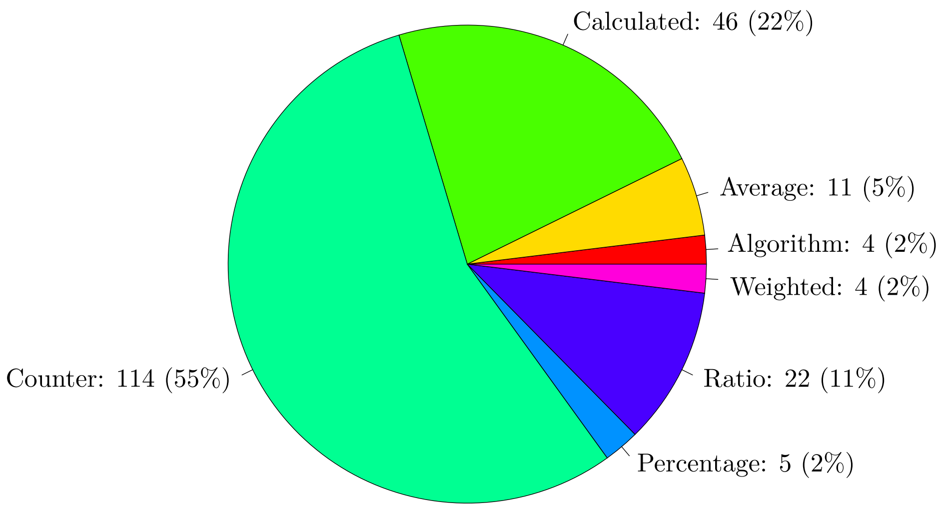


Figure 6.8: Type of validation per Metric

Notations (RQ1.1)

Research question **RQ1.1** was: which notations have been used in the literature to identify complexity metrics for process models? Figure 6.9 shows the notations that were used in papers to define metrics or validate metrics. The objective was to identify declarative process model notations, or data-centric process model notations, that had been previously used because these types of metrics could be easily adapted to CMMN. However, as shown in the figure, all of the notations uncovered during this SLR were for imperative process models. The

Business Process Model and Notation (BPMN) [OMG13], generic graph, and Event-driven Process Chain (EPC) [van99] were the most common notations used to define and validate metrics.

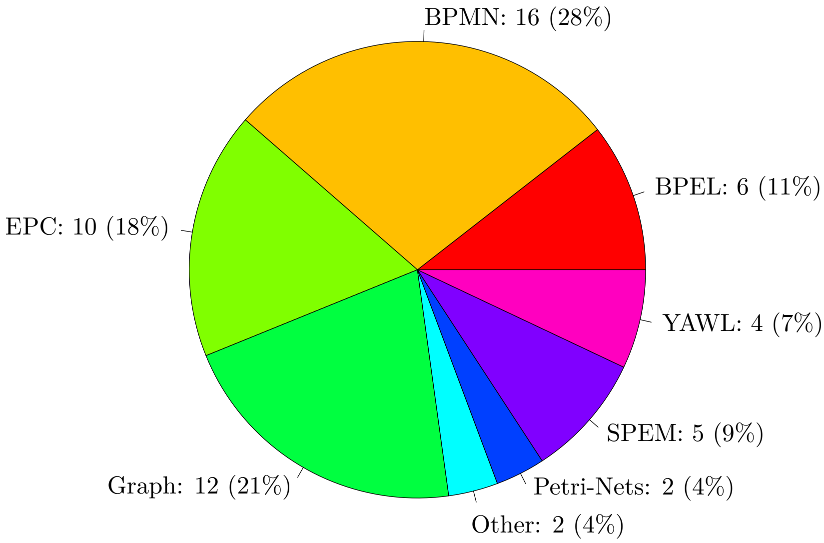


Figure 6.9: Notations for process models used in the primary and secondary papers

6.4.1.2 Research Methods used to Validate Complexity Metrics (RQ2)

Research question **RQ2** was: which types of research methods, present in the literature, have been used to validate complexity metrics for process models? As expected (see Section 6.1.3), the validation methods fell into theoretical validation and empirical validation (see Figure 6.1). Therefore, two inventories were collected, one for theoretical validation and one for empirical validation. Table 6.6 shows all of the theoretical validation frameworks and the papers that used these in order to conduct theoretical validation. Table 6.7 presents the research methods used to empirically validate complexity metrics for process models. This table was created by collecting a description of each research method described in the primary and secondary papers that claimed to have empirically validated complexity metrics, and then clustering and labeling the different approaches. During the SLR it was noticed that metrics that were validated using intuition or anecdotal validations never failed that validation, therefore, some of the analyses performed on the empirical validations were separated into intuition validation (containing both intuition and anecdotal) and empirical validation (containing the rest).

Table 6.6: Theoretical validation frameworks used to validate complexity metrics for process models

Framework	**Papers**
Weyuker [Wey88]	[Car05b; Car08; Çoş14; Fu+10; HZ09; HB09; Muk+10b; SH14]
Briand et al. [Bri+96]	[Çoş14; Muk+10b; Rol09]
DISTANCE [PD00]	[Rol09] *apud* [Rol+06b][1]
Kaner and Bond [KB04]	[Çoş14]
Morasca [Mor99; Mor08]	[Ant+11]

Table 6.7: Research methods used to empirically validate complexity metrics for process models

Category	**Empirical validation design**	**Sample papers**
Anecdotal	Presents few versions of the process models and asks experts for their opinion (either the best process model, or best metric).	[Dan+96; Kho+09; KN12]
Intuition	Same as anecdotal, but instead of experts uses the researcher's intuition or heuristics with no strong justification or validation.	[Muk+10b; RV04; Van+08a]
Metric correlation	Correlates metrics against themselves. No distinction between independent and dependent variables.	[Bor+09a; Lv09; Mäe11]
Comparing against measuring stick	Compares the metrics or the process models against a known entity (for example compared against the best of class [best practice], or against source code implementation metrics, a set of criteria, or against other processes).	[Dan+96; Lat01; Sol+13a]

Continued on next page

[1][Rol+06b] mention the theoretical validation that was conducted in [Rol09] which is not in this SLR because does not meet the language inclusion criteria.

Table 6.7 – *Continued from previous page*

Category	**Empirical validation design**	**Sample papers**
Comparing against other metrics	Similar to metric correlation, with two distinctions: first, compares new metrics against known metrics; secondly, may not use correlations for the comparison. In addition, there is no discussion about the validity of the metrics used for comparison. Fenton and Pfleeger [FP98] warn against this type of validation unless the metrics being used for the correlation have been thoroughly validated for exactly the same attributes as the new metrics.	[Çoş14; Klu+14; SH14]
Human	Uses human subjects, but does not describe the type of experimental design used.	[Mäe11; Men+07c; S+́10b]
Error prediction	Uses logistic regression to identify metric sets that can predict a binary error variable.	[Men06; Men+07c; Van+08a]
Within-subjects	Uses a within-subjects experimental design with human subjects.	[Gar+04b; Men+07c; Van+08a]
Online-survey	Uses an online survey	[MS08]

6.4.1.3 Validating Complexity Metrics (RQ3)

Research question **RQ3** was: how have the complexity metrics for process models described in the literature been validated? The motivation was to understand the established validation strategies that could be used to validate potential CMMN complexity metrics. At the paper level, Figure 6.10 shows how papers have addressed the validation of metrics. At the metric level, Figure 6.11 shows how many metrics have been subjected to validation. Table 6.8 presents the theoretical validations used and how many metrics have been validated. Table 6.9 presents the empirical validations used and how many metrics have been subjected to validation. Theoretical validation is conducted on metrics that follow measurement theory to ensure that they have good properties, therefore all of the metrics subjected to theoretical validations, found in this review, were validated. However, metrics subjected to empirical validation may fail the validation. Table 6.10 summarizes the type of validation and the number of metrics that have undergone it.

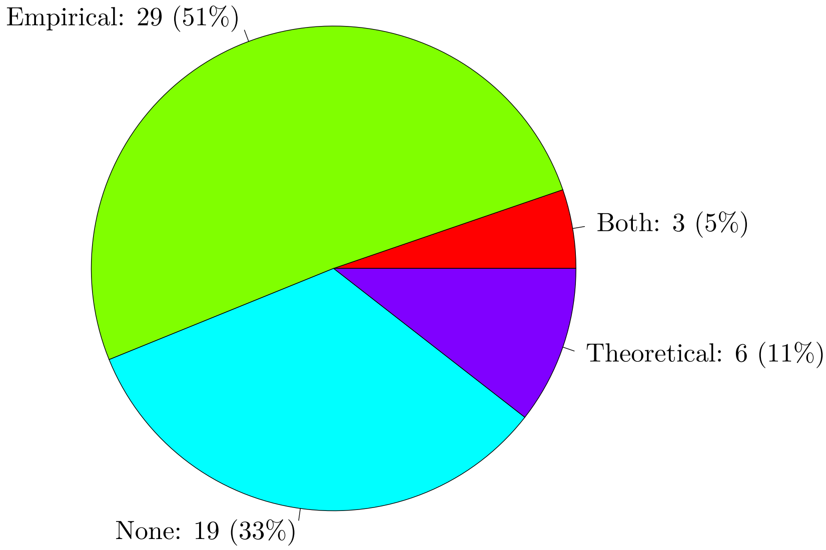


Figure 6.10: Type of validations reported in the primary and secondary papers

Table 6.8: Number of metrics that have been theoretically validated

Validation	**Count**	**Percentage**
Briand+Kaner+Weyuker	1	0.49
Morasca+Weyuker	2	0.97
Briand+DISTANCE+Weyuker	3	1.46
Morasca	4	1.94
Briand+Weyuker	5	2.43
Weyuker	6	2.91
Briand+DISTANCE	57	27.67
None	128	62.14

Table 6.9: Number of metrics that have been empirically validated

Validation	**Count**	**Percentage**
Against metrics, Intuition	1	0.49
Against others, Error prediction, Human experiment	1	0.49
Against others, Error prediction, Human experiment, Within-Subjects	1	0.49
Against others, Intuition	1	0.49
Anecdotal	1	0.49
Error prediction, Human experiment, Online Survey, Within-Subjects	1	0.49
Intuition, Within-Subjects	2	0.97
Against others, Anecdotal	3	1.46
Against metrics	4	1.94
Error prediction, Human experiment, Within-Subjects	4	1.94

Against others	5	2.43
Error prediction	5	2.43
Error prediction, Human experiment, Online Survey	7	3.40
Metric correlation	7	3.40
Intuition	15	7.28
Error prediction, Human experiment	17	8.25
Within-Subjects	57	27.67
None	74	35.92

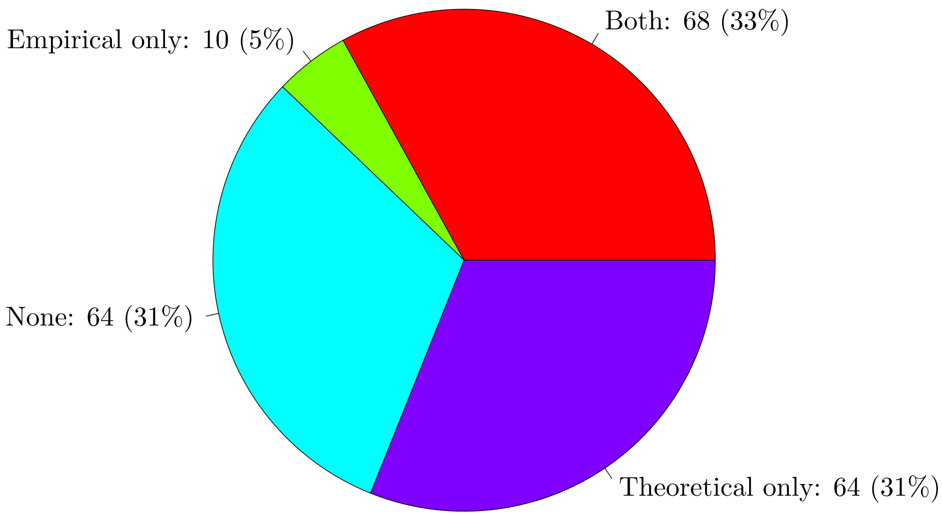


Figure 6.11: Number of metrics that have been validated

Table 6.10: Summary of validation type per number of metrics

Type of validation			Number of metrics
Metrics that have undergone empirical validation			132
Intuition and anecdotal		17	
Other empirical validations		115	
Validated at least once	68		
Never validated	47		
Totals	115		
Failed validation at least once	108		
Never failed validation	7		
Totals	115		
Metrics that have undergone theoretical validation			78

Which Complexity Metrics have been Validated (RQ3.1)

Research question **RQ3.1** was: which of the complexity metrics for process models, identified in the literature, have been validated? Table 6.11 shows the metrics that have undergone both theoretical and empirical validation. In this table the metrics that have undergone empirical validation were classified into three columns. The failed column indicates how many times a metric has failed an empirical validation, the intuition column indicates metrics that were validated using either intuition or anecdotal validations (see Table 6.6), and the validated column indicates metrics that were successfully validated. Empirical validation of complexity metrics is normally done by using a concept that operationalizes complexity and encodes it as a variable. Therefore multiple concepts may be tested for a particular metric (see Table 54, see Appendix D in file 28 (SLR-analysis.pdf)). In addition, some researchers have conducted a series of controlled validations [Gar+04a; Gar+04b; Gar+04c; Gar+04d; Rol+07a; Rol+08; Rol+09a; Rol+09b; S+́10b]. Thus some metrics have undergone multiple empirical validations. Table 6.11 indicates how many times a metric has been validated successfully and how many times it has failed. None of the metrics in this SLR that were validated using intuition or anecdotal validations failed that validation. It is for this reason that the intuition column was used to record them.

Table 6.11: Metrics that have undergone both theoretical and empirical validations

	Empirically validated			
Metric	**Failed**	**Intuition**	**Validated**	**Theoretically validated**
NA	**34 times**		29 times	Briand, DISTANCE, Weyuker
NDWP	11 times		**27 times**	Briand, DISTANCE
NWP	14 times		**22 times**	Briand, DISTANCE
NEDDB	9 times		**11 times**	Briand, DISTANCE
TNE	**12 times**		10 times	Briand, DISTANCE
TNG	10 times		10 times	Briand, DISTANCE, Weyuker
NSFG	10 times		10 times	Briand, DISTANCE
RDWPOut	**27 times**		9 times	Briand, DISTANCE
NEDEB	**11 times**		9 times	Briand, DISTANCE
TNCS	**12 times**		8 times	Briand, DISTANCE
NCD	**12 times**		8 times	Briand, DISTANCE
NDOOut	**12 times**		8 times	Briand, DISTANCE
NPF	**12 times**		8 times	Briand, DISTANCE
TNEE	**16 times**		7 times	Briand, DISTANCE
NTSE	**16 times**		7 times	Briand, DISTANCE
NID	**13 times**		7 times	Briand, DISTANCE
NTIE	**13 times**		7 times	Briand, DISTANCE
NPR	**30 times**		6 times	Briand, DISTANCE
RPRA	**30 times**		6 times	Briand, DISTANCE

Continued on next page

Table 6.11 – *Continued from previous page*

Metric	Empirically validated			Theoretically validated
	Failed	**Intuition**	**Validated**	
NP	**14 times**		6 times	Briand, DISTANCE
NSFA	**14 times**		6 times	Briand, DISTANCE
NSFE	**14 times**		6 times	Briand, DISTANCE
RDWPIn	**31 times**		5 times	Briand, DISTANCE
CLP	**15 times**		5 times	Briand, DISTANCE
CLA	**16 times**		4 times	Briand, DISTANCE
NDOIn	**16 times**		4 times	Briand, DISTANCE
NMF	**16 times**		4 times	Briand, DISTANCE
CFC	4 times		4 times	Morasca, Weyuker
NCS	**7 times**		3 times	Briand, DISTANCE
NEMsE	**7 times**		3 times	Briand, DISTANCE
NIMsE	**7 times**		3 times	Briand, DISTANCE
RWPA	**24 times**		2 times	Briand, DISTANCE
NITE	**9 times**		1 time	Briand, DISTANCE
CADAC			1 time	Briand, Kaner, Weyuker
S			1 time	Weyuker
IF4BP			1 time	Briand, Weyuker
ACCSA		1 time		Briand, Weyuker
ALSA		1 time		Briand, Weyuker
CCBP		1 time		Briand, Weyuker
SCBP		1 time		Briand, Weyuker

How many Subjects have been used to Validate Complexity Metrics (RQ3.2)

Research question **RQ3.2** was: how many human subjects are used during the validation of complexity metrics for process models? Empirical validations uncovered by this SLR have been conducted using human subjects or software experiments. For human subjects, of the 22 validation studies using primary data, the maximum number of subjects used, as evidenced by García et al.'s [Gar+04a] study, was 86. Four of the reported validations in two of the papers [Rol+09a; S+́10b] used secondary data that combined multiple experiments, with the number of reported combined subjects being between 56 and 115. Figure 6.12 shows the frequency distribution of the number of human subjects used in empirical validations using primary data for the validation of complexity metrics for process models. However, none of the studies that used human subjects for empirical validation justified the sample size. Therefore, these numbers cannot be considered a good indication of the actual numbers of human subjects required to validate complexity metrics.

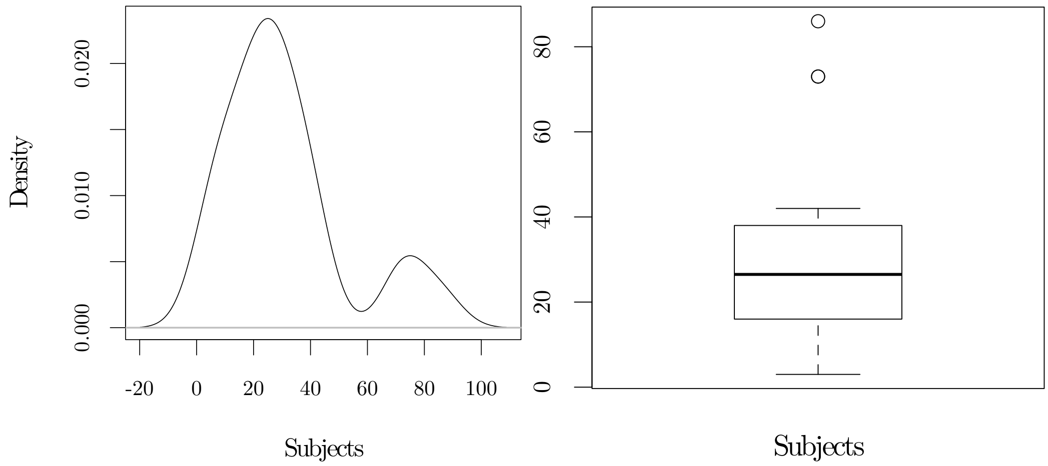


Figure 6.12: Number of human subjects being used for empirical validation

6.5 Discussion

This SLR confirmed Polančič and Cegnar's [PC16] findings that very few metrics have undergone both theoretical and empirical validations, and that a large number of metrics have not been validated at all. Sánchez-González et al. [S+10a] also found that there was a tendency to propose metrics without empirical support. Polančič and Cegnar found that 3% (two out of 66) of metrics had been validated both theoretically and empirically, while this review found that 19% (40 out of 206) had undergone both types of validations. Polančič and Cegnar found that 41% of metrics had not been validated, while this review found that 31% of proposed metrics had not been validated. This review also found that 45% of primary papers introduced metrics without any validation.

The main objective of the research questions was to identify the best practices described in the literature of complexity metrics for process models that could be used to create and validate metrics for CMMN. For this purpose several inventories were created:

- An inventory of metrics was created (see Table B.42) containing 206 non-duplicated metrics. No complexity metrics for declarative process models or for data-centric process models were uncovered during the creation of the inventory of metrics.
- An inventory of process modeling notations used when proposing or validating metrics was created (Figure 6.9). All of the notations used in primary and secondary papers were for imperative process models.
- An inventory of theoretical validation frameworks used in the literature (see Table 6.6) was created.
- An inventory of research methods used in the literature to empirically validate complexity metrics for process models (see Table 6.7) was created.

- An inventory of metrics that have undergone both theoretical and empirical validations (see Table 6.11) was created.
- An inventory of concepts that have been used to empirically validate complexity metrics for process models (see Table 54, see Appendix D in file 28 (SLR-analysis.pdf)) was created.

6.5.1 Principal Findings

This section identifies and outlines the principal findings of this review as follows:

Large number of proposed metrics without validation. As described above (see Figure 6.1) there is consensus that metrics must undergo both theoretical and empirical validations [Cal+01; Muk+10a; PC16; Ser+02; Son09]. However, this review found that only 31% of the proposed metrics had undergone both theoretical and empirical validations. In addition, a large number of metrics were never validated (also 31%). This finding corresponds with a similar finding in the research done by Sánchez-González et al. [Ś+10a] indicating that there is a tendency to propose metrics without any empirical support.

No complexity metrics for declarative process models have been proposed. This SLR did not uncover any complexity metrics for declarative process models. All of the metrics were for imperative process models.

There is no agreement on research methods for empirical validation. Nine research methods for empirical validation were uncovered by this review (see Table 6.7). However, the research methods used by most papers were not well established research designs. With the exception of within-subject experimental design and survey experimental design, all of the other research methods used to empirically validate complexity metrics were not well-known research methods.

Some consensus on theoretical validation is emerging. Most authors conducting theoretical validation have used Weyuker's [Wey88] nine properties, followed by Briand et al.'s [Bri+96] framework (see Table 6.6). However, only 14 theoretical validation studies (versus 45 empirical validation studies) were uncovered (see Table 6.4).

Insufficient guidance on sample size for empirical validation was uncovered. Of the 27empirical validation studies using human subjects, none justified the sample size used.

Low rigor and relevance of most empirical validations. Of the 45 empirical validations uncovered by this review, only nine were located in the upper right quadrant (with both rigor and relevance greater than or equal to $1^1/_2$). This is partly because none of the papers describing empirical validations justified the sample size used. This may also be the case because most of the empirical validations were done using a convenience sample of students (more than 74% of all the validations using humans) or models.

6.5.2 Strengths and Weaknesses

To ensure the reliability of this review internal and external threats to validity were considered during the design phase of this SLR.

Internal validity threats to an SLR include researcher bias and how well the review answers the research questions. Threats to internal validity were minimized by designing a protocol based on the guidelines outlined by Biolchini et al. [Bio+05], Kitchenham [Kit04], Kitchenham and Charters [KC07], Kitchenham et al. [Kit+04], Sjoberg et al. [Sjo+07], and Wohlin et al. [Woh+12]. However, this review was conducted by a single researcher, and even though steps were taken to eliminate bias, it cannot be completely ruled out. Opportunities for researcher bias may have been present when rejecting or classifying borderline papers.

The search strategy and the inclusion and exclusion criteria were designed during the planning phase, and so should minimize threats to internal validity. There is the potential that important papers on the area of complexity metrics for process models may not have been uncovered by this review, because these were present in research databases not included in this SLR. This threat was minimized by using a diverse set of seven research databases, and by using a backward snowballing approach to uncover papers not uncovered during the initial search.

External validity threats to an SLR include how applicable the findings of the review are to the research question. The findings of this review are consistent with those of Polančič and Cegnar [PC16] who conducted an SLR of complexity metrics for process models. Although, Polančič and Cegnar [PC16] only addressed one of the questions contained within this review (**RQ1** above), it must be noted that their SLR focused on a ten years period (from 2005 to 1 February 2015), and found only 66 complexity metrics for process models. This review also identified the 66 metrics identified by Polančič and Cegnar as part of the 281 identified metrics. The purpose for conducting this SLR was to identify metrics and research methods that could be adapted to create and validate complexity metrics for CMMN. The findings of this review appear to have served its expected purpose, which was to offer a basis for identifying complexity metrics for CMMN, as well as to identify research methods to validate these metrics.

As described by Kitchenham and Charters [KC07] any conflict of interest on the part of the researcher should be disclosed. This researcher had no conflict of interest with this SLR, as stated in the application for ethical clearance from UNISA (see Appendix D file 15 (2016-05-23 MAMarin_Student_Ethical_Clearance-v5.pdf)).

6.6 Summary

This chapter contributes a current SLR of research into complexity metrics for process models to the general body of knowledge in the area of complexity metrics for process models. This review was designed to identify metrics and research methods that could be applied to CMMN. The review identified complexity metrics for process models featured in published research conducted over the last 20 years (from the beginning of 1996 to the middle of 2016), and how these were validated. The SLR followed the guidelines for software engineering literature reviews created by Biolchini et al. [Bio+05], Kitchenham [Kit04], Kitchenham and Charters [KC07], Kitchenham et al. [Kit+04], Sjoberg et al. [Sjo+07], and Wohlin et al. [Woh+12], and adapted the quality evaluation of rigor and relevance described by Bin Ali et al. [Bin+14], Dybå and Dingsøyr [DD08], and Vasconcellos et al. [Vas+17] for its purposes. The information gathered in this SLR forms the basis for Chapters 7 and 8.

Chapter 7

Metrics for Case Management

This chapter analyzes the applicability of the process modeling complexity metrics that were identified in Chapter 6 for Case Management Model and Notation (CMMN) [OMG14a]. The chapter contributes a set of CMMN metrics and sub-metrics to the broader knowledge base and also provides the theoretical validation of the proposed metrics. An evaluation of the method complexity of the CMMN notation was conducted in Chapter 4, and it was concluded that it compares favorably against other methods. Expanding on the work done in Chapter 6, this chapter goes one step further and defines process modeling complexity metrics for CMMN. A Systematic Literature Review (SLR) was conducted in Chapter 6 that did not uncover any complexity metrics for declarative process models that could be directly applied to CMMN. This chapter fills this that gap in the literature by defining process modeling complexity metrics for CMMN. The formalization of metrics in this chapter is based on the CMMN formalization that was discussed in Section 3.3. The complexity metrics for CMMN proposed in this chapter will be empirically validated in Chapter 8. Material from this chapter was previously published in Marin et al. [Mar+15b].

This chapter is organized as follows. Section 7.1 uses the results from the SLR conducted in Chapter 6 to identify metrics that can be adapted to CMMN. Section 7.2 defines a set of proposed metrics for CMMN. Material from this section has been published in Marin et al. [Mar+15b]. Section 7.3 validates the proposed CMMN metrics using the formal framework for software measurements as defined by Briand et al. [Bri+96], and the properties for software complexity measures as defined by Weyuker [Wey88].

7.1 Applicability of Current Process Metrics to Case Management Modeling and Notation

The SLR conducted in Chapter 6 did not uncover any complexity metrics for declarative process models or for data-centric process models that could be applied directly to CMMN. All of the notations for process models used in the primary and secondary papers in the review were for imperative process models (see Figure 6.9). The literature review identified 206 non-duplicated process modeling complexity metrics (see Table 6.4). It was possible that some of the metrics that were uncovered by the review could be adapted for use in CMMN. Table 55, see Appendix D in file 28 (SLR-analysis.pdf) present the analysis that was conducted during this review to identify metrics that could potentially be used as a basis for complexity metrics for declarative process models. An analysis of each metric was conducted in order to identify suitable metrics for CMMN (which is a declarative process model) where all of the metrics were analyzed and classified into 14 clusters that could be used to create metrics for declarative process models (see Table 7.1).

The analysis presented in Table 7.1 informs the CMMN metrics that are proposed in Section 7.2. However, not all of the suggested metrics in Table 7.1 were used because only metrics that were thought to have a good possibility of capturing the complexity of CMMN were proposed in Section 7.2 . Most of the applicable metrics were counters, followed by cognitive complexity metrics where a set of weights was used to calculate the metric. The SLR identified several cognitive complexity metrics including [Çoş14; GL06a; GL06b; SW03]. These cognitive complexity metrics are not directly applicable to CMMN because they are based on control structures [Fig+10] like sequence, branching, iterations, etc., which are common in imperative process models but not present in CMMN. However, weights could be given to other elements in the model, based on how complex those elements looked to an observer.

Table 7.1: Process metrics that could be adapted to CMMN

Cluster	**Metrics**	**Potential suggestion**
Case count	`NSP` ([Abr+10] see Table B.34)	Count case tasks (type: Counter)
Collapsed stages	`NCS` ([Rol+06b] see Table B.9), `TNCS` ([Rol+06b] see Table B.9)	Count collapsed stages (type: Counter)
Data count	`NDO` ([Abr+10] see Table B.34), `TNDO` ([Rol+06b] see Table B.9)	Count data objects (type: Counter)
Durfee square	`DSM` ([KN12] see Table B.37)	Adapt Durfee square metric (type: Calculated)

Continued on next page

Table 7.1 – *Continued from previous page*

Cluster	**Metrics**	**Potential suggestion**
Event count	TNE ([Abr+10] see Table B.34), S_E ([Men07] see Table B.16), TNE ([Rol+06b] see Table B.9)	Count events (type: Counter)
Halstead	D ([Car+06] see Table B.10), N ([Car+06] see Table B.10), V ([Car+06] see Table B.10)	Adapt Halstead's primitive measures (type: Calculated)
Hierarchy depth	Depth ([La +11b] see Table B.35), HH ([Kre10] see Table B.33)	Depth of stage hierarchy (type: Counter)
Hierarchy width	WH ([Kre10] see Table B.33)	Width of stage hierarchy (type: Counter)
Modeling concepts	DMC ([La +11b] see Table B.35)	Count modeling concepts used in the model (type: Counter)
Perfect square	PSM ([KN12] see Table B.37)	Adapt perfect square metric (type: Calculated)
Stage count	TSAC ([Lï5] see Table B.40)	Count stages (type: Counter)
Task count	NOA ([Car+06] see Table B.10), NA ([Abr+10] see Table B.34), TNT ([Abr+10] see Table B.34), NOBA ([Muk+10b] see Table B.30), NA ([GL06b] see Table B.11), TBAC ([Lï5] see Table B.40), AC ([Car05b] see Table B.8), S_N ([Men07] see Table B.16), TNA ([Rol+06b] see Table B.9), NT ([Rol+06b] see Table B.9), TNT ([Rol+06b] see Table B.9), $Size_A$ ([Ant+11] see Table B.36), NOA ([HB09] see Table B.28), NN ([Kre10] see Table B.33), NA ([Gar+03] see Table B.5)	Count tasks (type: Counter)
Unique tasks	NCl ([Kre10] see Table B.33)	Count unique tasks (type: Counter)
Weights	CCBP ([Muk+10b] see Table B.30), CW ([GL06b] see Table B.11), CC_{YAWL} ([GL06a] see Table B.14), CADAC ([Çoş14] see Table B.39)	Assign weight to elements and sum them (type: Weighted)

7.2 Defining Metrics for Case Management Modeling and Notation

This section defines metrics that are consistent with the formal framework for software measurements as defined by Briand et al. [Bri+96]. This section is based on the CMMN formalization described in Section 3.3. Although this thesis is interested in complexity, it also defines size and length metrics because, as concluded by Muketha et al. [Muk+10a], size and length are similar to the complexity activity metrics proposed by Cardoso [Car07b]. The section starts by defining three size metrics, followed by length and complexity metrics.

Definition 7.1. (Size metric): The size of a model $\mathcal{C}$ denoted by $CS(\mathcal{C})$ is defined as the cardinality of $\mathcal{E}$,

$$CS(\mathcal{C}) = |\mathcal{E}|$$

The size of a module $\ulcorner z \urcorner$ is defined as the cardinality of $\mathcal{E}_z$,

$$CS(\ulcorner z \urcorner) = |\mathcal{E}_z|$$

By Equation (3.3.1) it follows that,

$$CS(\mathcal{C}) = |\mathcal{E}| \ = \sum_{z \in \mathcal{L}} |\mathcal{E}_z| \ = \sum_{z \in \mathcal{L}} CS(\ulcorner z \urcorner)$$

Note that CS can be informally calculated by counting the nodes in the forest graph of elements $\mathcal{E}$ of a model $\mathcal{C}$.

Note that,

Size for Figure 3.4 is $CS(example1) = |\mathcal{E}| = 1$.

Size for Figure 3.5 is $CS(example2) = |\mathcal{E}| = 4$.

Size for Figure 3.6 is $CS(example3) = |\mathcal{E}| = 8$.

Size for $\ulcorner x \urcorner$ in Figure 3.6 is $CS(\ulcorner x \urcorner) = |\mathcal{E}_x| = 7$.

Size for $\ulcorner y \urcorner$ in Figure 3.6 is $CS(\ulcorner y \urcorner) = |\mathcal{E}_y| = 3$.

Size for $\ulcorner z \urcorner$ in Figure 3.6 is $CS(\ulcorner z \urcorner) = |\mathcal{E}_z| = 1$.

In addition to the size metric $CS(\mathcal{C})$, a set of size sub-metrics were defined that correspond to the number of different elements in $\mathcal{E}$. Table 7.2 includes these sub-metrics.

Table 7.2: Size counters

Type	**Counter**	**Description**
Scope	$CS_{SC}(\mathcal{C})$	Number of cases
	$CS_{SS}(\mathcal{C})$	Number of stages
	$CS_{SDS}(\mathcal{C})$	Number of discretionary stages
	$CS_{SPF}(\mathcal{C})$	Number of plan fragments
Data	$CS_{DI}(\mathcal{C})$	Number of case file items
Plan	$CS_{PT}(\mathcal{C})$	Number of tasks
	$CS_{PDT}(\mathcal{C})$	Number of discretionary tasks
	$CS_{PE}(\mathcal{C})$	Number of event listeners
	$CS_{PM}(\mathcal{C})$	Number of milestones
Optional	$CS_{OC}(\mathcal{C})$	Number of connectors
Decorator	$CAS_{DCP}(\mathcal{C})$	Number of collapsed planning table decorators
	$CAS_{DEP}(\mathcal{C})$	Number of expanded planning table decorators
	$CAS_{DAC}(\mathcal{C})$	Number of autocomplete decorators
	$CAS_{DC}(\mathcal{C})$	Number of collapsed decorators
	$CAS_{DE}(\mathcal{C})$	Number of expanded decorators
	$CAS_{DMA}(\mathcal{C})$	Number of manual activation decorators
	$CAS_{DRN}(\mathcal{C})$	Number of repetition decorators
	$CAS_{DR}(\mathcal{C})$	Number of required decorators
Sentry	$CAS_{SE}(\mathcal{C})$	Number of entry criteria sentries
	$CAS_{SX}(\mathcal{C})$	Number of exit criteria sentries
Marker	$CAS_{MH}(\mathcal{C})$	Number of non-blocking human markers
	$CAS_{MP}(\mathcal{C})$	Number of process markers
	$CAS_{MC}(\mathcal{C})$	Number of case markers
	$CAS_{MHB}(\mathcal{C})$	Number of participant markers
	$CAS_{MT}(\mathcal{C})$	Number of timer markers

Definition 7.2. (Annotator size metric): The number of annotators in a model $\mathcal{C}$ denoted by $CAS(\mathcal{C})$ is defined as the cardinality of $\mathcal{A}$,

$$CAS(\mathcal{C}) = |\mathcal{A}|$$

We also define counters corresponding to a number of different annotators in $\mathcal{A}$. Table 7.2 includes these counters.

Note that,

Size for Figure 3.4 is $CAS(example1) = |\mathcal{E}| = 0$.

Size for Figure 3.5 is $CAS(example2) = |\mathcal{E}| = 3$.

Size for Figure 3.6 is $CAS(example3) = |\mathcal{E}| = 6$.

Definition 7.3. (Total size metric): The total size of a model $\mathcal{C}$ denoted by $CTS(\mathcal{C})$ is defined as the cardinality of $\mathcal{E}$ plus the cardinality of $\mathcal{A}$,

$$\begin{aligned} CTS(\mathcal{C}) &= |\mathcal{E}| + |\mathcal{A}| \\ &= CS + CAS \end{aligned}$$

Note that,

Size for Figure 3.4 is $CTS(example1) = |\mathcal{E}| = 1$.

Size for Figure 3.5 is $CTS(example2) = |\mathcal{E}| = 7$.

Size for Figure 3.6 is $CTS(example3) = |\mathcal{E}| = 14$.

Definition 7.4. (Length metric): The length of a model $\mathcal{C}$ denoted by $CL(\mathcal{C})$ is defined as the maximum nesting depth of a model. The length $CL(\mathcal{C})$ can be calculated by the following algorithm:

```
function CL(𝒞)
    int m ← 0
    for each case-plan c ∈ ℒ do
        m ← max(m, depth(m, c))
    end for
    return m
end function

function depth(s)
    int d ← 0
    if s ∉ ℳ then
        return 0
    end if
    for all e in scope s do
        d ← max(d, depth(e))
    end for
    return d + 1
end function
```

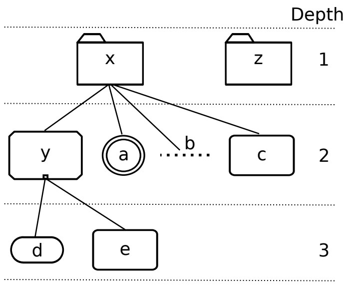


Figure 7.1: Tree view of elements in Example 3

The length of a module $\ulcorner z \urcorner$ is defined as the maximum nesting depth of the module, and can be calculated using the following algorithm:

1: **function** CL($\ulcorner z \urcorner$)
2: **assert**($z \in \mathcal{M}$)
3: **return** depth(z)
4: **end function**

As shown in Figure 7.1, the elements $\mathcal{E}$ of a model $\mathcal{C}$ can be organized as a forest graph where each tree is a case element module. We used the CMMN element icons to represent the nodes of the tree. Figure 7.1 shows a forest with two trees $\ulcorner x \urcorner$, $\ulcorner z \urcorner$, and a subtree $\ulcorner y \urcorner$. It also shows six leaves, namely d, e, a, b, c, and z.

Note that,

Length for Figure 3.4 is $CL(example1) = 1$.

Length for Figure 3.5 is $CL(example2) = 2$.

Length for Figure 3.6 is $CL(example3) = 3$.

Length for $\ulcorner x \urcorner$ in Figure 3.6 is $CL(\ulcorner x \urcorner) = 3$.

Length for $\ulcorner y \urcorner$ in Figure 3.6 is $CL(\ulcorner y \urcorner) = 2$.

Length for $\ulcorner z \urcorner$ in Figure 3.6 is $CL(\ulcorner z \urcorner) = 1$.

Definition 7.5. (Complexity metric): The complexity of a model $\mathcal{C}$ denoted by $CC(\mathcal{C})$ is defined as:

$$CC(\mathcal{C}) = \begin{cases} 0 & \text{if } \mathcal{C} = \varnothing \\ \sum_{i \in \mathcal{E} \cup \mathcal{A}} W_i & \text{if } \mathcal{C} \neq \varnothing \end{cases}$$

Where, the weight W_i, is given in Table 7.3.

The complexity of a model resembles a cognitive complexity metric with cognitive weights. There are several cognitive complexity metrics that have been defined for Business Process

Management (BPM) [Çoş14; GL06a; GL06b; SW03]. However, those cognitive complexity metrics are not applicable to CMMN because they are based on control structures [Fig+10] like sequence, branching, iterations, etc., which are not present in CMMN. For the CMMN complexity metric $CC(\mathcal{C})$, we assign weights to elements in $\mathcal{E}$ and annotators in $\mathcal{A}$. The weights were assigned based on the researcher's intuition. Higher weights were given to less frequently used elements and to annotators that increased the number of elements in the visual canvas because less frequently used elements require better recall capabilities by users and clutter by annotators makes the model more difficult to read.

The complexity of a module $\ulcorner z \urcorner$ is defined as,

$$CC(\ulcorner z \urcorner) = \sum_{i \in \mathcal{E}_z \cup \mathcal{A}_z} W_i$$

Note that,

Complexity for Figure 3.4 is $CC(example1) = 1$.

Complexity for Figure 3.5 is $CC(example2) = 7$
(weights w=1, o=1, p=0, q=1, k=1, l=0, j=3).

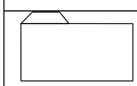
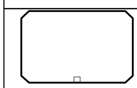
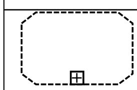
Complexity for Figure 3.6 is $CC(example3) = 11$
(weights z=1, x=1, a=2, b=0, c=1, y=1, d=1, e=1, h=1, i=0, j=0, k=1, l=0, m=1).

Complexity for $\ulcorner x \urcorner$ in Figure 3.6 is $CC(\ulcorner x \urcorner) = 10$
(weights x=1, a=2, b=0, c=1, y=1, d=1, e=1, h=1, i=0, j=0, k=1, l=0, m=1).

Complexity for $\ulcorner y \urcorner$ in Figure 3.6 is $CC(\ulcorner y \urcorner) = 3$
(weights y=1, d=1, e=1).

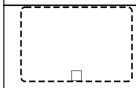
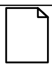
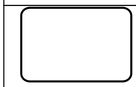
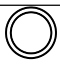
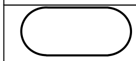
Complexity for $\ulcorner z \urcorner$ in Figure 3.6 is $CC(\ulcorner z \urcorner) = 1$.

Table 7.3: CMMN weights

$\mathcal{E} \cup \mathcal{A}$	**Description**	**Weight**
	case element	1
	stage element	1
	discretionary stage element	2

Continued on next page

Table 7.3 – *Continued from previous page*

$\mathcal{E} \cup \mathcal{A}$	**Description**	**Weight**
	plan fragment element	3
	case file item element	1
	task element	1
	discretionary task element	2
	event listener element	2
	milestone element	1
....	connector (sentry) element	0
	collapsed planning table	1
	expanded planning table	2
■	autocomplete	2
⊞	collapsed	0
⊟	expanded	1
▷	manual activation	1
#	repetition	1
!	required	1
◊	entry criterion with associated connector	1
	entry criterion without a connector	2
◆	exit criterion with associated connector	1
	exit criterion without a connector	3
	non-blocking human	1
	process	0
	case referring to a case element not in this model	0
	case referring to a case element in this model	1
	participant	0
	timer	0

7.3 Theoretical Validation

In this section a theoretical validation of each metric is conducted using Briand et al.'s framework [Bri+96]. The complexity metric is further validated against the nine properties for software complexity measures as defined by Weyuker [Wey88]. Although measurement theory has been questioned by Misra and Kilic [MK07] as a way of evaluating complexity measures, it still serves the purpose of formalizing and validating important mathematical qualities of a complexity metric.

7.3.1 Briand's Framework

Briand et al.'s framework [Bri+96] categorizes software metrics into size, length, complexity, cohesion, and coupling. The framework is based on systems and modules. Briand et al. defined a system S as a pair $S = \langle E, R \rangle$, where E represents a set of elements for S, and R is a binary relationship in E ($R \subseteq E \times E$). A module of S is defined as $m = \langle E_m, R_m \rangle$ for $E_m \subseteq E$, $R_m \subseteq E_m \times E_m$ and $R_m \subseteq R$.

For our purposes a system corresponds to a model $\mathcal{C}$ and a module corresponds to a module ($\ulcorner m \urcorner$, $m \in \mathcal{M}$). E corresponds to modeling elements $\mathcal{E}$ ($E = \mathcal{E}$) as described in Table 3.3. R corresponds to the two binary relationships in $\mathcal{C}$ ($\mathcal{U}$ and $\mathcal{V}$). We define R as follows,

$$R = \{[\![a, b]\!] \mid [\![a, b]\!] \in \mathcal{U}\} \cup \{\langle a, b \rangle \mid \langle a, b \rangle \in \mathcal{V}\}$$

Note that $[\![a, b]\!] \neq \langle a, b \rangle$, $[\![a, b]\!]$ is an unordered pair, while $\langle a, b \rangle$ is an ordered pair.

This section uses the terminology used in Briand et al.'s framework [Bri+96] to introduce each property followed by a short proof against our metrics.

7.3.1.1 Size

Briand et al. define a function $Size(S)$ as being characterized by three properties. This section defines three size metrics, namely $CS(\mathcal{C})$, $CAS(\mathcal{C})$, and $CTS(\mathcal{C})$. In addition, it defines a set of counters that are required to define the three metrics (see Table 7.2). This section validates $CS(\mathcal{C})$ against the three properties defined by Briand et al.

Validation of $CS(\mathcal{C})$

Size 1. Non-negativity. The size of a model $S = \langle E, R \rangle$ is non-negative.

$(S = \langle E, R \rangle) \implies Size(S) \geq 0$

Proof. In terms of Definition 7.1, $CS(\mathcal{C})$ is the cardinality of $\mathcal{E}$ and the cardinality of a set cannot be negative.

$$\therefore\ CS(\mathcal{C}) = |\mathcal{E}| \geq 0$$ □

Size 2. Null value. The size of a model $S = \langle E, R \rangle$ is zero if E is empty.

$$(S = \langle E, R \rangle\ \wedge\ E = \varnothing) \implies Size(S) = 0$$

Proof. By definition, the cardinality of an empty set is zero.

$$\therefore\ (\mathcal{E} = \varnothing) \implies CS(\mathcal{C}) = |\mathcal{E}| = |\varnothing| = 0$$ □

Size 3. Module additivity. The size of a module $S = \langle E, R \rangle$ is equal to the sum of the sizes of two of its modules $m_1 = \langle E_{m1}, R_{m1} \rangle$ and $m_2 = \langle E_{m2}, R_{m2} \rangle$ such that any element of S is in either m_1 or in m_2.

$$(m_1 \subseteq S\ \wedge\ m_2 \subseteq S\ \wedge\ E = E_{m1} \cup E_{m2}\ \wedge\ E_{m1} \cap E_{m2} = \varnothing)$$
$$\implies Size(S) = Size(m_1) + Size(m_2)$$

Proof. Consider a model $\mathcal{C}$ with two disjoint modules $\ulcorner x \urcorner$ and $\ulcorner y \urcorner$ such that each element in $\mathcal{C}$ is either in $\ulcorner x \urcorner$ or in $\ulcorner y \urcorner$, but not both $((\mathcal{E}_x \cap \mathcal{E}_y = \varnothing)\ \wedge\ (\mathcal{E} = \mathcal{E}_x \cup \mathcal{E}_y))$. It follows, based on Equation (3.3.1), that x and y are the only two case elements in $\mathcal{L}$ ($\mathcal{L} = \{x, y\}$), thus $\mathcal{E}$ is partitioned by $\ulcorner x \urcorner$ and $\ulcorner y \urcorner$.

$$\therefore\ CS(\mathcal{C}) = |\mathcal{E}| = |\mathcal{E}_x| + |\mathcal{E}_y| = CS(\ulcorner x \urcorner) + CS(\ulcorner y \urcorner)$$

□

Validation of *CAS*($\mathcal{C}$)

Since space is limited, this section will not repeat the definitions of the three properties of the $Size(S)$ function. It will only use their short descriptions.

Size 1. Non-negativity.

Proof. By Definition 7.2, $CAS(\mathcal{C})$ is the cardinality of $\mathcal{A}$ and the cardinality of a set cannot be negative.

$$\therefore\ CAS(\mathcal{C}) = |\mathcal{A}| \geq 0$$ □

Size 2. Null value.

Proof. In terms of Definition 3.10, each annotator 'a' in $\mathcal{A}$ is related to one and only one element x in $\mathcal{E}$. Therefore, if $\mathcal{E}$ is empty, then $\mathcal{A}$ must be empty, and the cardinality of an empty set is zero.

$$\therefore\ (\mathcal{E} = \varnothing) \implies (\mathcal{A} = \varnothing) \implies CAS(\mathcal{C}) = |\mathcal{A}| = |\varnothing| = 0 \qquad \square$$

Size 3. Module additivity.

Proof. Consider a model $\mathcal{C}$ with two disjoint modules $\ulcorner x \urcorner$ and $\ulcorner y \urcorner$ such that each element in $\mathcal{C}$ is either in $\ulcorner x \urcorner$ or in $\ulcorner y \urcorner$, but not both $((\mathcal{E}_x \cap \mathcal{E}_y = \varnothing)\ \wedge\ (\mathcal{E} = \mathcal{E}_x \cup \mathcal{E}_y))$. It follows, based on Equation (3.3.1), that x and y are the only two case elements in $\mathcal{L}$ ($\mathcal{L} = \{x, y\}$), thus $\mathcal{E}$ is partitioned by $\ulcorner x \urcorner$ and $\ulcorner y \urcorner$. But, in terms of Definition 3.10, each annotator 'a' in $\mathcal{A}$ is related to one and only one element x in $\mathcal{E}$. Therefore, if $\mathcal{E}$ is partitioned by $\ulcorner x \urcorner$ and $\ulcorner y \urcorner$ then $\mathcal{A}$ is also partitioned and $((\mathcal{A}_x \cap \mathcal{A}_y = \varnothing)\ \wedge\ (\mathcal{A} = \mathcal{A}_x \cup \mathcal{A}_y))$. Then,

$$\therefore\ CAS(\mathcal{C}) = |\mathcal{A}| = |\mathcal{A}_x| + |\mathcal{A}_y|\ =\ CAS(\ulcorner x \urcorner) + CAS(\ulcorner y \urcorner) \qquad \square$$

Validation of $\boldsymbol{CTS}(\mathcal{C})$

Again, this section will not repeat the definitions of the three properties of the $Size(S)$ function. It only uses their short description.

Size 1. Non-negativity.

Proof. In terms of Definition 7.3, $CTS(\mathcal{C})$ is the addition of two cardinalities ($\mathcal{E}$ and $\mathcal{A}$) which cannot be negative.

$$\therefore\ CTS(\mathcal{C}) = (|\mathcal{E}| + |\mathcal{A}|) \geq 0 \qquad \square$$

Size 2. Null value.

Proof. Using the *null value* proofs for $CS(\mathcal{C})$ and $CAS(\mathcal{C})$, we get

$$\therefore\ (\mathcal{E} = \varnothing) \implies (\mathcal{A} = \varnothing) \implies CTS(\mathcal{C}) = |\mathcal{E}| + |\mathcal{A}| = |\varnothing| + |\varnothing| = 0 \qquad \square$$

Size 3. Module additivity.

Proof. Using the *null value* proofs for $CS(\mathcal{C})$ and $CAS(\mathcal{C})$, we get

$\therefore\ CTS(\mathcal{C}) = |\mathcal{E}| + |\mathcal{A}| = (|\mathcal{E}_x| + |\mathcal{E}_y|) + (|\mathcal{A}_x| + |\mathcal{A}_y|) = (|\mathcal{E}_x| + |\mathcal{A}_x|) + (|\mathcal{E}_y| + |\mathcal{A}_y|) = CTS(\ulcorner x \urcorner) + CTS(\ulcorner y \urcorner)$

□

7.3.1.2 Length

Briand et al. define a function $Length(S)$ as being characterized by five properties that we use to validate the $CL(\mathcal{C})$ metric.

Length 1. Non-negativity. The length of a model $S = \langle E, R\rangle$ is non-negative.

$$(S = \langle E, R\rangle) \implies Length(S) \geq 0$$

Proof. CL is defined as the maximum nesting depth of a model and is calculated using algorithm $CL(\mathcal{C})$. Analyzing algorithm $CL(\mathcal{C})$, the variables (m and d) are initialized to zero, and only increased by one or assigned the maximum of itself and the depth which always returns $d + 1$.

$\therefore\ CL(\mathcal{C}) \geq 0$ □

Length 2. Null value. The length of a model $S = \langle E, R\rangle$ is zero if E is empty.

$$(S = \langle E, R\rangle \ \wedge\ E = \varnothing) \implies Length(S) = 0$$

Proof. Consider an empty model $\mathcal{C}$, then $\mathcal{E} = \varnothing \implies \mathcal{L} = \varnothing$ (because $\mathcal{L} \subseteq \mathcal{E}$). Analyzing algorithm $CL(\mathcal{C})$, it initializes m to zero, and if $\mathcal{L} = \varnothing$ then 'depth(m, c)' is never invoked, thus forcing the return of m which is zero.

$\therefore\ (\mathcal{E} = \varnothing) \implies CL(\mathcal{C}) = 0$ □

Length 3. Non-increasing monotonicity for connected components. Adding relationships between elements of a module m does not increase the length of the model $S = \langle E, R\rangle$.

$(S = \langle E, R\rangle \ \wedge\ m = \langle E_{m'}, R_m\rangle \ \wedge\ m \subseteq S \ \wedge\ m$ is a connected component of $S \ \wedge\ S' = \langle E, R'\rangle \ \wedge\ R' = R \cup \{\langle e_1, e_2\rangle\} \ \wedge\ \langle e_1, e_2\rangle \notin R \ \wedge\ e_1 \in E_{m'} \ \wedge\ e_2 \in E_{m'})$
$\implies Length(S) \geq Length(S')$

Proof. Consider a model $\mathcal{C}$ with two modeling elements in a module ($a, b \in \ulcorner x \urcorner$). There are two cases:

case 1 a and b are in different scopes. They cannot be related by scope, as this will require moving them within submodules of module $\ulcorner x \urcorner$ adding (a, b) to R which will change the structure of modules violating $\langle e_1, e_2 \rangle \notin R$. They cannot be related by an event, because in terms of Definition 3.15, in order to be related by an event they must necessarily be in the same scope.

case 2 a and b are in the same scope. They are already related by scope $[\![a, b]\!] \in \mathcal{U}$. Assuming $\langle a, b \rangle \notin \mathcal{V}$, we can relate them by an event and add them to $\mathcal{V}_m$, which corresponds to adding $\langle a, b \rangle$ to R' and leaving R invariant.

Therefore, a and b can be related only by an event, and the scope relationship $\mathcal{U}$ is not affected. Then, neither the length of $\ulcorner x \urcorner$ ($CL(\ulcorner x \urcorner)$), nor the length of the model $\mathcal{C}$ ($CL(\mathcal{C})$) has changed.

The maximum nesting depth of a forest is greater than or equal to the maximum nesting depth of any of the trees or subtrees, and $\ulcorner x \urcorner$ is either a tree or a subtree in a model $\mathcal{C}$.

$\therefore \; CL(\mathcal{C}) \geq CL(\ulcorner x \urcorner)$ □

Length 4. Non-decreasing monotonicity for non-connected components. Adding relationships between the elements of two modules m_1 and m_2 does not decrease the length of the model $S = \langle E, R \rangle$.

$(S = \langle E, R \rangle \;\wedge\; m_1 = \langle E_{m1}, R_{m1} \rangle \;\wedge\; m_2 = \langle E_{m2}, R_{m2} \rangle \;\wedge\; m_1 \subseteq S \;\wedge\; m_2 \subseteq S \;\wedge$ m_1, m_2 are separate connected components of $S \;\wedge\; S' = \langle E, R' \rangle \;\wedge\; R' = R \cup \{\langle e_1, e_2 \rangle\} \;\wedge$ $\langle e_1, e_2 \rangle \notin R \;\wedge\; e_1 \in E_{m1} \;\wedge\; e_2 \in E_{m2})$
$\implies \mathit{Length}(S) \geq \mathit{Length}(S')$

Proof. Adding relationships between the elements of two modules is not allowed in CMMN models. □

Length 5. Disjoint modules. The length of a model $S = \langle E, R \rangle$ made up of two disjoint modules m_1 and m_2 is equal to the maximum of the lengths of the modules m_1 and m_2.

$(S = m_1 \cup m_2 \;\wedge\; m_1 \cap m_2 = \varnothing \;\wedge\; E = E_{m1} \cup E_{m2})$
$\implies \mathit{Length}(S) = \mathit{max}(\mathit{Length}(m_1), \mathit{Length}(m_2))$

Proof. Consider a model $\mathcal{C}$ with two disjoint modules $\ulcorner x \urcorner$ and $\ulcorner y \urcorner$, such that $\mathcal{E} = \mathcal{E}_x \cup \mathcal{E}_y \ \wedge \ \mathcal{E}_x \cap \mathcal{E}_y = \varnothing$. Therefore, by Equation (3.3.1) x and y are the only two case elements in $\mathcal{L}$ ($\mathcal{L} = \{x, y\}$) and they create the only two trees ($\ulcorner x \urcorner, \ulcorner y \urcorner$) in the forest. The maximum nesting depth of a forest is equal to the maximum nesting depth of its trees.

$\therefore \ CL(\mathcal{C}) = max(CL(\ulcorner x \urcorner), CL(\ulcorner y \urcorner))$ □

7.3.1.3 Complexity

Briand et al. define a function *Complexity*(S) as being characterized by five properties that we use to validate the $CC(\mathcal{C})$ metric. Complexity for Briand et al. is distinct from cognitive complexity, since complexity in the framework is an intrinsic attribute of an object and not a perceived psychological complexity.

Complexity 1. Non-negativity. The complexity of a model $S = \langle E, R \rangle$ must be non-negative.

$$(S = \langle E, R \rangle) \implies Complexity(S) \geq 0$$

Proof. In terms of Definition 7.5, $CC(\mathcal{C}) = \sum_{i \in \mathcal{E} \cup \mathcal{A}} W_i$, where weight W_i is a positive integer from 0 to 3 as shown in Table 7.3. Suppose we replace each element in $\mathcal{E}$ with its weight and call the resulting set $\mathcal{E}^W$, and replace each annotator in $\mathcal{A}$ with its weight and call that set $\mathcal{A}^W$. Then, $CC(\mathcal{C}) = \sum_{i \in \mathcal{E} \cup \mathcal{A}} W_i = \sum_{p \in \mathcal{E}^W \cup \mathcal{A}^W} p \ \wedge \ p \in \mathbb{N}_0$, therefore $(\sum_{p \in \mathcal{E}^W \cup \mathcal{A}^W} p) \in \mathbb{N}_0$.

$\therefore \ CC(\mathcal{C}) \geq 0$ □

Complexity 2. Null value. The complexity of a model $S = \langle E, R \rangle$ is zero if R is empty.

$$(S = \langle E, R \rangle \ \wedge \ R = \varnothing) \implies Complexity(S) = 0$$

Proof. In terms of Definition 7.5, the complexity of an empty model is zero.

$\therefore \ (\mathcal{C} = \varnothing) \implies R = \mathcal{U} = \mathcal{V} = \varnothing \ \wedge \ CC(\mathcal{C}) = 0$ □

Complexity 3. Symmetry. The complexity of a model $S = \langle E, R \rangle$ does not depend on the convention chosen to represent the relationships between its elements.

$$(S = \langle E, R \rangle \ \wedge \ S^{-1} = \langle E, R^{-1} \rangle) \implies Complexity(S) = Complexity(S^{-1})$$

Proof. For a model $\mathcal{C}$, R is given by the two relationships $\mathcal{U}$ and $\mathcal{V}$. In terms of Definition 7.5, $CC(\mathcal{C}) = \sum_{i \in \mathcal{E} \cup \mathcal{A}} W_i$, which does not depend on $\mathcal{U}$ or $\mathcal{V}$, or the convention used to represent $\mathcal{U}$ and $\mathcal{V}$.

$\therefore \; CC(\mathcal{C}) = CC(\mathcal{C}')$ □

Complexity 4. Module monotonicity. The complexity of a model $S = \langle E, R \rangle$ is no less than the sum of the complexities of any two of its modules with no relationships in common.

$$(S = \langle E, R \rangle \;\wedge\; m_1 = \langle E_{m1}, R_{m1} \rangle \;\wedge\; m_2 = \langle E_{m2}, R_{m2} \rangle \;\wedge\; m_1 \cup m_2 \subseteq S \;\wedge\; R_{m1} \cap R_{m2} = \varnothing)$$
$$\implies \; Complexity(S) \geq Complexity(m_1) + Complexity(m_2)$$

Proof. Consider a model $\mathcal{C}$ with two modules $\ulcorner m1 \urcorner$ and $\ulcorner m2 \urcorner$ such that $\mathcal{E}_{m1} \cup \mathcal{E}_{m2} \subseteq \mathcal{E} \;\wedge\; \mathcal{U}_{m1} \cap \mathcal{U}_{m2} = \varnothing \;\wedge\; \mathcal{V}_{m1} \cap \mathcal{V}_{m2} = \varnothing$. We can ignore relationship $R = \mathcal{U} \cup \mathcal{V}$, because in terms of Definition 7.5, $CC(\mathcal{C})$ does not depend on $\mathcal{U}$ or $\mathcal{V}$.

Because $\mathcal{E}_{m1} \cup \mathcal{E}_{m2} \subseteq \mathcal{E}$, we can define T such that $\mathcal{E} = T \cup \mathcal{E}_{m1} \cup \mathcal{E}_{m2} \;\wedge\; T \cap \mathcal{E}_{m1} = T \cap \mathcal{E}_{m2} = T \cap \mathcal{E}_{m1} \cap \mathcal{E}_{m2} = \varnothing$.

Annotators in $\mathcal{A}$ are associated with elements in $\mathcal{E}$, therefore we can define a function f such that $f : \mathcal{A} \to \mathcal{E}$. We can separate the elements of $\mathcal{A}$ such that $\mathcal{A}_{m1} = \{x \mid f(x) \in \mathcal{E}_{m1}\}$, $\mathcal{A}_{m2} = \{x \mid f(x) \in \mathcal{E}_{m2}\}$, and $\mathcal{A}_T = \{x \mid f(x) \in T\}$.

By way of contradiction, assume $\mathcal{A}_{m1} \cap \mathcal{A}_{m2} \cap \mathcal{A}_T = \{a\}$. This means that there is an annotator 'a' in $\mathcal{A}$ for which $f(a) \notin T \cup \mathcal{E}_{m1} \cup \mathcal{E}_{m2}$, but $\mathcal{E} = T \cup \mathcal{E}_{m1} \cup \mathcal{E}_{m2}$. We have an annotator '$a$' without an image in $\mathcal{E}$ which contradicts our definition of f. Therefore, 'a' cannot exist, and $\mathcal{A} = \mathcal{A}_{m1} \cup \mathcal{A}_{m2} \cup \mathcal{A}_T \;\wedge\; \mathcal{A}_{m1} \cap \mathcal{A}_{m2} = \mathcal{A}_{m1} \cap \mathcal{A}_T = \mathcal{A}_{m2} \cap \mathcal{A}_T = \mathcal{A}_{m1} \cap \mathcal{A}_{m2} \cap \mathcal{A}_T = \varnothing$.

We have two cases for T:

case 1 $T = \varnothing$, in which case $\mathcal{E} = \mathcal{E}_{m1} \cup \mathcal{E}_{m2} \;\wedge\; \mathcal{E}_{m1} \cap \mathcal{E}_{m2} = \varnothing$. In terms of Equation (3.3.1) and complexity Complexity 5, we have $CC(\mathcal{C}) = CC(\ulcorner m1 \urcorner) + CC(\ulcorner m2 \urcorner)$

case 2 $T \neq \varnothing$, in which case $T \subset \mathcal{E} \;\wedge\; \mathcal{E}_{m1} \cap \mathcal{E}_{m2} = T \cap \mathcal{E}_{m1} = T \cap \mathcal{E}_{m2} = T \cap \mathcal{E}_{m1} \cap \mathcal{E}_{m2} = \varnothing$. Thus, $CC(\mathcal{C}) = \sum_{i \in \mathcal{E} \cup \mathcal{A}} W_i = \sum_{i \in \mathcal{E}_{m1} \cup \mathcal{E}_{m2} \cup T \cup \mathcal{A}_{m1} \cup \mathcal{A}_{m2} \cup \mathcal{A}_T} W_i = \sum_{i \in \mathcal{E}_{m1} \cup \mathcal{A}_{m1}} W_i + \sum_{i \in \mathcal{E}_{m2} \cup \mathcal{A}_{m2}} W_i + \sum_{i \in T \cup \mathcal{A}_T} W_i = CC(\ulcorner m1 \urcorner) + CC(\ulcorner m \urcorner) + CC(T)$

$\therefore \; CC(\mathcal{C}) \geq CC(\ulcorner m1 \urcorner) + CC(\ulcorner m2 \urcorner)$ □

Complexity 5. Disjoint module additivity. The complexity of a model $S = \langle E, R\rangle$ composed of two disjoint modules m_1 and m_2 is equal to the sum of the complexities of the two modules.

$$(S = \langle E, R\rangle \ \wedge \ S = m_1 \cup m_2 \ \wedge \ m_1 \cap m_2 = \varnothing)$$
$$\implies Complexity(S) = Complexity(m_1) + Complexity(m_2)$$

Proof. Consider a model $\mathcal{C}$ with two modules $\ulcorner m1 \urcorner$ and $\ulcorner m2 \urcorner$ such that $\mathcal{E} = \mathcal{E}_{m1} \cup \mathcal{E}_{m2} \ \wedge \ \mathcal{E}_{m1} \cap \mathcal{E}_{m2} = \varnothing$. In terms of Equation (3.3.1), $m1$ and $m2$ are case elements and $\mathcal{L} = \{m1, m2\}$.

Annotators in $\mathcal{A}$ are associated with elements in $\mathcal{E}$. Therefore, we can define a function f such that $f : \mathcal{A} \to \mathcal{E}$. We can separate the elements of $\mathcal{A}$ such that $\mathcal{A}_{m1} = \{x \mid f(x) \in \mathcal{E}_{m1}\}$ and $\mathcal{A}_{m2} = \{x \mid f(x) \in \mathcal{E}_{m2}\}$.

By way of contradiction, assume $\mathcal{A}_{m1} \cap \mathcal{A}_{m2} = \{a\}$. That means that there is an annotator 'a' in $\mathcal{A}$ for which $f(a) \notin \mathcal{E}_{m1} \cup \mathcal{E}_{m2}$, but $\mathcal{E} = \mathcal{E}_{m1} \cup \mathcal{E}_{m2}$. We have an annotator '$a$' without an image in $\mathcal{E}$ which contradicts our definition of f. Therefore, 'a' cannot exist, and $\mathcal{A} = \mathcal{A}_{m1} \cup \mathcal{A}_{m2} \ \wedge \ \mathcal{A}_{m1} \cap \mathcal{A}_{m2} = \varnothing$.

Now, we have

$$\begin{aligned} CC(\mathcal{C}) &= \sum_{i \in \mathcal{E} \cup \mathcal{A}} W_i = \sum_{i \in \mathcal{E}_{m1} \cup \mathcal{E}_{m2} \cup \mathcal{A}_{m1} \cup \mathcal{A}_{m2}} W_i = \sum_{i \in \mathcal{E}_{m1} \cup \mathcal{A}_{m1}} W_i + \sum_{i \in \mathcal{E}_{m2} \cup \mathcal{A}_{m2}} W_i \\ &= CC(\ulcorner m1 \urcorner) + CC(\ulcorner m2 \urcorner) \end{aligned}$$

$\therefore \ CC(\mathcal{C}) = CC(\ulcorner m1 \urcorner) + CC(\ulcorner m2 \urcorner)$ □

7.3.2 Weyuker's Properties

Weyuker [Wey88] proposed a set of nine properties for complexity software metrics that have been widely used to validate business process metrics [Muk+10a]. Briand et al. [Bri+96] found that Weyuker's [Wey88] properties were consistent with complexity in his framework.

The Weyuker [Wey88] notation uses P, Q, and R to denote programs, which for our purposes corresponds with CMMN models. To avoid confusion with the Briand et al. [Bri+96] framework, this section will use T instead of R. Weyuker uses the set cardinality operator $|P|$ to denote complexity, which we have already used to denote cardinality. Therefore, we will use $\|P\|$ as the complexity of model P. Weyuker anticipates that for any P, its complexity $\|P\|$ will be a non-negative number, hence complexity can be compared and ordered.

$$\|P\| \leq \|Q\| \ \vee \ \|Q\| \leq \|P\|$$

We will use the terminology used by Weyuker [Wey88] and the classifications used by Srinivasan and Devi [SD14] to introduce each property, followed by a short proof against the complexity metric $CC(\mathcal{C})$.

Property 1. Non-coarseness. A metric should not rank all models as equally complex.

$$(\exists P)(\exists Q)(\|P\| \neq \|Q\|)$$

Proof. Let P be example 2 (see Figure 3.5) with $CC(P) = 7$, and Q be example 3 (see Figure 3.6) with $CC(Q) = 11$.

$$\therefore \ (\exists P)(\exists Q)(CC(P) \neq CC(Q))$$ □

Property 2. Granularity. A metric should rank only a finite number of models with the same complexity.

Let c be a non-negative number, then there are only finitely many models of complexity c.

Proof. Assuming a model can only be renamed (see Property 8) in a finite number of ways, then consider a number $c \in \mathbb{N}_0$, and a model P such that $CC(P) = c$. In terms of Definition 7.5, $CC(P)$ is calculated using $\mathcal{E}_P$ and $\mathcal{A}_P$. We need to show that there is a finite number of $\mathcal{E}_P \cup \mathcal{A}_P$ sets such that $CC(P) = c$. Note that if $c = 0$ then $\mathcal{E}_P \cup \mathcal{A}_P = \varnothing \ \wedge \ CC(\varnothing)$, thus there is only one $\mathcal{E}_P \cup \mathcal{A}_P$ that gives $c = 0$ □

We start by proving that $\mathcal{E}_P \cup \mathcal{A}_P$ is finite for $c > 0$. The only element in $\mathcal{E}_P$ (see Table 3.3) that has a weight of 0 is the connector (see Table 7.3), but a connector must be associated with a sentry in $\mathcal{A}_P$ with a weight of 1 (entry or exit criteria). We cannot add an infinite number of connectors to $\mathcal{E}_P$ without adding sentries and changing the value of $CC(P)$. Thus, $CC(P) = c \implies |\mathcal{E}_P| \neq \infty$, and model P has a size $CS(P) = |\mathcal{E}_P| = n$. Set $\mathcal{E}_P$ is finite $(\exists n)(|\mathcal{E}_P| \leq n)$. In terms of Definition 3.10, each element in $\mathcal{A}_P$ is associated with a single element in $\mathcal{E}_P$. In $\mathcal{A}_P$ (see Table 3.4) markers and decorators are bound to $|\mathcal{E}_P|$, but sentries are not. However, sentries' (entry and exit criteria) weights range from 1 to 3. We cannot have an infinite number of sentries, because the number is bound by c (i.e., $\sum_{i \in \{x|x \in \{1,2,3\} \times |\mathcal{A}_P|\}} i \leq c$). Therefore, $\mathcal{A}_P$ is also finite and $(\exists m)(|\mathcal{A}_P| \leq m)$. Moreover $\mathcal{E}_P \cap \mathcal{A}_P = \varnothing \implies |\mathcal{E}_P| + |\mathcal{A}_P| = |\mathcal{E}_P \cup \mathcal{A}_P| \leq n + m$. Therefore, $\mathcal{E}_P \cup \mathcal{A}_P$ is a finite set.

Without loss of generality, we can summarize Table 7.3 into four weight categories $(0, 1, 2, 3)$, and we know $|\mathcal{E}_P \cup \mathcal{A}_P| \leq n + m$. Using brute force, we can count all combinations with a repetition of four categories $(0, 1, 2, 3)$ into 1 to $n + m$ slots, using $C(n + r - 1, r)$. This will give us all possible sets $\mathcal{E}_P \cup \mathcal{A}_P$ that can produce, among other complexities, $CC(P) = c$.

We calculate $\sum_{r=1}^{n+m} C(4+r-1, r) = 2^{4+n+m-1}$. Therefore, there is a finite number of sets $\mathcal{E}_P \cup \mathcal{A}_P$ such that $CC(P) = \sum_{i \in \mathcal{E}_P \cup \mathcal{A}_P} W_i = c$ □

Property 3. Non-uniqueness (Notion of equivalence). A metric should allow some models to have the same complexity.

$$(\exists P)(\exists Q)(P \neq Q \ \wedge \ \|P\| = \|Q\|)$$

Proof. Let P be example 2 (see Figure 3.5) with $CC(P) = 7$, and Q a similar model, but changing modeling element o (case file item) to a task t. Thus, Q is different from P by one modeling element. The complexity of Q is $CC(Q) = 7$, because the weight for a task is the same as the weight for a case file item (see Table 7.3).

$$\therefore \ (\exists P)(\exists Q)(P \neq Q \ \wedge \ CC(P) = CC(Q))$$ □

Property 4. Design details are important. Two distinct but equivalent models that compute the same function need not have the same complexity.

$$(\exists P)(\exists Q)(P \equiv Q \ \wedge \ \|P\| \neq \|Q\|).$$

Proof. Let P be example 3 (see Figure 3.6) with $CC(P) = 11$, and Q a similar model, but includes an extra task t inside case x. Task t is a dummy task that does nothing when it executes (a 'skip' statement). Operationally, Q is equivalent to P ($Q \equiv P$), because they compute the same function. However, the complexity of Q is $CC(Q) = 12$ because task t adds a weight of 1 (see Table 7.3).

$$\therefore \ (\exists P)(\exists Q)(Q \equiv P \ \wedge \ CC(P) \neq CC(Q))$$ □

Property 5. Monotonicity. The complexity of two models joined together is greater than or equal to the complexity of either model considered separately.

$$(\forall P)(\forall Q)(\|P\| \leq \|Q;P\| \ \wedge \ \|Q\| \leq \|P;Q\|).$$

Proof. Let $CC(P) = \sum_{i \in \mathcal{E}_P \cup \mathcal{A}_P} W_i$, and $CC(Q) = \sum_{i \in \mathcal{E}_Q \cup \mathcal{A}_Q} W_i$.

Then, $CC(P;Q) = \sum_{i \in \mathcal{E}_P \cup \mathcal{E}_Q \cup \mathcal{A}_P \cup \mathcal{A}_Q} W_i = \sum_{i \in \mathcal{E}_Q \cup \mathcal{E}_P \cup \mathcal{A}_Q \cup \mathcal{A}_P} W_i = CC(Q;P)$.

We have two cases:

case 1 $P = \varnothing \lor Q = \varnothing$. Assume P is an empty model, then $\mathcal{E}_P = \mathcal{A}_P = \varnothing$. Therefore, $CC(P;Q) =$
$\sum_{i \in \varnothing \cup \mathcal{E}_Q \cup \varnothing \cup \mathcal{A}_Q} W_i = \sum_{i \in \mathcal{E}_Q \cup \mathcal{A}_Q} W_i = CC(Q)$. Assuming Q is an empty model will give the same result $CC(P;Q) = CC(P)$. Assuming both P and Q are empty leads to the same result $CC(P;Q) = CC(P) = CC(Q) = 0$.

case 2 $P \neq \varnothing \land Q \neq \varnothing$. Then, $CC(P;Q) = \sum_{i \in \mathcal{E}_P \cup \mathcal{E}_Q \cup \mathcal{A}_P \cup \mathcal{A}_Q} W_i = \sum_{i \in \mathcal{E}_P \cup \mathcal{A}_P} W_i + \sum_{i \in \mathcal{E}_Q \cup \mathcal{A}_Q} W_i = CC(P) + CC(Q)$.

$\therefore \ (\forall P)(\forall Q)(CC(P) \leq CC(P) + CC(Q) \land CC(Q) \leq CC(P) + CC(Q))$ □

Property 6. Nonequivalence of interaction. Given two models with the same complexity, when each is joined to a third model the resulting complexity may be different between the two.

a:$(\exists P)(\exists Q)(\exists T)(\|P\| = \|Q\| \land \|P;T\| \neq \|Q;T\|)$

b:$(\exists P)(\exists Q)(\exists T)(\|P\| = \|Q\| \land \|T;P\| \neq \|T;Q\|)$

Proof. For complexity $CC$the order of concatenation of models is not important. Let $CC(P) = \sum_{i \in \mathcal{E}_P \cup \mathcal{A}_P} W_i$, and $CC(T) = \sum_{i \in \mathcal{E}_T \cup \mathcal{A}_T} W_i$. Then, $CC(P;T) = \sum_{i \in \mathcal{E}_P \cup \mathcal{E}_T \cup \mathcal{A}_P \cup \mathcal{A}_T} W_i = \sum_{i \in \mathcal{E}_T \cup \mathcal{E}_P \cup \mathcal{A}_T \cup \mathcal{A}_P} W_i = CC(T;P)$ Therefore, $CC(P;T) = CC(T;P)$ which is the same as $\|P;T\| = \|T;P\|$, and we have only one case.

Let P be example 2 (see Figure 3.5) with $CC(P) = 7$, and Q similar to P but changing modeling element o to a task t. Thus, Q is different from P by one modeling element, still $CC(P) = CC(Q) = 7$ (which is the same as in property Property 3). Now let the case annotator 'l' in P invoke case z, and the case annotator 'l' in Q invoke case x. This does not change $CC(P)$ or $CC(Q)$, because neither case z or case x are in the models, therefore the weight of 'l' remains 0 (see Table 7.3).

Let T be example 1 (see Figure 3.4) with $CC(T) = 1$. Concatenating P with T produces $CC(P;T) = 9$ (weights z=1, w=1, o=1, p=0, q=1, k=1, l=1, j=3), where 'l'=1, because case z is in $P;T$. However, concatenating Q with T is $CC(Q;T) = 8$ (weights z=1, w=1, o=1, p=0, q=1,k=1, l=0, j=3), where 'l'=0, because case x is not in $Q;T$.

$\therefore \ (\exists P)(\exists Q)(\exists T)(CC(P) = CC(Q) \land CC(P;T) \neq CC(Q;T))$ □

Property 7. Permutation. Complexity should be responsive to the order of statements.

$(\exists P)(\exists Q)(Permutation(Q, P) \land \|P\| \neq \|Q\|)$

Proof. Let P be example 2 (see Figure 3.5) with $CC(P) = 7$, and Q be similar to P but with connector p attached to exit criterion 'j' instead of entry criterion 'k'. Thus, Q is a permutation of P with $CC(Q) = 6$ because in $CC(Q)$ the weight of 'k' has increased from 1 to 2, and the weight of 'j' has decreased from 3 to 1 (see Table 7.3).

$\therefore\ (\exists P)(\exists Q)(Permutation(Q, P) \wedge CC(P) \neq CC(Q))$ □

Property 8. Renaming. Complexity should not be affected by renaming.

$(\forall P)(\forall Q)(Rename(Q, P)\ \wedge\ \|P\| = \|Q\|)$

Proof. The names of elements and annotators do not affect CC. Let P be example 2 (see Figure 3.5) with $CC(P) = 7$, and Q be similar to P but with different names. Suppose we rename w, o, p, q, k, l, j in $\mathcal{E}_P$ to a, b, c, d, e, f, q in $\mathcal{E}_Q$. Thus, Q is renamed P. Note that Q has the same number and type of modeling elements and annotators as P, and that they are organized in exactly the same way. Thus, the complexity of Q remains $CC(Q) = 7$. This can be done with any model.

$\therefore\ (\forall P)(\forall Q)(Rename(Q, P)\ \wedge\ CC(P) = CC(Q))$ □

Property 9. Interaction may increase complexity.

$(\exists P)(\exists Q)(\|P\| + \|Q\| < \|P;Q\|)$.

Proof. Let P be example 1 (see Figure 3.4) with $CC(P) = 1$, and Q be example 2 (see Figure 3.5) with $CC(Q) = 7$. Assume that case annotator 'l' in Q invokes case z, then in Q the weight of case annotator 'l' is 0, but when joined with P that has case z, the weight of annotator 'l' becomes 1, and $CC(P;Q) = 9$ (see Table 7.3).

$\therefore\ (\exists P)(\exists Q)(CC(P) + CC(Q) < CC(P;Q))$ □

7.4 Summary

This chapter provided a formal description of a set of CMMN process modeling complexity metrics. The identified metrics were theoretically validated by using the suggestions made by Briand et al. [Bri+95] and Misra et al. [Mis+12]. Each metric was validated through the application of measurement theory as suggested by Briand et al. and by using Weyuker's [Wey88] properties, as suggested by Misra et al. As evidenced in this chapter, it is clear that Briand et al. and Weyuker assumed that software systems are built using a procedural style, based on directed acyclic graphs. Briand et al. used directed acyclic graphs to describe their framework and to provide examples, and Weyuker used a procedural language to illustrate her properties. Despite using a procedural style both approaches were useful to theoretically validating the proposed CMMN complexity metrics.

The complexity metrics for CMMN defined in this chapter will be empirically validated in Chapter 8. Material from this chapter was previously published in Marin et al. [Mar+15b].

Chapter 8

Empirical Validation of Case Management Metrics

Chapter 7 defined a set of metrics for Case Management Model and Notation (CMMN) [OMG14a]. These metrics included: the size *CS*, the annotators' size *CAS*, the total size *CTS*, the length *CL*, and the complexity *CC*. In addition, a set of sub-metrics were defined, similar to those used by Rolón et al. [Rol+06b], in which all of the components of a model are counted and used as a metric. The metrics were theoretically validated in Section 7.3 using the formal framework for software measurements defined by Briand et al. [Bri+96], and the complexity metric *CC* was further validated using the nine properties for complexity measures as defined by Weyuker [Wey88]. Briand et al.'s [Bri+96] framework and Weyuker's [Wey88] properties are categorized as axiomatic or property-based approaches to validation [SD14], and are commonly used for validating software metrics [Muk+10a].

This chapter attempts to use a novel approach to validate complexity metrics based on pairwise comparisons. The Systematic Literature Review (SLR) of the last 20 years, presented in Chapter 6, did not uncover any studies that used a pairwise comparison to validate complexity metrics. In addition, this study departs from most validations of complexity metrics for process models in several important ways:

1. It uses power calculations during the experimental design phase to estimate the minimum acceptable sample size. As described in Section 6.4.1.3, the SLR did not uncover any papers that justified the sample sizes.

2. As a result of the power calculations a larger sample size will be required for this study. The maximum sample size for human validations uncovered by the SLR was 86 subjects as evidenced in the work done by García et al. [Gar+04a] (see Section 6.4.1.3).

3. This study targets professional process modelers instead of students. In the SLR, 74% of the papers used students as the subjects, and another 2% used professors and students. The rest of the papers reported using experts, academics, practitioners, professionals, and students.

4. This study uses a within-subjects pairwise comparison experimental design. While within-subjects experiments have been used in some of the empirical validations that made use of human subjects [Gar+04b; Men+07c; Van+08a], none have attempted a pairwise comparison.

This chapter describes the empirical validation of the metrics, and is organized as follows. Section 8.2 describes the methodology used for the empirical validation, including the experimental design in Section 8.2.3. Section 8.3 presents the results and the analysis of the statistical test.

One appendix and several supplementary documents complement the content of this chapter. Appendix C contains the approval letter from the College of Science, Engineering and Technology's (CSET) Research and Ethics Committee. Appendix D describes a set of files containing supplementary material that includes: file 27 (The6Models.pdf) which describes six models used in this experiment and the process used to generate them; file 15 (2016-05-23 MAMarin_Student_Ethical_Clearance-v5.pdf) which contains the latest ethical clearance form that was submitted to CSET's Research and Ethics Committee; file 26 (Cherries.pdf) which contains a filled in Checklist for Reporting Results of Internet E-Surveys (CHERRIES) [Eys12] for the survey and tutorial; file 19 (2016-06-15 Survey-Example.pdf) which contains a sample survey; file 20 (2016-06-15 Survey-Tutorial.pdf) which presents the tutorial that was included in the survey; file 1 (dataset-all(description).pdf) which describes all of the variables that were generated from the survey; file 25 (Basic-stats.pdf) which contains other survey statistics that complement Section 8.3.

8.1 Empirical Validation

The validation of complexity metrics for business processes requires both theoretical and empirical validation [Mis+12; Muk+10a; SD14]. Accordingly to Muketha et al. [Muk+10a] the main goal of empirical validation is to establish whether the new metric measures what it is intended to measure. The literature offers two main strategies to empirically validate metrics:

Human validation. This validation is done using case studies, experiments, or surveys [Mis+12; Muk+10a; SD14]. Several authors have followed this approach, including

Cardoso [Car06b] and Rolón et al. [Rol+08]. These experiments have used students as subjects for the empirical validation.

Comparing against other metrics. Preliminary empirical validation can be done by comparing the proposed metrics against other well-known metrics [Mis+12]. However, Fenton and Pfleeger [FP98] warn against this type of validation unless the metrics being used for the correlation have been thoroughly validated for exactly the same attributes as the new metrics. Several authors have used comparison as the only empirical validation for their metrics, including Lassen and van der Aalst [Lv09], Mendling et al. [Men+07a], and Muketha et al. [Muk+10b].

This thesis followed Fenton and Pfleeger's [FP98] suggestion and pursued the human validation approach because there are no known CMMN metrics against which a comparison can be conducted. The empirical validation used the CMMN complexity metrics identified in Section 7.2 to calculate the complexity of a set of models, and to explore the relationship between calculated complexity, model comprehension and perceived complexity. The independent variables, or treatment, were the calculated complexity and these were operationalized by using a set of six CMMN models. Each of the identified complexity metrics was calculated for each of the CMMN models. The dependent variables included the model comprehension, perceived complexity, pairwise comparison, and a set of weights.

8.2 Methodology

The objective of the empirical validation was to compare the calculated complexity metrics identified in Section 7.2 against human perceived complexity. This thesis adopted a quantitative approach using an online survey. The research question was: *does calculated complexity correlate with human perceived complexity?* However, in order for a person to evaluate the perceived complexity of a process model, he or she must spend time understanding the model. For this reason, model comprehension questions were used to force the subjects to understand the process model. A secondary objective was to fine tune the weights used to calculate complexity CC (see Table 7.3).

8.2.1 Hypotheses

This thesis hypothesized a positive correlation between calculated complexity and perceived complexity, a negative correlation between calculated complexity and model comprehension, and a negative correlation between model comprehension and perceived complexity as presented in Figure 8.1. Calculated complexity was expected to correlate negatively with model

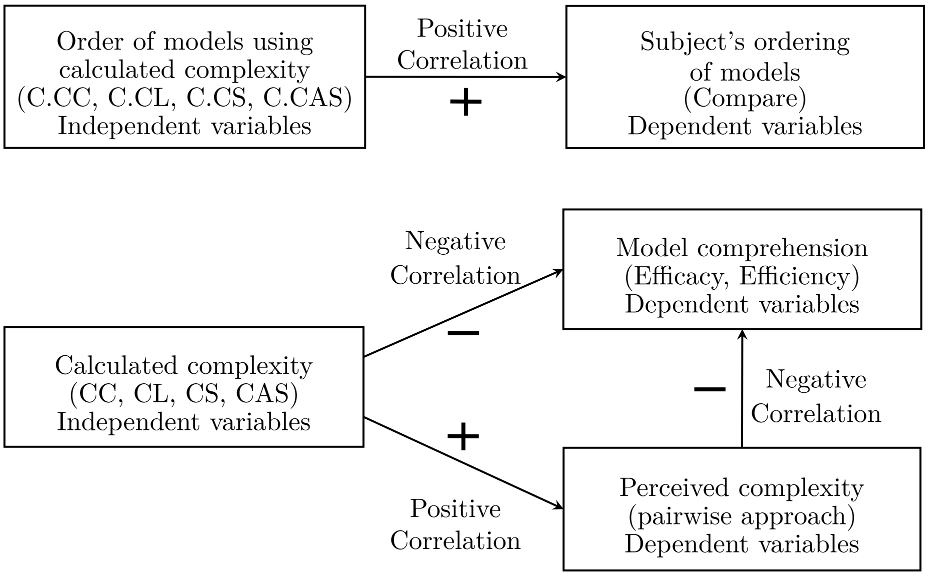


Figure 8.1: Hypothesized relationships between variables.

comprehension because as the model becomes more complex the user comprehension of the model should decrease. The calculated comprehension was expected to correlate positively with perceived complexity because calculated complexity should be consistent with user perceived complexity. Finally, perceived complexity was expected to correlate negatively with model comprehension, because as the model becomes more complex the user comprehension of the model should decrease. These hypotheses were in part based on the findings of other researchers projects regarding imperative process models as uncovered by the SLR, like the positive correlation between calculated complexity and human perceived complexity found by Cardoso [Car05d; Car06b; Car08], and Mendling et al. [Men+07c] and Mendling and Strembeck's [MS08] findings on the correlation between model understanding and complexity metrics.

The following five hypotheses were formulated:

$H1_0$: There is no significant relationship between the *calculated complexity* of the process models presented, as measured by the complexity metrics created for this study (*CC*, *CL*, *CS*, and *CAS*), and the subject's *model comprehension* of the models.

$H1_a$: There is a significant relationship between the *calculated complexity* of the process models presented, as measured by the complexity metrics created for this study (*CC*, *CL*, *CS*, and *CAS*), and the subject's *model comprehension* of the models.

$H2_0$: There is no significant relationship between the *calculated complexity* of the process models, as measured by the complexity metrics created for this study (*CC*, *CL*, *CS*, and *CAS*), and the subject's *perceived complexity* of the models.

$H2_a$: There is a significant relationship between the *calculated complexity* of the process models, as measured by the complexity metrics created for this study (CC, CL, CS, and CAS), and the subject's *perceived complexity* of the models.

$H3_0$: There is no significant relationship between the subject's *perceived complexity* of the process models, and the subject's *model comprehension* of the same process models.

$H3_a$: There is a significant relationship between the subject's *perceived complexity* of the process models, and the subject's *model comprehension* of the same process models.

$H4_0$: There is no significant relationship between the *pairwise ordering* of the process models using *calculated complexity*, as measured by the complexity metrics created for this study (CC, CL, CS, and CAS), and the subject's *pairwise ordering* of the models.

$H4_a$: There is a significant relationship between the *pairwise ordering* of the process models using *calculated complexity*, as measured by the complexity metrics created for this study (CC, CL, CS, and CAS), and the subject's *pairwise ordering* of the models.

$H5_0$: The *weights* of the CMMN symbols used to calculate CC (see Table 7.3) are the same as the subject's *perceived symbol complexity*.

$H5_a$: The *weights* of the CMMN symbols used to calculate CC (see Table 7.3) are different from the subject's *perceived symbol complexity*.

8.2.2 Operational Definition of Variables

Calculated complexity was operationalized using a set of six process models, each one using a different calculated complexity (see Appendix D file 27 (The6Models.pdf)). The *complexity metrics* created for this study (CC, CL, CS, and CAS) were calculated to create independent variables for each of the six process models. The dependent variables were based on the subjects' answers to the survey. The survey was designed to collect the required dependent variables to test each hypothesis

Table 8.1 provides an overview of the main variables. Each subject taking part in the survey was exposed to two treatments in the form of two models (A and B). Therefore, the names of most of the variables are tagged with an A or a B, which indicates the treatment. Independent variables start with '`iv.`', and so, '`iv.A.CC`' corresponds to the independent variable calculated CC for treatment A. A detailed description of the variables in the dataset is listed in Appendix D file 1 (dataset-all(description).pdf).

Table 8.1: Operational definition of variables

Type	Scale	Name	Source	Values	Hypotheses

Independent	Ordinal	`iv.C.CC`
		`iv.C.CL`
		`iv.C.CS`
		`iv.C.CAS`
			Calculated based on the order of the calculated metric	Likert scale, 1 to 9	H4

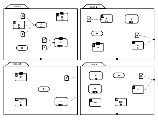
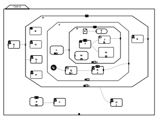
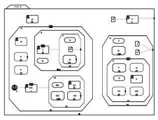
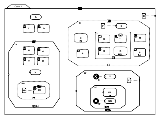
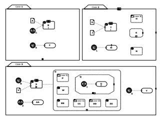
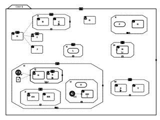
	Ratio	`iv.A.CC`
		`iv.B.CC`
		`iv.A.CL`
		`iv.B.CL`
		`iv.A.CS`
		`iv.B.CS`
		`iv.A.CAS`
		`iv.B.CAS`
			Calculated for each of the six models	Integer, 0 to 200	H1, H2

	Ordinal	iv.W.*	Calculated using Table 7.3	Likert scale, 1 to 8	H5
Dependent	Ordinal	`C.Compare`	Survey question	Likert scale, 1 to 9	H4
	Ratio	`A.Correct` `B.Correct`	Survey, count of correct answers	count, 0 to 5	H1, H3
		`A.Time` `B.Time`	Survey, time used to answer the questions	duration in seconds	
		`A.Efficacy` `B.Efficacy`	Calculated, correct answers divided by number of questions	float, 0 to 1	
		`A.Efficiency` `B.Efficiency`	Calculated, correct answers divided by time		
	Ordinal	`A.perceived` `B.perceived`	Survey question	Likert scale, 1 to 7	H2, H3
		Weights.*	Survey question	Likert scale, 1 to 8	H5

8.2.2.1 Independent Variables

Table 8.2 presents the values of the independent variables (*CC*, *CL*, *CS*, *CAS*) for each of the models. The calculations and the models are described in Appendix D file 27 (The6Models.pdf).

Table 8.2: Values of the independent variables *CC*, *CL*, *CS*, *CAS*

Model	model 1 	model 2 	model 3 	model 4 	model 5 	model 6 
CC	72	82	87	96	102	114
CL	2	5	5	4	4	4
CS	44	36	42	40	45	37
CAS	46	54	48	50	45	53
CTS	90	90	90	90	90	90

Note that:

CC(Model 1) $<$ CC(Model 2) $<$ CC(Model 3) $<$ CC(Model 4) $<$ CC(Model 5) $<$ CC(Model 6).

CL(Model 1) $<$ CL(Model 4) $=$ CL(Model 5) $=$ CL(Model 6) $<$ CL(Model 2) $=$ CL(Model 3).

CS(Model 2) $<$ CL(Model 6) $<$ CL(Model 4) $<$ CL(Model 3) $<$ CL(Model 1) $<$ CL(Model 5).

CAS(Model 5) $<$ CAS(Model 1) $<$ CAS(Model 3) $<$ CAS(Model 4) $<$ CAS(Model 6) $<$ CAS(Model 2).

$CTS = 90$ for all of the models.

CTS is the number of icons in the model, which was kept constant. Therefore, it was not used as a variable in the study.

8.2.3 Experimental Design

The research methodology followed a within-subjects experimental design. The subjects consisted of a convenient sample of professional process modelers sourced through a set of online forums and mailing lists.

The experiment was implemented using an online survey and tutorial. A short tutorial was included, because the target population might not have been familiar with CMMN. The experiment used a repeated measures design with counterbalancing where each subject was exposed to only two models. The survey included a short description of the experiment, a letter of informed consent, a demographics and previous experience questionnaire, a short 30 minute tutorial of CMMN, the two CMMN models, and finally a form exploring the CMMN notation complexity.

The survey questions were designed to examine the relationship between the independent variable, the calculated complexity (*CC*, *CL*, *CS*, and *CAS*), and the dependent variables, subject's pairwise comparison, model comprehension, and perceived complexity. The independent variables constituted the calculated complexity (*CC*, *CL*, *CS*, and *CAS*) for each of the six CMMN models. The dependent variables were calculated based on the subject's responses to a pairwise comparison question, five comprehension questions, and one perceived complexity question per model, and a question on CMMN symbols.

The treatment consisted of six CMMN models (see Appendix D file 27 (The6Models.pdf)). Each model had five questions (Activity period, Concurrency, Exclusiveness, Order, Repetition, and Notation), which corresponded to the suggestions made by Laue and Gadatsch [LG11] and Melcher et al. [Mel+10; Men+07c]. The six CMMN models were used in pairs to produce the equivalent of 30 surveys (resulting from the permutations with repetitions $P_2^6 = \frac{6!}{(6-2)!} = 30$). The assignment of subjects to a treatment was based on the order of arrival of the subjects on the online survey page. The surveys were assigned to subjects using a round-robin algorithm based on the order of arrival.

The following five sections describe the research design used to explore each of the five hypotheses.

8.2.3.1 Model Comprehension

This section describes the experimental design used to explore the relationship between the *calculated complexity* of the process models presented, as measured by the complexity metrics created for this study (*CC*, *CL*, *CS*, and *CAS*), and the subject's *model comprehension* of the models (see hypotheses $H1_0$ and $H1_a$).

Each survey had two models (A and B), and each model had a set of five model comprehension questions. Model comprehension was included in the survey to force the subjects to familiarize themselves with the two models in order to provide feedback on perceived model complexity. Model comprehension was expected to correlate negatively with calculated complexity, because as the model becomes more complex the user comprehension of the model should decrease.

Goal: To force the subject to understand the two models (A and B) and to form an opinion of their complexity.

Independent variables: Calculated complexity (`iv.A.CC`, `iv.A.CL`, `iv.A.CS`, `iv.A.CAS`, `iv.B.CC`, `iv.B.CL`, `iv.B.CS`, and `iv.B.CAS`).

Dependent variables: Number of correct answers (`A.Correct` and `B.Correct`), time used to answer (`A.Time` and `B.Time`), Efficacy (`A.Efficacy` and `B.Efficacy`), and Efficiency (`A.Efficiency` and `B.Efficiency`).

Significance level: The statistical tests were evaluated at a 5% level of significance ($\alpha = 0.05$).

Hypothesis: The hypotheses $H1_0$ and $H1_a$ were adapted to the operational variables and to models A and B as follows:

$H1a_0$**:** There is no significant relationship between the *calculated complexity* (`iv.A.CC`, `iv.A.CL`, `iv.A.CS`, and `iv.A.CAS`) and the subject's *model comprehension* (`A.Correct`, `A.Time`, `A.Efficacy`, and `A.Efficiency`) of the models.

$H1a_0 : \rho = 0$

$H1a_a$**:** There is a significant relationship between the *calculated complexity* (`iv.A.CC`, `iv.A.CL`, `iv.A.CS`, and `iv.A.CAS`) and the subject's *model comprehension* (`A.Correct`, `A.Time`, `A.Efficacy`, and `A.Efficiency`) of the models.

$H1a_a : \rho \neq 0$

$H1b_0$**:** There is no significant relationship between the *calculated complexity* (`iv.B.CC`, `iv.B.CL`, `iv.B.CS`, and `iv.B.CAS`) and the subject's *model comprehension* (`B.Correct`, `B.Time`, `B.Efficacy`, and `B.Efficiency`) of the models.

$H1b_0 : \rho = 0$

$H1b_a$**:** There is a significant relationship between the *calculated complexity* (`iv.B.CC`, `iv.B.CL`, `iv.B.CS`, and `iv.B.CAS`) and the subject's *model comprehension* (`B.Correct`, `B.Time`, `B.Efficacy`, and `B.Efficiency`) of the models.

$H1b_a : \rho \neq 0$

Data analysis: Pearson's or Spearman's correlations were planned depending on data normality. Spearman's correlation was used because data was not confirmed as normal. Two sets of correlations were run, one set for observation A and one set for observation B.

Power analysis: The following analysis was performed using G*Power [Fau+07]:

Table 8.3: A priori power calculation for two-tailed correlation

t tests	Correlation: Point biserial model		
Analysis:	A priori: Compute required sample size		
		large effect	medium effect
Input:	Tail(s)	= Two	= Two
	Effect size $\mid \rho \mid$	= **0.50**	= **0.30**
	α err prob	= 0.05	= 0.05
	Power ($1 - \beta$ err prob)	= 0.80	= 0.80
Output:	Noncentrality parameter δ	= 2.9439203	= 2.8477869
	Critical t	= 2.0638986	= 1.9900634
	Df	= 24	= 80
	Total sample size	= **26**	= **82**
	Actual power	= 0.8063175	= 0.8033045

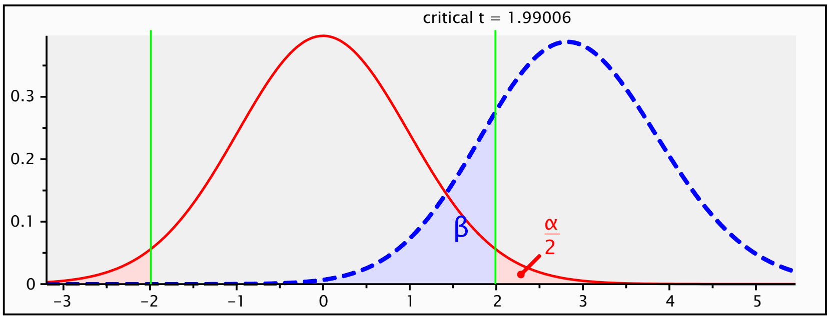


Figure 8.2: A priori power calculation for two-tailed correlation

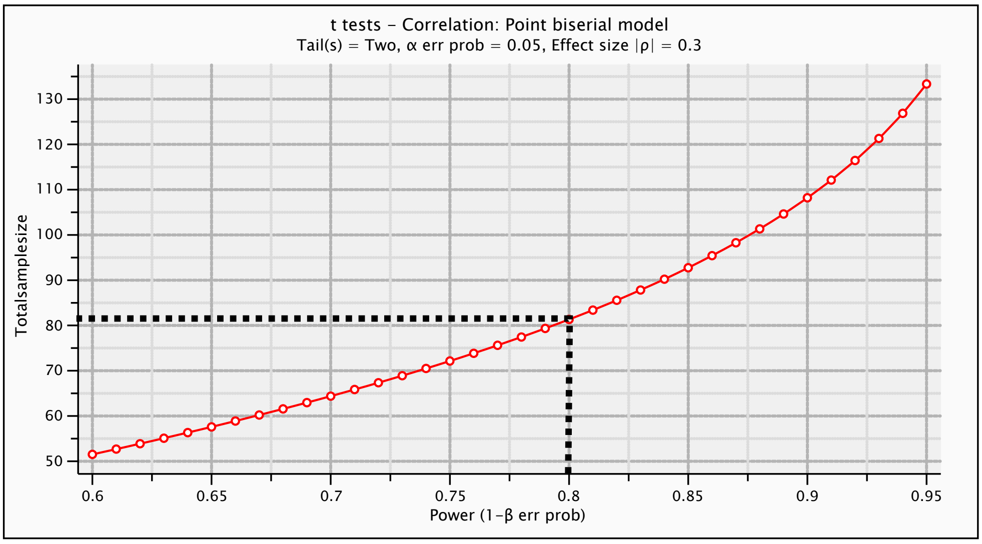


Figure 8.3: Plot of a priori power calculation for two-tailed correlation

8.2.3.2 Perceived Complexity

This section describes the experimental design used to explore the relationship between the *calculated complexity* of the process models, as measured by the complexity metrics created for this study (*CC*, *CL*, *CS*, and *CAS*), and the subject's *perceived complexity* of the models (see Hypotheses $H2_0$ and $H2_a$).

Each model (A and B) in the survey had a single perceived complexity question in addition to the model comprehension questions. It was expected that a subject would form an opinion about the perceived complexity after it had been exposed to the comprehension questions. It was also expected that perceived complexity would positively correlate with calculated complexity.

Goal: Determine if calculated complexity (*CC*, *CL*, *CS*, *CAS*) matches human perception of complexity.

Independent variables: Calculated complexity (`iv.A.CC`, `iv.A.CL`, `iv.A.CS`, `iv.A.CAS`, `iv.B.CC`, `iv.B.CL`, `iv.B.CS`, and `iv.B.CAS`).

Dependent variables: Perceived complexity (`A.perceived` and `B.perceived`).

Significance level: The statistical tests were evaluated at a 5% level of significance ($\alpha = 0.05$).

Hypothesis: The hypotheses $H2_0$ and $H2_a$ were adapted to the operational variables and to models A and B as follows:

$H2a_0$**:** There is no significant relationship between the *calculated complexity* (`iv.A.CC`, `iv.A.CL`, `iv.A.CS`, and `iv.A.CAS`) and the subject's *perceived complexity* (`A.perceived`) of the models.

$H2a_0 : \rho = 0$

$H2a_a$**:** There is a significant relationship between the *calculated complexity* (`iv.A.CC`, `iv.A.CL`, `iv.A.CS`, and `iv.A.CAS`) and the subject's *perceived complexity* (`A.perceived`) of the models.

$H2a_a : \rho \neq 0$

$H2b_0$**:** There is no significant relationship between the *calculated complexity* (`iv.B.CC`, `iv.B.CL`, `iv.B.CS`, and `iv.B.CAS`) and the subject's *perceived complexity* (`B.perceived`) of the models.

$H2b_0 : \rho = 0$

$H2b_a$**:** There is a significant relationship between the *calculated complexity* (`iv.B.CC`, `iv.B.CL`, `iv.B.CS`, and `iv.B.CAS`) and the subject's *perceived complexity* (`B.perceived`) of the models.

$H2b_a : \rho \neq 0$

Data analysis: Pearson's or Spearman's correlations were planned depending on data normality. Spearman's correlation was used because data was not confirmed as normal. Two sets of correlations were run, one set for observation A and one set for observation B.

Power analysis: Using G*Power [Fau+07] a power analysis was calculated resulting in the same results provided in Table 8.3 and Figures 8.2 and 8.3.

8.2.3.3 Perceived Complexity and Model Comprehension

This section describes the experimental design used to explore the relationship between the subject's *perceived complexity* of the process models, and the subject's *model comprehension* of the same process models (see hypotheses $H3_0$ and $H3_a$).

Each model (A and B) in the survey had both perceived complexity and model comprehension questions. Perceived complexity was expected to negatively correlate with model comprehension.

Goal: Determine the relationship between perceived complexity (`A.perceived` and `B.perceived`) and model comprehension questions (`A.Correct`, `B.Correct`, `A.Time`, `B.Time`, `A.Efficacy`, `B.Efficacy`, `A.Efficiency`, and `B.Efficiency`).

Independent variables: None.

Dependent variables: Perceived complexity (`A.perceived` and `B.perceived`) and model comprehension questions (`A.Correct`, `B.Correct`, `A.Time`, `B.Time`, `A.Efficacy`, `B.Efficacy`, `A.Efficiency`, and `B.Efficiency`).

Significance level: The statistical tests were evaluated at a 5% level of significance ($\alpha = 0.05$).

Hypothesis: The hypotheses $H3_0$ and $H3_a$ were adapted to the operational variables and to models A and B as follows:

$H3a_0$**:** There is no significant relationship between the subject's *perceived complexity* (`A.perceived`) of the process models, and the subject's *model comprehension* (`A.Correct`, `A.Time`, `A.Efficacy`, and `A.Efficiency`) of the same process models.

$H3a_0 : \rho = 0$

$H3a_a$**:** There is a significant relationship between the subject's *perceived complexity* (`A.perceived`) of the process models, and the subject's *model comprehension* (`A.Correct`, `A.Time`, `A.Efficacy`, and `A.Efficiency`) of the same process models.

$H3a_a : \rho \neq 0$

$H3b_0$**:** There is no significant relationship between the subject's *perceived complexity* (`B.perceived`) of the process models, and the subject's *model comprehension* (`B.Correct`, `B.Time`, `B.Efficacy`, and `B.Efficiency`) of the same process models.

$H3b_0 : \rho = 0$

$H3b_a$**:** There is a significant relationship between the subject's *perceived complexity* (`B.perceived`) of the process models, and the subject's *model comprehension* (`B.Correct`, `B.Time`, `B.Efficacy`, and `B.Efficiency`) of the same process models.

$H3b_a : \rho \neq 0$

Data analysis: Pearson's or Spearman's correlations were planned depending on data normality. Spearman's correlation was used because data was not confirmed as normal. Two sets of correlations were run, one set for observation A and one set for observation B.

Power analysis: Using G*Power [Fau+07] a power analysis was calculated resulting in the same results described in Table 8.3 and Figures 8.2 and 8.3.

8.2.3.4 Pairwise Comparison

This section describes the experimental design used to explore the relationship between the *pairwise ordering* of the process models using *calculated complexity*, as measured by the complexity metrics created for this study (CC, CL, CS, and CAS), and the subject's *pairwise ordering* of the models (see Hypotheses $H4_0$ and $H4_a$).

Each survey contained a single pairwise comparison question. After the subjects were exposed to the comprehension and perceived complexity questions for the two models (A and B), they were asked to pairwise compare the two models.

Goal: Understand which (if any) independent variable order matches the human pairwise comparison order. There are four independent variable orders (one per variable):

CC(Model 1) $<$ CC(Model 2) $<$ CC(Model 3) $<$ CC(Model 4) $<$ CC(Model 5) $<$ CC(Model 6).

CL(Model 1) $<$ CL(Model 4) $=$ CL(Model 5) $=$ CL(Model 6) $<$ CL(Model 2) $=$ CL(Model 3).

CS(Model 2) $<$ CL(Model 6) $<$ CL(Model 4) $<$ CL(Model 3) $<$ CL(Model 1) $<$ CL(Model 5).

CAS(Model 5) $<$ CAS(Model 1) $<$ CAS(Model 3) $<$ CAS(Model 4) $<$ CAS(Model 6) $<$ CAS(Model 2).

Independent variables: Calculated pairwise comparison of (`iv.C.CC`, `iv.C.CL`, `iv.C.CS`, and `iv.C.CAS`), generated from the pairwise matrix calculation.

Dependent variables: Pairwise comparisons variable (`C.Compare`).

Significance level: The statistical tests were evaluated at a 5% level of significance ($\alpha = 0.05$).

Hypothesis: The Hypotheses $H4_0$ and $H4_a$ were adapted to the operational variables as follows:

$H4a_0$**:** There is no significant relationship between the ordering of the *calculated complexity* (`iv.C.CC`, `iv.C.CL`, `iv.C.CS`, and `iv.C.CAS`) and the subject's ordering of the models (`C.Compare`).

$H4a_0 : \mu(m1vs2) = Metric[m1vs2], \ldots, \mu(m5vs6) = Metric[m5vs6]$

Where, $Metric[m1vs2]$ corresponds to the value of `iv.C.CC`, `iv.C.CL`, `iv.C.CS`, or `iv.C.CAS`.

$H4a_a$**:** There is a significant relationship between the ordering of the *calculated complexity* (`iv.C.CC`, `iv.C.CL`, `iv.C.CS`, and `iv.C.CAS`) and the subject's ordering of the models (`C.Compare`).

$H4a_a : \mu(m1vs2) \neq Metric[m1vs2], \ldots, \mu(m5vs6) \neq Metric[m5vs6]$

Data analysis: One-way ANOVA with post-hoc analysis as follows:

1. The ANOVA test will determine whether there is a significant *mean difference between the six models*. The null hypothesis for this test is that there is no difference in the means of the fifteen comparisons (`iv.set` or `iv.C.calc` variables).
2. A post-hoc test will be used to determine the exact ordering of all six models. If the ANOVA test's null hypothesis is rejected, then a one-sided Tukey multiple comparisons test (Tukey's Honest Significant Difference test) can be conducted to identify the order of the six models.

Power analysis: The following power analysis was done using G*Power [Fau+07]:

Table 8.4: A priori power calculation for ANOVA

F tests	ANOVA: Fixed effects, omnibus, one-way		
Analysis:	A priori: Compute required sample size		
		large effect	medium effect
Input:	Effect size f	= **0.4**	**0.25**
	a err prob	= 0.05	0.05
	Power ($1 - \beta$ err prob)	= 0.80	0.80
	Number of groups	= 15	15
Output:	Noncentrality parameter λ	= 21.6000000	19.6875000
	Critical F	= 1.7750306	1.7248562
	Numerator *df*	= 14	14
	Denominator *df*	= 120	300
	Total sample size	= **135**	**315**
	Actual power	= 0.8333867	0.8159594

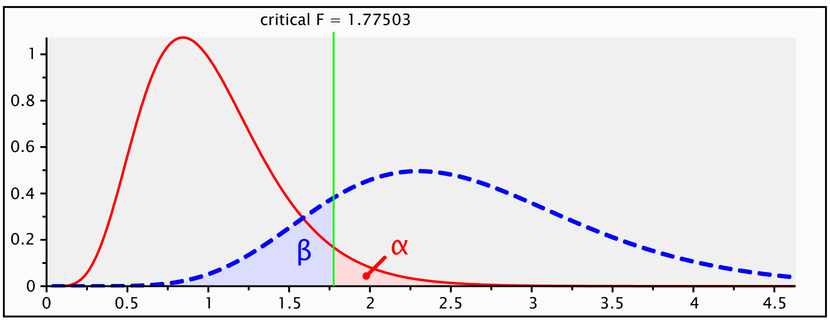


Figure 8.4: A priori power calculation for ANOVA

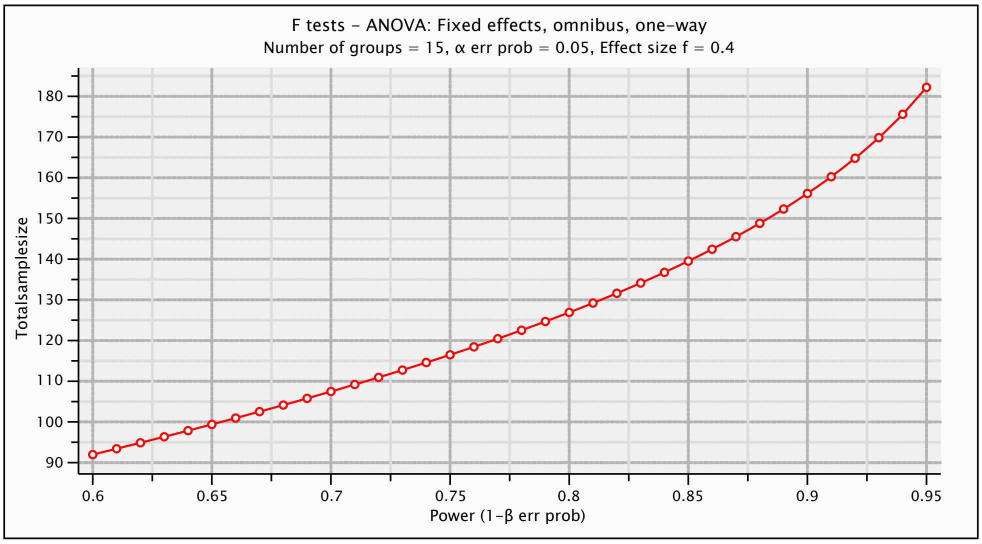


Figure 8.5: Plot of a priori power calculation for ANOVA

Calculating the independent variables

The pairwise comparison values were collected in the ***C.Compare*** variable. This variable used the 9-point scale proposed by Saaty [Saa80] as follows:

1. Model A is absolutely more complex than Model B.
2. Model A is strongly more complex than Model B.
3. Model A is more complex than Model B.
4. Model A is slightly more complex than Model B.
5. Model A and Model B are equally complex.
6. Model B is slightly more complex than Model A.
7. Model B is more complex than Model A.
8. Model B is strongly more complex than Model A.
9. Model B is absolutely more complex than Model A.

The ***C.Compare*** variable compares two models, and so it is equivalent to a $Comparison(i, j)$ variable. The $Comparison(i, j)$ variable defines a 6×6 matrix (using the six CMMN models)

$$M = \begin{pmatrix} 0 & a_{1,2} & \cdots & a_{1,6} \\ a_{2,1} & 0 & \cdots & a_{2,6} \\ \vdots & \vdots & \ddots & \vdots \\ a_{6,1} & a_{6,2} & \cdots & 0 \end{pmatrix}$$

Where $a_{i,j}$ is the mean of $Comparison(i, j)$ (i.e., $\overline{Comparison(i, j)}$). The diagonal $\sum_i^n a_{i,i}$ is empty, because there is no survey comparing a model with itself.

Note that we can always transform $Comparison(i, j)$ into $Comparison(j, i)$, because the values in the 9-point scale are reciprocal. The reciprocal of value v is $|v - 10|$. We will have at least three observations for each $Comparison(i, j)$. But, we could transform $Comparison(j, i)$ into $Comparison(i, j)$ which would give us a total of six observations for $Comparison(i, j)$. In this case, we can transform the matrix into a lower triangular matrix in which all of the entries above the main diagonal are zero:

$$M = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ a_{2,1} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{6,1} & a_{6,2} & \cdots & 0 \end{pmatrix}$$

Note that we can create a 6×6 pairwise comparisons matrix for the independent variable CC, assuming that the range of CC $(max(CC) - min(CC))$ is the maximum spread of complexity between the models. Then we can use the range of CC and the 9-point scale, which is reciprocal with a midpoint of five, to calculate the cells in the matrix as follows:

$$a_{i,j} = 5 + \frac{CC_j - CC_i}{(max(CC) - min(CC))/4} \tag{8.2.1}$$

This equation gives a range from 1 to 9 corresponding to the 9-point scale used in the pairwise comparison question. Using Equation (8.2.1) we get

$$CC = \begin{pmatrix} 5 & 5.95 & 6.43 & 7.29 & 7.86 & 9 \\ 4.05 & 5 & 5.48 & 6.33 & 6.9 & 8.05 \\ 3.57 & 4.52 & 5 & 5.86 & 6.43 & 7.57 \\ 2.71 & 3.67 & 4.14 & 5 & 5.57 & 6.71 \\ 2.14 & 3.1 & 3.57 & 4.43 & 5 & 6.14 \\ 1 & 1.95 & 2.43 & 3.29 & 3.86 & 5 \end{pmatrix}$$

Note that $a_{ij} + a_{ji} = 10$, because the reciprocal of value v is $|v - 10|$ and the lower triangle is reciprocal with the upper triangle. We can rewrite the matrix as a lower triangular matrix as follows:

$$CC = \begin{pmatrix} - & & & & & \\ 4.05 & - & & & & \\ 3.57 & 4.52 & - & & & \\ 2.71 & 3.67 & 4.14 & - & & \\ 2.14 & 3.1 & 3.57 & 4.43 & - & \\ 1 & 1.95 & 2.43 & 3.29 & 3.86 & - \end{pmatrix}$$

We can do the same with the other metrics (*CL*, *CS*, and *CAS*) and will get the following matrices:

$$CL = \begin{pmatrix} - & & & & & \\ 1 & - & & & & \\ 1 & 5 & - & & & \\ 2.33 & 6.33 & 6.33 & - & & \\ 2.33 & 6.33 & 6.33 & 5 & - & \\ 2.33 & 6.33 & 6.33 & 5 & 5 & - \end{pmatrix} \quad CS = \begin{pmatrix} - & & & & & \\ 8.56 & - & & & & \\ 5.89 & 2.33 & - & & & \\ 6.78 & 3.22 & 5.89 & - & & \\ 4.56 & 1 & 3.67 & 2.78 & - & \\ 8.11 & 4.56 & 7.22 & 6.33 & 8.56 & - \end{pmatrix}$$

$$CAS = \begin{pmatrix} - & & & & & \\ 1.44 & - & & & & \\ 4.11 & 7.67 & - & & & \\ 3.22 & 6.78 & 4.11 & - & & \\ 5.44 & 9 & 6.33 & 7.22 & - & \\ 1.89 & 5.44 & 2.78 & 3.67 & 1.44 & - \end{pmatrix}$$

The values of the matrices provide the following independent variables:

iv.C.calc	iv.C.CC	iv.C.CL	iv.C.CS	iv.C.CAS
m2vs1	4.05	1	8.56	1.44
m3vs1	3.57	1	5.89	4.11
m4vs1	2.71	2.33	6.78	3.22
m5vs1	2.14	2.33	4.56	5.44
m6vs1	1	2.33	8.11	1.89
m3vs2	4.52	5	2.33	7.67
m4vs2	3.67	6.33	3.22	6.78
m5vs2	3.1	6.33	1	9
m6vs2	1.95	6.33	4.56	5.44
m4vs3	4.14	6.33	5.89	4.11
m5vs3	3.57	6.33	3.67	6.33
m6vs3	2.43	6.33	7.22	2.78
m5vs4	4.43	5	2.78	7.22
m6vs4	3.29	5	6.33	3.67
m6vs5	3.86	5	8.56	1.44

8.2.3.5 Complexity Weights Validation

This section describes the experimental design used to explore whether the weights of the CMMN symbols used to calculate *CC* (see Table 7.3) are the same as the subject's *perceived* *symbol complexity* (see hypotheses $H5_0$ and $H5_a$).

This experiment was intended to validate the complexity metric CC weights (see Table 7.3). The weights for the complexity metric CC were assigned based on researcher intuition [Mar+15b]. The last section of the survey contained a set of 35 questions using an 8-point Likert scale. Each subject was exposed to a random subset of proximately 12 questions (a third of the 35 questions). Each question corresponds to one of the weights used to calculate CC.

Goal: Validate the weights used to calculate CC.

Independent variables: Weights used to calculate CC.

Dependent variables: `Weights.CasePlan`, ... , `Weights.ExitCritOR` (35 in total).

Significance level: The statistical tests were evaluated at a 5% level of significance ($\alpha = 0.05$).

Hypothesis: The hypotheses $H5_0$ and $H5_a$ were adapted to the operational variables as follows:

Hypothesis	**Description**
$H5.01_0 : \mu(\texttt{Weights.CasePlan}) = 3.33$	Number of case plans
$H5.01_a : \mu(\texttt{Weights.CasePlan}) \neq 3.33$	
$H5.02_0 : \mu(\texttt{Weights.Stage}) = 3.33$	Number of non-discretionary stages
$H5.02_a : \mu(\texttt{Weights.Stage}) \neq 3.33$	
$H5.03_0 : \mu(\texttt{Weights.DStage}) = 5.67$	Number of discretionary stages
$H5.03_a : \mu(\texttt{Weights.DStage}) \neq 5.67$	
$H5.04_0 : \mu(\texttt{Weights.PlanFrag}) = 8$	Number of plan fragments
$H5.04_a : \mu(\texttt{Weights.PlanFrag}) \neq 8$	
$H5.05_0 : \mu(\texttt{Weights.CFileItem}) = 3.33$	Number of case file items
$H5.05_a : \mu(\texttt{Weights.CFileItem}) \neq 3.33$	
$H5.06_0 : \mu(\texttt{Weights.Task}) = 3.33$	Number of non-discretionary tasks
$H5.06_a : \mu(\texttt{Weights.Task}) \neq 3.33$	
$H5.07_0 : \mu(\texttt{Weights.DTask}) = 5.67$	Number of discretionary tasks
$H5.07_a : \mu(\texttt{Weights.DTask}) \neq 5.67$	
$H5.08_0 : \mu(\texttt{Weights.Event}) = 5.67$	Number of event listeners
$H5.08_a : \mu(\texttt{Weights.Event}) \neq 5.67$	
$H5.09_0 : \mu(\texttt{Weights.Milestone}) = 3.33$	Number of milestones
$H5.09_a : \mu(\texttt{Weights.Milestone}) \neq 3.33$	
$H5.10_0 : \mu(\texttt{Weights.Connector}) = 1$	Number of connectors
$H5.10_a : \mu(\texttt{Weights.Connector}) \neq 1$	
$H5.11_0 : \mu(\texttt{Weights.CPlanningT}) = 3.33$	Number of collapsed planning table decorators
$H5.11_a : \mu(\texttt{Weights.CPlanningT}) \neq 3.33$	

Continued on next page

Table 8.6 – *Continued from previous page*

Hypothesis	**Description**
$H5.12_0 : \mu(\texttt{Weights.EPlanningT}) = 5.67$	Number of expanded planning table decorators
$H5.12_a : \mu(\texttt{Weights.EPlanningT}) \neq 5.67$	
$H5.13_0 : \mu(\texttt{Weights.AComplete}) = 5.67$	Number of autocomplete decorators
$H5.13_a : \mu(\texttt{Weights.AComplete}) \neq 5.67$	
$H5.14_0 : \mu(\texttt{Weights.Collapsed}) = 1$	Number of collapsed decorators
$H5.14_a : \mu(\texttt{Weights.Collapsed}) \neq 1$	
$H5.15_0 : \mu(\texttt{Weights.Expanded}) = 3.33$	Number of expanded decorators
$H5.15_a : \mu(\texttt{Weights.Expanded}) \neq 3.33$	
$H5.16_0 : \mu(\texttt{Weights.ManualA}) = 3.33$	Number of manual activation decorators
$H5.16_a : \mu(\texttt{Weights.ManualA}) \neq 3.33$	
$H5.17_0 : \mu(\texttt{Weights.Repetition}) = 3.33$	Number of repetition decorators
$H5.17_a : \mu(\texttt{Weights.Repetition}) \neq 3.33$	
$H5.18_0 : \mu(\texttt{Weights.Required}) = 3.33$	Number of required decorators
$H5.18_a : \mu(\texttt{Weights.Required}) \neq 3.33$	
$H5.19_0 : \mu(\texttt{Weights.HumanIcon}) = 1$	Number of participant markers
$H5.19_a : \mu(\texttt{Weights.HumanIcon}) \neq 1$	
$H5.20_0 : \mu(\texttt{Weights.EntryCritWC}) = 3.33$	Entry criterion with associated connector
$H5.20_a : \mu(\texttt{Weights.EntryCritWC}) \neq 3.33$	
$H5.21_0 : \mu(\texttt{Weights.EntryCrit}) = 5.67$	Entry criterion without a connector
$H5.21_a : \mu(\texttt{Weights.EntryCrit}) \neq 5.67$	
$H5.22_0 : \mu(\texttt{Weights.ExitCritWC}) = 3.33$	Exit criterion with associated connector
$H5.22_a : \mu(\texttt{Weights.ExitCritWC}) \neq 3.33$	
$H5.23_0 : \mu(\texttt{Weights.ExitCrit}) = 8$	Exit criterion without a connector
$H5.23_a : \mu(\texttt{Weights.ExitCrit}) \neq 8$	
$H5.24_0 : \mu(\texttt{Weights.CaseTask}) = 1$	Case task (case plan not included in the same model)
$H5.24_a : \mu(\texttt{Weights.CaseTask}) \neq 1$	
$H5.25_0 : \mu(\texttt{Weights.CaseTasknim}) = 3.33$	Case task (case plan included in the same model)
$H5.25_a : \mu(\texttt{Weights.CaseTasknim}) \neq 3.33$	
$H5.26_0 : \mu(\texttt{Weights.EntryCritAND}) = 1$	AND entry criteria
$H5.26_a : \mu(\texttt{Weights.EntryCritAND}) \neq 1$	
$H5.27_0 : \mu(\texttt{Weights.EntryCritOR}) = 1$	OR entry criteria
$H5.27_a : \mu(\texttt{Weights.EntryCritOR}) \neq 1$	
$H5.28_0 : \mu(\texttt{Weights.ExitCritAND}) = 1$	AND exit criteria
$H5.28_a : \mu(\texttt{Weights.ExitCritAND}) \neq 1$	
$H5.29_0 : \mu(\texttt{Weights.ExitCritOR}) = 1$	OR exit criteria
$H5.29_a : \mu(\texttt{Weights.ExitCritOR}) \neq 1$	
$H5.30_0 : \mu(\texttt{Weights.NBHTask}) = 3.33$	Non-blocking human task
$H5.30_a : \mu(\texttt{Weights.NBHTask}) \neq 3.33$	
$H5.31_0 : \mu(\texttt{Weights.ProcTask}) = 1$	Process task (both discretionary and non)
$H5.31_a : \mu(\texttt{Weights.ProcTask}) \neq 1$	
$H5.32_0 : \mu(\texttt{Weights.BHTask}) = 1$	Blocking human task (both discretionary and non)

Continued on next page

Table 8.6 – *Continued from previous page*

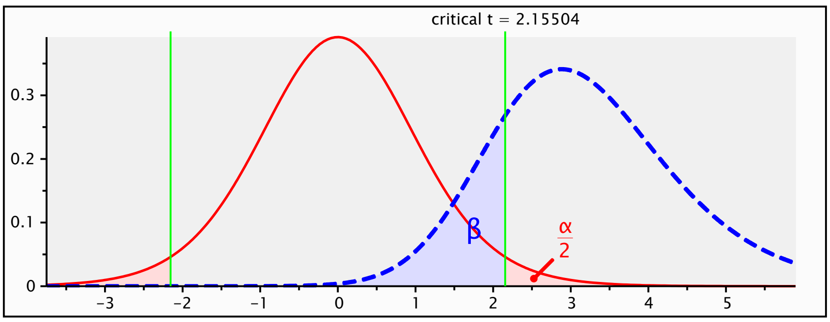
Hypothesis	**Description**
$H5.32_a : \mu(\texttt{Weights.BHTask}) \neq 1$	
$H5.33_0 : \mu(\texttt{Weights.UserEvent}) = 1$	User event listener
$H5.33_a : \mu(\texttt{Weights.UserEvent}) \neq 1$	
$H5.34_0 : \mu(\texttt{Weights.TimerEvent}) = 1$	Timer event
$H5.34_a : \mu(\texttt{Weights.TimerEvent}) \neq 1$	

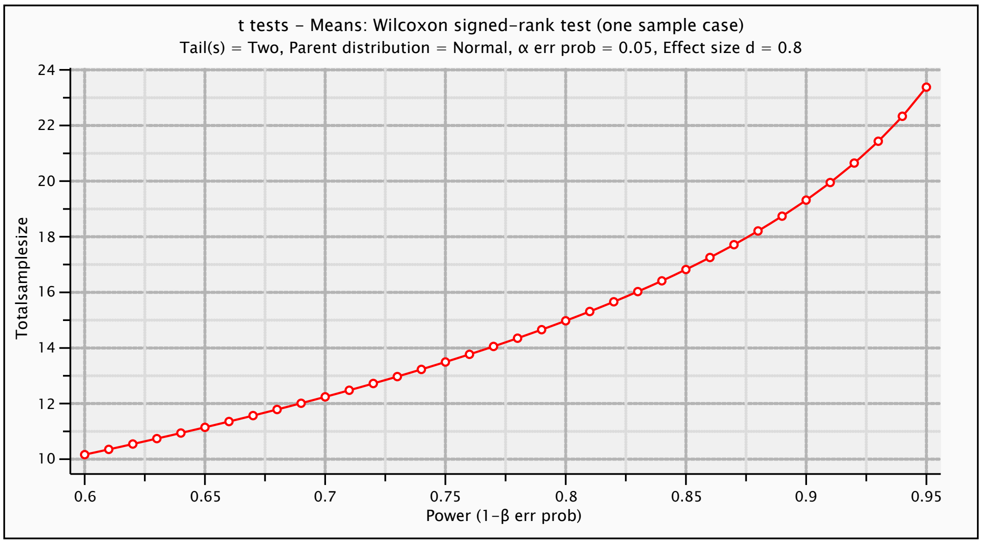
Data analysis: Because the dependent variables (Weights*) are in an ordinal scale, a two-tailed Wilcoxon signed-rank test was used instead of a t-test.

Power analysis: The following power analysis was done using G*Power [Fau+07]:

Table 8.7: A priori power calculation for a two-tailed Wilcoxon signed-rank test

t-tests -	Means: Wilcoxon signed-rank test (one sample case)		
Analysis:	A priori: Compute required sample size		
		large effect	medium effect
Input:	Tail(s)	= Two	= Two
	Parent distribution	= Normal	= Normal
	Effect size d	= **0.80**	= **0.50**
	α err prob	= 0.05	= 0.05
	Power ($1 - \beta$ err prob)	= 0.80	= 0.80
Output:	Noncentrality parameter δ	= 3.0277590	= 2.8906114
	Critical t	= 2.1550415	= 2.0358928
	Df	= 13.3239449	= 32.4225380
	Total sample size	= **15**	= **35**
	Actual power	= 0.8006782	= 0.8006915





Calculating the population mean

The weights used to calculate the complexity metric CC have values from zero to three that were assigned based on researcher intuition (see Table 7.3). The survey question that explores the perceived complexity of the CMMN elements uses an 8-point Likert scale with values from one (very easy) to eight (very difficult) for each of the CMMN symbols utilized in the survey. The subject's answers to that question were encoded in 34 '`Weights.*`' ordinal scale dependent variables (see Table 8.1).

To test hypotheses $H5_0$ and $H5_a$, it was hypothesized that the population mean would correspond with the zero to three values presented in Table 7.3. In order to compare these values with the '`Weights.*`' variables, they needed to be scaled up into the 8-point Likert scale. Therefore, they were linearly rescaled into an 8-point scale, using

$$c = a * value + b \tag{8.2.2}$$

where,

$a = \frac{(new.max - new.min)}{(old.max - oldmin)} = 2.3333$

$b = new.max - a * old.max = 1$

In addition, the resulting c value from Equation (8.2.2) was rounded as follows $\mu = \lfloor c + \frac{1}{2} \rfloor$.

Applying Equation (8.2.2) to rescale the zero to three values in a range between 1 and 8, we get the values presented in Table 8.8:

Table 8.8: Population mean

CC weight	c	μ
0	1	1
1	3.333	3
2	5.666	6
3	7.999	8

Table 8.9: LinkedIn groups postings

LinkedIn group	**Members**	**Posted date**
CMMN	172	6/15/2016
Business Process Model and Notation (BPMN) [OMG13]	6,815	6/23/2016
Business Process Management (BPM) Group	13,757	6/23/2016
FileNet Professionals	5,979	6/23/2016
business process Improvement	108,583	6/27/2016
BPM Guru / BPM Leader	17,128	6/27/2016
Adaptive Case Management (ACM)	1,164	6/27/2016
workflow/BPM	9,729	6/30/2016
IBM Enterprise Content Management	4,431	6/30/2016
BPM Professionals Group	25,746	7/20/2016
BPM Forum	5,884	7/20/2016
IBM Case Manager (Advanced case management)	168	7/20/2016
FileNet Professionals	6,056	7/20/2016
FileNet Alumni	595	7/20/2016
IBM Advanced case management	352	7/20/2016

8.2.4 Participants

The target population of subjects was professionals in the area of process modeling. Subjects were recruited via LinkedIn professional groups invitations (see Table 8.9 and Appendix D file 18 (2016-06-15 Invitation.pdf)) and emails. Invitations were also posted on a few professional websites [Kem16; Mar16b] (see Appendix D files 22 (2016-07-06 Column 2.pdf) and 21 (2016-07-05-ART-Case-Management-Modeling-MMarin.pdf)).

8.2.4.1 Ethical Considerations

This study complied with the University of South Africa's (UNISA) research ethics policy [Uni07]. The subjects were presented with an informed consent form before the start of the survey.

As part of the ethical clearance application, CSET's Research and Ethics Committee (see Appendix D file 15 (2016-05-23 MAMarin_Student_Ethical_Clearance-v5.pdf)) reviewed an initial version of the survey instrument and the experiment description. The research was approved by the CSET's Research and Ethics Committee (see Appendix C).

The first page of the survey contained a detailed description of the experiment including the expected duration and the number of pages of the survey. Subjects were provided with an informed consent question that they had to answer "yes" to in order to proceed with the survey (see Appendix D file 19 (2016-06-15 Survey-Example.pdf)). Subjects had the option to decline participation in the study and to withdraw at any time during the survey and tutorial.

8.2.5 Research Instruments

The research instruments consisted of an online survey (see Appendix D files 19 (2016-06-15 Survey-Example.pdf) and 11 (results-survey338792 (description).pdf)) and a CMMN tutorial (see file 20 (2016-06-15 Survey-Tutorial.pdf)). The tutorial was developed for this research using eXeLearning [eXe15] (see Appendix D files 63 (Tutorial.pdf) and 67 (tutorial.elp)). After the experiment was conducted the tutorial was made available[1] to the public. The survey was developed using LimeSurvey [Lim16] (see Appendix D files 68 (CMMN Complexity metrics project.pdf) and 70 (limesurvey_survey_338792.lss)) and was hosted by a private hosting service[2].

A pilot was conducted to review and test the survey and the tutorial (see Appendix D files 17 (2016-06-10 Survey and Tutorial Pilot.pdf) and 16 (2016-06-08 Pilot-full-answers.pdf)). The pilot was conducted from 31 May 2016 to 9 June 2016. Comments and suggestions from the pilot were implemented in the final instruments.

8.2.5.1 Data Description

The raw data collected from the survey does not contain any identifiable personal information and has been included in the supplementary material (see Appendix D files 12 (results-survey338792.csv), 13 (survey_archive_338792.lsa), 10 (out-comments.txt) and 11 (results-survey338792 (description).pdf)). The raw data was transformed into a data-set suitable for processing using R [R C16] for statistical analysis (see Appendix D files 2 (dataset-all.csv) and 3 (dataset-clean.csv)).

[1]Link to CMMN Tutorial at http://cmmn.byethost4.com
[2]Link to Survey at http://cmmn.limequery.org/

8.2.6 Limitations

In order to maintain subject privacy the survey was open to anyone, and no control was exercised over the subjects. The survey was designed to maintain the privacy of the subjects. No personal identifiable information was collected from the subjects. In the whole survey there were only three mandatory questions that the subjects were required to answer (the letter of informed consent, the tutorial completion, and the pairwise comparison). Therefore, subjects could skip questions and were allowed to abandon the survey at any time. The end result of this flexibility was that it was impossible to control who participated in the survey. Some of the demographic questions were designed to identify subjects' experience and suitability for the study, but these were optional questions.

8.2.6.1 Minimizing Threats to Validity

This section looks at the threats to validity in the context of experimental design. Section 8.3.5 evaluates the threats to validity in the context of measurement.

Internal Validity

Threats to internal validity are those that affect the independent variable thus threatening the relationship between treatment and outcome [Woh+12]. The following steps were taken to minimize threats to internal validity:

Subjects' domain expertize. Instead of using models from a particular domain, the models were generated in a domain agnostic manner and were labeled with letters instead of text labels.

Subject fatigue. To minimize subject fatigue, each subject was exposed to only two models and the order of the models was selected using a round-robin algorithm.

Differences between groups. A round-robin algorithm was used to assign the participants to one of the 30 different combinations of the two models and was also used to minimize differences between groups.

External Validity

Threats to external validity are those that limit the ability to generalize the results to industrial practice [Woh+12]. The following steps were taken to minimize threats to external validity:

Selection. To avoid selection bias, professionals were invited to participate instead of students.

Population. To avoid limiting the population to a few online forum participants, subjects were encouraged to invite others, and a large representative set of professional forums and websites was used.

Location. To minimize geographical location bias, the survey was available to subjects worldwide.

Construct validity

Threats to construct validity are those that affect the ability to generalize the results from the experiment to concepts or theory [Woh+12]. The following steps were taken to minimize threats to construct validity:

Mono-operation bias. In order to avoid a single treatment of an independent variable, six models were used as well as a set of independent variables represented by the metrics.

Mono-method bias. In order to avoid a single type of observation and measurement a set of different experiments were designed and subjects were exposed to multiple questions for similar concepts. For example, each subject was exposed to a perceived complexity question and a pairwise complexity comparison question.

Conclusion validity

Conclusion validity deals with the issues that affect the ability to draw the correct conclusion [Woh+12]. The following steps were taken to minimize threats to conclusion validity:

Sample size. To minimize invalid statistical conclusions, power calculations were conducted for all of the statistical tests that were planned to identify minimal sample sizes.

Model selection. An algorithm was used to generate the models so as to avoid researcher and domain biases (see Appendix D file 27 (The6Models.pdf)).

Arrangement of elements. To avoid subjects having to spend time finding the CMMN elements mentioned in the questions, the elements were labeled consistently from left to right and from top to bottom using an alphabetic sequence.

8.2.7 Procedures

The data was collected from the online survey that was available to participants from 15 June 2016 to 15 August 2016. The survey was hosted by a private hosting service, and the raw data was collected and backed up daily during the two month period.

Subjects were invited via online professional forums or emails. Subjects followed the link in the invitation and were presented with a short description of the project, survey, and tutorial. Subjects who clicked to the next page were counted as participants and were presented with the letter of informed consent. Subjects who accepted the letter of informed consent were routed to the survey. Subjects who completed the demographics form were routed to the tutorial, and after the tutorial they continued with the survey.

The data was processed using R [R C16]. In the spirit of literate research and programming, all of the R scripts that were used have been included in the supplementary material (see Appendix D file 50 (Instructions(read-me-first).pdf)). R scripts were used to transform the raw data into a data set suitable for further statistical processing. R was also used for statistical analysis and sections of this thesis were generated using R.

8.3 Analysis

This section describes the analysis of the data collected from the online survey.

8.3.1 Sample Size

As shown in Figure 1, see Appendix D in file 25 (Basic-stats.pdf), out of the 333 subjects that looked at the survey, only 258 agreed to the informed consent form. Of those, only 106 completed the survey. Two completed surveys were empty with only the mandatory questions having been answered (informed consent, tutorial completion, and pairwise comparison). However, there were four incomplete surveys that provided enough information to test more than one hypothesis.

In addition, answers from subjects who used less than one minute to analyze the model and provide six answers (five comprehension questions and one perceived complexity question) were removed. Only surveys that provided enough information to test more than one hypothesis were used in this study. The number of resulting surveys was 108.

```
Number of usable surveys: 108
Number of samples per hypothesis:
```

```
H1 (Model comprehension): model A 105 model B:  101
H2 (Perceived complexity): model A 104 model B: 100
H3 (Perceived vs comprehension): 100
H4 (Pairwise comparison): 105
H5 (Weights validation): between 6 and 15
```

For hypotheses H1, H2, and H3, a sample size of 80 was required (see Table 8.3). Therefore, enough sample data was available for testing these hypotheses. For hypothesis H4, a sample size of 135 was required (see Table 8.4). This sample size was not reached, therefore the test was considered exploratory research. Finally, for hypothesis H5, a sample size of 15 was required (see Table 8.7). The sample size was not reached, therefore the test was considered exploratory research.

8.3.2 Normality

In order to determine normality, measures of skewness and kurtosis were calculated for the dependent variables using a ratio scale, as well as the Shapiro-Wilk test of normality along with a series of plots. Table 8.10 shows the results. Skewness and kurtosis were found to be within the standard range for most variables, except for time (A.Time and B.Time) where both values (skewness and kurtosis) were found to be high, indicating non-normality. The Shapiro-Wilk tests indicated significant non-normality for all variables.

In addition, a series of plots were constructed in order to further explore the extent of normality of the dependent variables (see Section 4.2, see Appendix D in file 25 (Basic-stats.pdf)). Descriptive statistics for the dependent variables were also calculated (see Section 4.1, see Appendix D in file 25 (Basic-stats.pdf)).

Table 8.10: skewness, kurtosis, and Shapiro test values

Name	N	skewness	kurtosis	W	p
A.Correct	105	-0.3989529	-0.4867852	0.941	$p < .001$
A.Time	105	6.5654974	47.3065292	0.311	$p < .001$
A.Efficacy	105	-0.3989529	-0.4867852	0.941	$p < .001$
A.Efficiency	105	1.2778460	2.5225782	0.911	$p < .001$
B.Correct	101	-0.1581400	-0.6619040	0.951	$p < .001$
B.Time	101	2.9707074	13.4409206	0.736	$p < .001$
B.Efficacy	101	-0.1581400	-0.6619040	0.951	$p < .001$
B.Efficiency	101	1.0547315	1.3000589	0.929	$p < .001$

8.3.3 Times

The 105 subjects who answered model A's questions took an average of 10 minutes (N 105, SD 20) to answer the questions; while the 101 subjects who answered model B's questions took an average of 5 minutes (N 101, SD 4), which seems to indicate either subject fatigue or learning.

Table 8.11: Answering times for model A and model B

Name	N	Mean	SD
A.Time	105	10	20
B.time	101	5	4

8.3.4 Hypothesis Testing

This section describes the statistical analysis conducted in order to test the hypotheses presented in Section 8.2.1.

8.3.4.1 Model Comprehension

Correlations were conducted in order to determine whether negative correlations existed between calculated complexity and model comprehension as predicted in Figure 8.1. Due to the presence of non-normality Spearman's correlation coefficient was used in all of the correlations conducted (see Section 8.3.2). Table 8.12 and Table 8.13 present the results of these correlations. In addition, a set of scatter-plots was created for visual inspection of the relationship between these variables were conducted (see Section 4.3.1, see Appendix D in file 25 (Basic-stats.pdf)). As shown in the tables, no significant correlations were found between these sets of variables, with the exception of `B.Efficiency` against `iv.B.CC` with $\rho(99) = 0.213, p < 0.05$. However, this correlation was not confirmed using `A.Efficiency` against `iv.A.CC` or the scatter-plots.

Looking at the descriptive statistics presented in Section 4.1, see Appendix D in file 25 (Basic-stats.pdf) Table 1, see Appendix D in file 25 (Basic-stats.pdf), the average number of correct answers for model A was 2.87 (N 99, SD 1.4) and for model B 2.98 (N 88, SD 1.33), which seems to indicate that, based on the number of correct answers, the subjects' maintained or showed a very small improvement in how they answered the questions for model B. However, subjects answered the questions related to the second model (model B) in half the time (average 5 minutes (N 101, SD 4)) that they used to answer the questions related to the first model (model A took an average of 10 minutes (N 105, SD 20)). Therefore, a post-hoc analysis using a two-sample Wilcoxon signed-rank test for pair data was conducted to test if the observations for model A and B had the same data distribution. For this test the null hypothesis was that observations (correct answers, time, efficacy, and efficiency) for both model A and model B had the same data distribution. Table 8.14 shows the results. The null hypothesis was rejected for time and efficiency with a $p = 0.001$. This seems to indicate that subjects did answer model B's questions faster (in about half of the time used for model A), and that they were more efficient 0.012 (N 88, SD 0.008) for model B versus 0.009 (N 99, SD 0.006) for model A). However, the null hypothesis cannot be rejected for correct answers and for efficacy. This seems to indicate that subjects maintained the same number of correct answers and efficacy.

Table 8.12: Correlation for model A against independent variables

Measure	iv.A.CC	iv.A.CL	iv.A.CS	iv.A.CAS
A.Correct	0.059(103)	-0.078(103)	0.019(103)	-0.019(103)
A.Time	0.072(103)	-0.098(103)	-0.054(103)	0.054(103)
A.Efficacy	0.059(103)	-0.078(103)	0.019(103)	-0.019(103)
A.Efficiency	-0.057(103)	0.074(103)	0.053(103)	-0.053(103)

Note: each cell contains ρ followed by p and (df). p is empty for $p > 0.05$, * for $p < 0.05$, ** for $p < 0.01$, and *** for $p < 0.001$

Table 8.13: Correlation for model B against independent variables

Measure	iv.B.CC	iv.B.CL	iv.B.CS	iv.B.CAS
B.Correct	0.066(99)	0.104(99)	0.026(99)	-0.026(99)
B.Time	-0.162(99)	0.053(99)	-0.036(99)	0.036(99)
B.Efficacy	0.066(99)	0.104(99)	0.026(99)	-0.026(99)
B.Efficiency	0.213*(99)	0.056(99)	0.026(99)	-0.026(99)

Table 8.14: Paired Wilcoxon test

Name	N	V	p
A.Correct vs. B.Correct	99	1838.50	0.891
A.Time vs. B.Time	99	3972.00	$p < .001$
A.Efficacy vs. B.Efficacy	99	1891.00	0.931
A.Efficiency vs. B.Efficiency	99	1484.00	$p < .001$

8.3.4.2 Perceived Complexity

Correlations were conducted in order to determine whether positive correlations existed between calculated complexity and perceived complexity as predicted in Figure 8.1. Due to the presence of non-normality Spearman's correlation was used. Tables 8.15 and 8.16 present the results of these correlations. In addition, a set of scatter-plots was created for visual inspection of the relationship between these variables (see Section 4.3.2, see Appendix D in file 25 (Basic-stats.pdf)). As shown in the tables, no significant correlations were indicated between these sets of variables, with the exception of `A.perceived` against `iv.A.CC` with $\rho(102) = 0.197, p < 0.05$ and against `iv.A.CAS` with $\rho(102) = -0.197, p < 0.05$. However, this was not confirmed by model B's correlations or scatter-plots.

Table 8.15: Correlation for model A against independent variables

Measure	iv.A.CC	iv.A.CL	iv.A.CS	iv.A.CAS
A.perceived	-0.109(102)	-0.073(102)	0.197*(102)	-0.197*(102)

Table 8.16: Correlation for model B against independent variables

Measure	iv.B.CC	iv.B.CL	iv.B.CS	iv.B.CAS
B.perceived	0.036(98)	-0.062(98)	-0.022(98)	0.022(98)

8.3.4.3 Perceived Complexity and Model Comprehension

Correlations were conducted in order to determine whether negative correlations existed between calculated complexity and model comprehension as predicted in Figure 8.1. Due to the presence of non-normality Spearman's correlation was used. In addition, a set of scatter-plots was created for visual inspection of the relationships between these variables (see Section 4.3.3, see Appendix D in file 25 (Basic-stats.pdf)). Table 8.17 presents the results of these correlations. As shown in Table 8.17, significant correlations were found between perceived complexity (A and B) and `A.Correct`, `A.Efficacy`, and `A.Efficiency`. But these were not confirmed by `B.Correct`, `B.Efficacy`, or `B.Efficiency`; which was likely a result of the fatigue effect.

Table 8.17: Correlation between model comprehension and perceived complexity

Measure	A.perceived	B.perceived
A.Correct	0.276**(102)	0.309**(96)
A.Time	-0.031(102)	0.012(96)
A.Efficacy	0.276**(102)	0.309**(96)
A.Efficiency	0.271**(102)	0.179(96)
B.Correct	0.106(96)	0.123(98)
B.Time	0.154(96)	0.074(98)
B.Efficacy	0.106(96)	0.123(98)
B.Efficiency	-0.084(96)	0.047(98)

8.3.4.4 Pairwise Comparison

The pairwise comparison test for hypotheses $H4_0$, and $H4_a$ described in Section 8.2.3.4 used a one way ANOVA as the first step. For visual inspection of the data, a set of box-plots and frequency plots were created (see Section 4.4, see Appendix D in file 25 (Basic-stats.pdf)). In addition, Table 3, see Appendix D in file 25 (Basic-stats.pdf) shows the descriptive statistics for each group.

First, a post-hoc power analysis calculation was conducted. This was done because the sample size of 105 observations was below that of the required sample size of 135 (see Table 8.4). The post-hoc power analysis calculation was conducted using 105 observations and keeping the rest of the parameters the same as indicated in Table 8.4 for large effect. This resulted in an actual power ($1 - \beta$ err probability) of 68% (0.685) and a critical F of 1.803.

Secondly, The ANOVA test with the null hypothesis $H4_0$ with all the group means being the same ($H4a_0 : \mu(m1vs2) = Metric[m1vs2], \ldots, \mu(m5vs6) = Metric[m5vs6]$) was conducted. Table 8.18 shows a $F(14, 90) = 0.61, p = 0.857$, where the p value is too high to reject the null hypothesis that all the group means are equal. Therefore, the groups may have the same mean, and the Tukey multiple comparisons test was not conducted because the ANOVA null hypothesis was not rejected.

As described in Section 8.2.2.1, the metric CTS was maintained constant at 90 for all of the models, therefore it was not used as a variable in the experiment. The fact that CTS was kept constant means that all of the models are equally complex under CTS, which corresponds to 5 in the 9-point Likert scale being used for `C.Compare`. Therefore, a post-hoc two-tailed Wilcoxon signed-rank test was conducted to test the null hypothesis that $\mu($`C.Compare`$) = 5$. As shown in Table 8.19, the null hypothesis cannot be rejected, indicating that CTS may be the correct complexity metric. A post-hoc power analysis calculation was conducted for the two-tailed Wilcoxon signed-rank test. The effect size was set to 0.50 (medium effect), α to 0.05, and the sample size to 105. This resulted in an actual power ($1 - \beta$ err probability) of 99% (0.998). However, more research is required to test this new hypothesis.

Table 8.18: C.Compare groups ANOVA

Df	Sum Sq	Mean Sq	F value	Pr(>F)
14	33.946	2.425	0.602	0.8570
90	362.568	4.029		

Table 8.19: C.Compare Wilcoxon

Name	N	Mean	SD	mu	V	p
C.Compare	105	4.77	1.95	5	1780.00	0.208

8.3.4.5 Complexity Weights Validation

The dependent variables for the set of weights are ordinal and based on a Likert scale. Therefore, instead of using a one-sample t-test, a Wilcoxon signed-rank test was selected to compare the dependent variables against the hypothesized population mean. Table 8.20 summarizes the results of the Wilcoxon signed-rank test. Statistically significant differences between the weight observations and the hypothesized population means were found with respect to some of the variables as shown in Table 8.20.

However, the sample size of those variables was between 7 and 13, and a post-hoc power calculation was conducted using the same parameters for large effect as shown in Table 8.7, resulting in an actual power ($1 - \beta$ err probability) between 40% (0.406 for a sample size of 7) and 73% (0.731 for a sample size of 13). Which is much lower than the original expected power of 80%. Therefore, there is only a 40% to 73% probability of rejecting the null hypothesis for large effects when it is actually false. A set of frequency plots for the weight dependent variables are shown in Section 4.4.3, see Appendix D in file 25 (Basic-stats.pdf). Each plot shows in parenthesis the hypothesized population mean, which allows for the interpretation of the data presented in Table 8.20.

Table 8.20: Wilcoxon signed-rank test

Name	N	Mean	SD	mu	V	p
Weights.CasePlan	10	1.60	0.97	3	2.50	$p < .05$
Weights.Stage	6	2.00	0.63	3	0.00	$p < .05$
Weights.DStage	13	2.54	1.56	6	0.00	$p < .01$
Weights.PlanFrag	11	3.73	1.79	8	0.00	$p < .01$
Weights.CFileItem	12	1.92	1.16	3	8.50	$p < .05$
Weights.Task	15	2.13	1.46	3	23.00	0.059
Weights.DTask	13	2.77	1.42	6	0.00	$p < .01$
Weights.Event	12	3.25	1.82	6	2.50	$p < .01$
Weights.Milestone	11	2.45	1.44	3	12.00	0.224
Weights.Connector	10	1.90	1.29	1	15.00	0.054
Weights.CPlanningT	12	3.83	1.47	3	19.00	0.09
Weights.EPlanningT	7	3.71	2.81	6	3.50	0.09
Weights.AComplete	10	3.50	1.96	6	0.00	$p < .05$
Weights.Collapsed	10	3.10	2.38	1	21.00	$p < .05$
Weights.Expanded	10	2.80	2.57	3	17.00	0.549
Weights.ManualA	11	3.36	1.75	3	33.50	0.566
Weights.Repetition	9	2.67	2.40	3	18.50	0.673
Weights.Required	14	2.29	1.64	3	25.00	0.152
Weights.EntryCritWC	8	3.25	1.83	3	16.00	0.797
Weights.EntryCrit	8	2.00	0.76	6	0.00	$p < .05$
Weights.ExitCritWC	7	3.71	2.29	3	9.00	0.783
Weights.ExitCrit	12	2.67	1.50	8	0.00	$p < .01$
Weights.NBHTask	10	3.30	1.95	3	27.00	0.624
Weights.ProcTask	8	3.25	1.98	1	28.00	$p < .05$
Weights.CaseTasknim	12	3.92	1.98	3	34.50	0.169
Weights.CaseTask	13	3.38	1.66	1	66.00	$p < .01$
Weights.HumanIcon	12	2.00	1.48	1	15.00	0.057
Weights.TimerEvent	12	2.08	1.08	1	36.00	$p < .05$
Weights.UserEvent	13	2.31	0.85	1	66.00	$p < .01$
Weights.BHTask	6	2.33	1.51	1	10.00	0.098
Weights.EntryCritAND	8	2.75	1.75	1	28.00	$p < .05$
Weights.EntryCritOR	9	3.56	2.01	1	45.00	$p < .01$
Weights.ExitCritAND	9	2.78	1.09	1	36.00	$p < .05$
Weights.ExitCritOR	8	2.88	1.73	1	28.00	$p < .05$

8.3.5 Measurement Validity

Section 8.2.6.1 evaluated the threats to validity in the context of experimental design. This section looks at the threats to validity in the context of measurement. Based on the survey data and data analysis the following threats to validity were identified:

Instrument. In retrospect, the time required to complete the combined survey and tutorial was too long (average 71 minutes (N 108, SD 79)). The tutorial was expected to take around 30 minutes, however, on average it took 44 minutes (N 106, SD 67). On average the survey was completed in 22 minutes (N 108, SD 22).

Fatigue or learning. There was some indication during data analysis that either fatigue or learning had played a role in the results. For example, in model A, subjects spent an average of 10 minutes (N 105, SD 20) answering the questions, whereas in model B the subjects spent an average of 5 minutes (N 101, SD 4).

Data accuracy. As a result of the previous two issues (instrument too long and fatigue) the data may not be sufficiently accurate. In addition, two of the subjects only answered the three mandatory questions required to complete the survey. Three of the subjects spent less than a minute answering model A's five questions and perceived complexity question, which is a concern as these required a detailed review of the model (the spent 7.1, 18.11, and 58.14 seconds each). In model B, six subjects took less than a minute to complete the five questions and the perceived complexity question (they spent 9.8, 12.53, 18.66, 35.96, 38.8, and 39.21 seconds each).

Sampling bias. The invitation to participate in the survey and tutorial was posted in several LinkedIn professional forums and on some websites. Only those who visited these forums and sites during the period that the survey was active were reached, and could opt to participate in the survey. Therefore, not all of the subscribers to the professional LinkedIn forums were reached. Subjects were encouraged to invite other participants via email. Participants in the LinkedIn CMMN forum received an email invitation, subscribers to BPtrends also received an email that was linked to an article containing an invitation to participate (see Appendix D file 21 (2016-07-05-ART-Case-Management-Modeling-MMarin.pdf)). Therefore, some groups may have had more representation than others.

Sample size. The expected sample size, as calculated by the power analysis, was not reached for some of the hypothesis testing (H4 and H5 – see Section 8.3.1). For hypotheses H4 and H5 the sample size required for the statistical tests was not achieved. Therefore, the results from those tests were considered exploratory research.

8.3.6 Findings

Although careful planning, sound experimental design, and efforts to minimize threats to validity (see Section 8.2.6.1) were in place, the experiment experienced some problems as described in Section 8.3.5. In retrospect, the experiment was too complex and required too many subjects to be conducted using an unsupervised online survey.

No evidence was found to support any of the four complexity metrics. This experiment failed to reject the five null hypotheses described in Section 8.2.1. Therefore, more research is required to re-test those hypotheses with smaller and more targeted experiments.

Evidence that a variable that was not included in this experiment may predict CMMN model complexity was uncovered (see Section 8.3.4.4). CMMN total size metric *CTS* defined in Definition 7.3 as $CTS = CS + CAS$, was kept constant at 90 (see Table 8.2). This was done to keep the number of icons constant in all of the six models used for the experiment. However, this meant that all of the six models were equally complex under *CTS*, and if that is true then `C.Compare` should have been five for all of the pairwise comparisons. Therefore, a post-hoc two-tailed Wilcoxon signed-rank test was conducted to test the null hypothesis that $\mu(\texttt{C.Compare}) = 5$. It was unable to reject the null hypothesis, which indicates a high likelihood of all of the six models being equally complex. *CTS* could also explain why no significant correlation was found between the metrics and model comprehension or perceived complexity. The results obtained (inability to reject the five null hypotheses) are consistent with all of the models being equally complex (under *CTS* in this case). However, more research will be required to test this new hypothesis.

What seems to be a learning effect was also detected (see Section 8.3.4.1). Subjects answered the questions related to the second model (model B) in half the time that they used to answer the questions related to the first model. Although this could be attributed to fatigue, the number of correct answers between the two models was maintained. This seems to indicate that subjects did answer model B's questions faster, and that they were more efficient, while they maintained the same number of correct answers and efficacy.

8.4 Summary

This chapter attempted to empirically validate the CMMN metrics that were defined in Chapter 7. It contributed a novel approach to the empirical validation of complexity metrics using pairwise comparisons. This novel approach to pairwise comparisons used to validate complexity metrics was not uncovered during the SLR presented in Chapter 6. In addition, this study justified the sample size used to validate the metrics. It used a larger sample size than any other study uncovered by the SLR, and mainly targeted professionals.

The experiment presented in this chapter failed to reject the five null hypotheses described in Section 8.2.1, which is not surprising considering that the SLR found that 94% of the 115 metrics that underwent empirical validation failed it at least once. However, it found evidence that a variable not included in this experiment, for methodological reasons, may predict CMMN model complexity. *CTS* was kept constant for the duration of this experiment, which means that all of the models were equally complex under this variable. A post-hoc test designed to identify if subjects found all of the models equally complex, seemed to indicate that there is evidence to support this new hypothesis. Further research on the contribution of *CTS* to CMMN model complexity is required.

Chapter 9

Conclusion and Future Areas for Investigation

This last chapter summarizes the results of this thesis. The purpose of this research was to explore complexity metrics for artifact-centric business process models. The results of the Systematic Literature Review (SLR) that was conducted for this thesis confirmed that to date, all of the proposed complexity metrics for process models have been for imperative process models rather than for artifact-centric business process models. This research was based on the assumption that complexity metrics for declarative process models can be derived from the research that has been conducted on complexity metrics for imperative process models. This thesis focused on Case Management Model and Notation (CMMN) [OMG14a] because of the potential practical impact that such research may have on vendors and users of this emerging standard.

The goal was to use CMMN as a proxy to fill the gap in the literature concerning complexity metrics for declarative process models. As such, this research contributes new knowledge to the literature by exploring complexity metrics and user comprehension of CMMN models. This research also makes contributions in the areas of formalizing CMMN by using first-order logic, identifying and comparing the model complexity of CMMN against other process modeling notations, clarifying the relationship between CMMN and Guard-Stage-Milestone (GSM) [Hul+11b], and proposing complexity metrics for CMMN.

Since this thesis contains exploratory research into complexity metrics that can be used by Business Artifact (BA) and CMMN, the work touches on what may seem, at first glance, to be unrelated topics. In order to begin investigating what metrics could be used in CMMN it was important to first understand the relationships between BA, GSM, and CMMN. This led to the use of the Business Artifacts with Lifecycle Services and Associations (BALSA) framework to foster such an understanding. In order to explore complexity metrics for CMMN, it was essential to understand CMMN method complexity and how it compared to other methods. An in-depth SLR into complexity metrics for process models was required to ground the CMMN complexity metrics in Business Process Management (BPM). Therefore, seemingly unrelated topics were covered in order to achieve the exploration into complexity metrics for BA and CMMN.

This chapter is organized as follows. Section 9.1 summarizes the findings and contributions of this study. Section 9.2 describes the implications of this study for researchers, practitioners, and vendors implementing CMMN. Section 9.3 describes the limitations of this study. Finally, Section 9.4 offers recommendations for future research.

9.1 Discussion of Findings

This thesis started by introducing the topic of BPM and business process modeling in Chapter 2, and found no suitable framework that could categorize all of the different business process modeling approaches. Therefore, a synthesis of the different approaches was proposed that used the dimensions advanced by Sowa and Zachman [SZ92] and Zachman [Zac87], the core concepts put forward by Caetano et al. [Cae+12] and Pereira et al. [Per+11a; Per+11b], the focus areas used by Sousa et al. [Sou+07], and the perspectives offered by Giaglis [Gia01]. This synthesis provided a framework for describing the different business process modeling approaches and for organizing them into a coherent system. This is important because as new modeling approaches are described, a consistent way of categorizing them is required.

The BALSA framework was used to understand CMMN's relationship to the BA approach and to compare CMMN with GSM (see Chapter 3). Formal descriptions of CMMN that were consistent with GSM formalisms were created, including a CMMN program, a case type, and a case model. Although important differences were found between CMMN and GSM, which were described in Section 3.4, for the most part the approaches are consistent and a clear development from GSM to CMMN can be observed.

The method complexity of CMMN was evaluated in Chapter 4 using the meta-model-based approach introduced by Rossi and Brinkkemper [RB96]. The results were compared to other popular process modeling methods, including Business Process Model and Notation (BPMN) [OMG13], Unified Modeling Language Activity Diagram (UML AD) [OMG09c], and Event-driven Process Chain (EPC) [van99], all of which have undergone similar evaluations by other researchers [Ind+09b; OMG14a; SC02; U.S09; zH08; zR08]. A set of counting principles was developed to evaluate the method complexity for meta-models described using Unified Modeling Language (UML) [OMG09c]. These counting principles were required because Rossi and Brinkkemper developed the meta-model approach based on Object, Property, Relationship, Role (OPRR) [RB96], and researchers using UML for method evaluations did not disclose how to deal with UML idiosyncrasies. This thesis found that CMMN compares favorably with BPMN, UML AD, and EPC.

Formal transformations in two directions between CMMN case types and GSM artifact types were developed and presented in Chapter 5. The transformations helped to further clarify the relationship between CMMN and GSM. As expected, the transformation from a GSM artifact type into a case type was found to be relatively simple and straight forward. The resulting case type modeled using CMMN was visually similar to the original artifact type. However, the transformation of a CMMN case type into a GSM artifact type was found to be far more complex, requiring the development of a set of patterns that corresponded to the CMMN entities' life cycles, and a syntax directed translation grammar with a set of rewriting rules in order for this to be accomplished. Despite this difficulty, such transformations are useful because they allow the formal verification work that was developed for GSM [Bel+12; Dam+11; Gon+15; Sol+13b] to be applied to CMMN, and may also allow for the formal operational semantics of GSM [Esh+13; Hul+11a] to be used for CMMN.

Chapter 6 presented an SLR on complexity metrics for process models that had been described in the literature of the last 20 years (from January 1996 to June 2016 inclusive) and how these were validated. Some of the findings include:

- The uncovering of 40 primary papers describing 206 non-duplicated metrics, of which 115 had undergone controlled empirical validations, and 78 had undergone theoretical validations.
- No complexity metrics for declarative process models or for data-centric process models were uncovered.
- Confirmed Polančič and Cegnar's [PC16] and Sánchez-González et al.'s [S+10a] findings that although a large number of metrics have been proposed not all of them have been validated.
- There was no consistency in how metrics were being empirically validated. Different researchers utilized different concepts to operationalize complexity. It was therefore common for a metric to undergo several empirical validations against multiple concepts. This led to metrics being validated against some concepts and failing some of the validations against other concepts. In particular, 94% of metrics that had undergone empirical validation had failed that validation for at least one other concept.

A set of metrics for CMMN was developed in Chapter 7. Based on the analysis of the metrics uncovered by the SLR it was found that some metrics, which were based on counting elements and other metrics which were based on weights could be adapted to CMMN. Therefore, the formalization of CMMN (described in Section 3.3) was used to formally describe a set of metrics for CMMN based on counting elements and weights. The metrics were theoretically validated using the formal framework for software measurements defined by Briand et al. [Bri+96] and the complexity metrics were further validated using Weyuker's properties [Wey88] for software complexity measures.

A novel approach to empirically validate complexity metrics based on pairwise comparisons was developed and implemented in Chapter 8 to validate the proposed CMMN metrics. In addition, the findings of the SLR were used to improve the experimental design and design protocol for the empirical validation. Therefore, this study departed from most empirical validations of complexity metrics for process models in several important ways:

1. It used power calculations to estimate the smallest acceptable sample size. Although, power calculations are a common statistical tool used to estimate sample size the SLR did not uncover any paper that used power calculations or any other method to justify the sample size.

2. It mainly targeted professional process modelers instead of students. The SLR found that more than 74% of the papers had used students as subjects.

3. It used a within-subjects pairwise comparison experimental design. The SLR uncovered only nine papers that used this experimental design [Car06b; Gar+04a; Gar+04b; Gar+04c; Gar+04d; Rol+07a; Rol+08; Rol+09a; Rol+09b] but none of them had performed a pairwise comparison.

4. It used a larger sample size than any of the other empirical validations uncovered in the SLR.

The experiment presented in Chapter 8 failed to empirically validate any of the proposed CMMN complexity metrics. This was not surprising considering that the SLR found that 94% of the 115 metrics that underwent empirical validation failed at least once. However, it found evidence that *CTS*, a metric not included in the experiment for methodological reasons, may predict CMMN model complexity.

9.2 Implications

This exploratory research on complexity metrics for CMMN improves the understanding of complexity metrics for declarative processes and has practical implications for researchers, implementors and users of CMMN. This section describes the implications of this research for each group.

The implications of this study for the research community are outlined below:

- The research contributes to the understanding of model complexity for BPM by adding declarative processes to the current knowledge regarding model complexity for imperative processes. This opens new areas of research on complexity metrics for declarative processes.
- By having uncovered trends in the areas of complexity metrics for process models via the SLR, specifically the lack of standardized empirical validation research methods and sample size calculations, this study opens new opportunities for research into these areas.
- The study provides the basis for further studies on method complexity for specifications using UML because it complements the work done by Indulska et al. [Ind+09b], Recker et al. [Rec+09], and Siau and Cao [SC02] in the area of method complexity, and it developed a set of counting principles that can be used to evaluate the method complexity of UML based specifications.
- By clarifying and formalizing the relationship between GSM and CMMN, new areas of research have become possible including the verification of CMMN processes, and the exploration into the visualization of GSM processes.
- This study advances research on CMMN [Hau+15; Kur+15; Sch+13] and provides the basis for future research by formalizing and identifying complexity metrics for CMMN.
- It complements the work done by Belardinelli et al. [Bel+12], Eshuis and van Gorp [Ev15], and Solomakhin et al. [Sol+12] on GSM transformations by adding CMMN to the mix of transformations, which opens the possibility of research into the relationship between CMMN and Data-Centric Dynamic Systems (DCDS).

This study has implications for vendors standardizing and implementing CMMN as outlined below:

- CMMN is a new process standard and the formalization and comparison against GSM could inform the evolution of CMMN.
- The transformation procedures between CMMN and GSM provide the basis for understanding and defining operational semantics for CMMN.
- Products based on CMMN may benefit from implementing some of the metrics from this research. Although the metrics were not empirically validated, these could still be used for other purposes including comparing CMMN models.
- Vendors implementing other BPM products could benefit from the SLR by using it to identify useful metrics for their products.

This study also has implications for practitioners as outlined below:

- The characterization of the CMMN method complexity provides new insights into method complexity and advances the ability to compare method notations. This could help practitioners select the right technology for their projects.

- Practitioners could benefit from the outcome of the SLR by identifying metrics for their projects that have undergone both theoretical and empirical validation and by understanding which concepts have been tested.

9.3 Limitations

This study has focused on CMMN as a representative approach of business artifact-centric business process models. Although this research has explored the relationship between CMMN with BA, particularly with GSM, its results may not be applicable to other artifact-centric business process model approaches because, as discussed by Hull [Hul08], changes to the four dimensions of the BALSA framework will produce different BA approaches.

The SLR presented in Chapter 6 was conducted by a single researcher, which may be viewed as problematic as it may introduce researcher bias. This study tried to minimize researcher bias by developing a strict research protocol during the planning phase of the review and by following the guidelines and recommendations put forward by Biolchini et al. [Bio+05], Kitchenham [Kit04], Kitchenham and Charters [KC07], Kitchenham et al. [Kit+04], Sjoberg et al. [Sjo+07], and Wohlin et al. [Woh+12]. Although, these precautions should have minimized researcher bias, it cannot be ruled out completely.

9.4 Recommendations for Future Research

This section provides recommendations for future research projects based on the contributions of this thesis.

The synthesis of the different process modeling approaches (see Table 2.1) introduced in Chapter 2 provided a basic framework for describing the different business process modeling approaches. Further work could develop this basic framework into a method for standardizing the terminology and categorizing the process modeling approaches. Additionally, a way to map the different process modeling approaches into a formal schema is required to better organize and understand the strengths and weaknesses of each approach. This will allow for a better and more objective comparison between the different process modeling approaches.

The formal descriptions of CMMN, including a program, a case type, and a case model presented in Chapter 3, provide the basis for future research into formal verification procedures for CMMN models and to develop formal operational semantics.

The evaluation of method complexity for CMMN, which was done in Chapter 4, provides the basis for further work to calibrate the meta-model method complexity proposed by Rossi and

Brinkkemper [RB96] with the UML meta-model. This is important, because organizations, e.g. the Object Management Group (OMG) [Gro17], currently use the UML meta-model to describe process modeling methods. The work done by Rossi and Brinkkemper used the OPRR method's modeling language as implemented in MetaEdit [Smo+91] to describe meta-models. However, most of the modern methods use UML to describe their meta-models. The use of UML introduces new nuances to the meta-models that were not present in OPRR. Therefore, research should be conducted to calibrate Rossi and Brinkkemper's approach with UML.

Standard specifications evolve and seem to become more complex over time. Therefore, the method complexity presented in Chapter 4 carefully identified the version of the specifications being used. Should a researcher take up the suggestion to recalibrate Rossi and Brinkkemper's [RB96] approach to UML, then additional research will be required to calculate method complexity for the latest versions of the standard specifications, including BPMN, CMMN, EPC, UML, and others.

Another avenue for future research is to identify subsets of the CMMN notation. As process modelers begin to use CMMN, it will be useful to identify the subsets of the CMMN specification that start to emerge. This is similar to the work that was conducted by zur Muehlen and Ho [zH08], zur Muehlen and Recker [zR08], and zur Muehlen et al. [zur+07] for BPMN.

Following Rossi and Brinkkemper's [RB96] suggestion, an empirical validation of the method complexity developed in Chapter 4 should be conducted. The meta-model-based method complexity approach proposed by Rossi and Brinkkemper and used in this research provides an analysis of the conceptual part of the techniques and methods, but it does not provide empirical validation.

Chapter 5 developed formal transformations in two directions between CMMN case types and GSM artifact types. Although we are able to translate from CMMN to GSM, the resulting models are extremely verbose and complex. Trying to formalize the CMMN execution semantics based on the transformation is difficult. Therefore, future work needs to be done to describe formal execution semantics for CMMN based on the CMMN specification and GSM's formal semantics.

The SLR that was provided on complexity metrics for process models and described in Chapter 6 only uncovered complexity metrics for imperative process models. Therefore, there is room for research on complexity metrics for declarative process models.

This SLR did not collect the statistical information used to validate metrics that was provided in some of the papers. This information was not collected because this study was not a meta-analysis. An additional area of future research may consist of collecting the statistical information used to validate metrics and conducting a meta-analysis with such information.

The set of metrics for CMMN described in Chapter 7 complies with the formal framework for software measurements put forward by Briand et al. [Bri+96], and with the nine properties described by Weyuker [Wey88]. However, both Briand et al. and Weyuker assume an imperative style based on directed acyclic graphs. Therefore, research is required to understand the applicability of Briand et al.'s framework and Weyuker's properties to declarative systems. Alternatively, research is required to define an axiomatic approach to the theoretical validation of metrics for declarative process models.

CMMN models have non-visual entities (see Tables 3.3 and 3.4). Some of these entities are roles and non-visualized system events. Empirical work is needed to understand the influence of CMMN's non-visual entities on potential complexity metrics.

This thesis did not explore all of the metrics uncovered by the SLR that could be adapted to CMMN (see Table 7.1). Further work is required to explore and adapt some of the other metrics to CMMN.

Although the empirical experiments described in Chapter 8 failed to empirically validate the proposed complexity metrics for CMMN, these did find evidence that *CTS*, a metric not included in the experiment for methodological reasons, may predict CMMN model complexity. Therefore, work to empirically validate *CTS* is required.

9.5 Summary

This chapter presented a summary of the thesis, its implications as well as its limitations. The list of recommendations for further research in the area of complexity metrics for declarative process models shows that this is an area that will benefit from more research.

This thesis includes a CD containing an electronic copy of this thesis, and all of the supplementary material described in Appendix D. In the spirit of reproducible research all of the code and data used and collected during this research can be found on the CD. In addition, parts of the text, statistics, and calculations that appear in Chapters 6 and 8 were generated using R source code that, following literate programming, includes the description and all of the steps required to reproduce these calculations.

Bibliography

[Abi+09] S. Abiteboul, P. Bourhis, A. Galland, and B. Marinoiu. "The AXML Artifact Model". In: *2009 16th International Symposium on Temporal Representation and Reasoning*. 2009, pp. 11–17. DOI: 10.1109/TIME.2009.9 (cit. on p. 60).

[Abr+10] F. B. E. Abreu, R. D. B. V. D. Porciúncula, J. M. Freitas, and J. C. Costa. "Definition and Validation of Metrics for ITSM Process Models". In: *Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the*. 2010, pp. 79–88. DOI: 10.1109/QUATIC.2010.13 (cit. on pp. xxiii, xxiv, xxvii, xxix–xxxiii, xxxvii, 106, 136, 137, 256, 263, 266, 268–281, 348, 349, 360, 362, 365–374, 376–381).

[Aho+07] A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman. *Compilers: Principles, Techniques, and Tools*. 2nd ed. Pearson Education, Inc., 2007 (cit. on pp. 20, 77, 100, 299).

[AR16] B. Allehyani and S. Reiff-Marganiec. "Maintaining Goals of Business Processes during Runtime Reconfigurations". In: *Proceedings of the 8th ZEUS Workshop, Vienna, Austria, January 27-28, 2016*. 2016, pp. 21–28 (cit. on p. 23).

[Ant+11] A. Antonini, A. M. Ferreira, S. Morasca, and G. Pozzi. "Software Measures for Business Processes". In: *ADBIS 2011, Research Communications, Proceedings II of the 15th East-European Conference on Advances in Databases and Information Systems, September 20-23, 2011, Vienna, Austria*. 2011, pp. 11–22 (cit. on pp. xxiv, xxxv, 24, 25, 124, 137, 251, 258, 259, 284, 285, 352, 354, 358, 362, 363, 367, 384, 385).

[Anu+15] I. G. Anugrah, R. Sarno, and R. N. E. Anggraini. "Decomposition Using Refined Process Structure Tree (RPST) and Control Flow Complexity Metrics". In: *2015 International Conference on Information Communication Technology and Systems (ICTS)*. 2015, pp. 203–208. DOI: 10.1109/ICTS.2015.7379899 (cit. on p. 357).

[AM14] N. Ari and N. Mamatnazarova. “Programming Languages”. In: *2014 11th International Conference on Electronics, Computer and Computation (ICECCO).* 2014, pp. 1–8. DOI: 10.1109/ICECCO.2014.6997548 (cit. on p. 20).

[Bag+13] B. Bagheri Hariri, D. Calvanese, G. de Giacomo, A. Deutsch, and M. Montali. “Verification of Relational Data-centric Dynamic Systems with External Services”. In: *Proceedings of the 32Nd Symposium on Principles of Database Systems.* PODS ’13. New York, New York, USA: ACM, 2013, pp. 163–174. DOI: 10.1145/2463664.2465221 (cit. on pp. 36, 40, 78, 259).

[BR05] W. Bandara and M. Rosemann. “What Are the Secrets of Successful Process Modelling? Insights From an Australian Case Study”. In: *Journal of Strategic Information Systems* 10.3 (2005), pp. 47–68 (cit. on pp. 14, 255).

[BB14] M. Ch. Barukh and B. Benatallah. “ProcessBase: A Hybrid Process Management Platform”. In: *Service-Oriented Computing: 12th International Conference, ICSOC 2014, Paris, France, November 3-6, 2014. Proceedings.* Ed. by X. Franch, A. K. Ghose, G. A. Lewis, and S. Bhiri. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 16–31. DOI: 10.1007/978-3-662-45391-9_2 (cit. on pp. 19, 20).

[Bec+03] J. Becker, P. Delfmann, Th. Falk, and R. Knackstedt. “Multiperspektivische Ereignisgesteuerte Prozessketten”. German. In: *Proceedings des 2. GI-Workshop EPK Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten.* Bamberg, Germany, 2003, pp. 45–60 (cit. on p. 68).

[Bei+92] W. W. Bein, J. Kamburowski, and M. F. M. Stallmann. “Optimal Reduction of Two-Terminal Directed Acyclic Graphs”. In: *SIAM Journal on Computing* 21.6 (1992), pp. 1112–1129. DOI: 10.1137/0221065 (cit. on p. 310).

[Bel+12] F. Belardinelli, A. Lomuscio, and F. Patrizi. “Verification of GSM-Based Artifact-Centric Systems through Finite Abstraction”. In: *Service-Oriented Computing.* Ed. by C. Liu, H. Ludwig, F. Toumani, and Q. Yu. Vol. 7636. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 17–31. DOI: 10.1007/978-3-642-34321-6_2 (cit. on pp. 5, 41, 43, 78, 195, 197).

[BE91] J. D. Berkley and R. G. Eccles. *Rethinking the Corporate Workplace: Case Managers at Mutual Benefit Life.* Case N9-492-015. Boston, MA: Harvard Business School, 1991, pp. 1–13. DOI: 10.1225/492015 (cit. on pp. 26, 28).

[Bha+05] K. Bhattacharya, R. Guttman, K. Lyman, F. T. Heath III, S. Kumaran, P. Nandi, F. Wu, P. Athma, C. Freiberg, L. Johannsen, and A. Staudt. "A Model-driven Approach to Industrializing Discovery Processes in Pharmaceutical Research". In: *IBM Systems Journal* 44.1 (2005), pp. 145–162. DOI: 10.1147/sj.441.0145 (cit. on pp. 31, 32).

[Bha+07] K. Bhattacharya, C. Gerede, R. Hull, R. Liu, and J. Su. "Towards Formal Analysis of Artifact-Centric Business Process Models". In: *Business Process Management: 5th International Conference, BPM 2007, Brisbane, Australia, September 24-28, 2007. Proceedings*. Ed. by G. Alonso, P. Dadam, and M. Rosemann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 288–304. DOI: 10.1007/978-3-540-75183-0_21 (cit. on pp. 36, 37, 252).

[Bha+09] K. Bhattacharya, R. Hull, and J. Su. "A Data-centric Design Methodology for Business Processes". In: *Handbook of Research on Business Process Modeling*. IGI Global, 2009, pp. 503–531. DOI: 10.4018/978-1-60566-288-6.ch023 (cit. on p. 40).

[Bin+14] N. Bin Ali, K. Petersen, and C. Wohlin. "A Systematic Literature Review on the Industrial Use of Software Process Simulation". In: *Journal of Systems and Software* 97 (2014), pp. 65–85. DOI: 10.1016/j.jss.2014.06.059 (cit. on pp. 111, 112, 133).

[Bio+05] J. C. Biolchini, P. G. Mian, A. C. C. Natali, and G. H. Travassos. *Systematic Review in Software Engineering*. Tech. rep. Technical Report RT-ES 679/05. Systems Engineering and Computer Science Department, COPPE / UFRJ, 2005 (cit. on pp. 103, 132, 133, 198).

[BÏ1] M. Böhringer. "Emergent Case Management for Ad-hoc Processes: A Solution Based on Microblogging and Activity Streams". In: *Business Process Management Workshops*. Ed. by M. zur Muehlen and J. Su. Vol. 66. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2011, pp. 384–395. DOI: 10.1007/978-3-642-20511-8_36 (cit. on p. 28).

[Boo+99] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1999 (cit. on p. 68).

[Bor+09a] S. Borgert, M. Dehmer, and E. Aitenbichler. "A Comparative Study of Complexity Measures to Analyze Business Process Models". In: *Proceedings of the International Symposium on Understanding Intelligent and Complex Systems, UICS 2009*. 2009 (cit. on pp. xxii, xxvii, xxxiv, xxxviii, 124, 260, 264, 282, 288, 340, 341, 363, 364, 382, 386).

[Bor+09b] S. Borgert, M. Dehmer, and E. Aitenbichler. “On quantitative Network Complexity Measures for Business Process Models”. In: *Acta Universitates Apulensis* 18 (2009), p. 10 (cit. on p. 340).

[BA14] M. Bouneffa and A. Ahmad. “The Change Impact Analysis in BPM Based Software Applications: A Graph Rewriting and Ontology Based Approach”. In: *Enterprise Information Systems: 15h International Conference, ICEIS 2013, Angers, France, July 4-7, 2013, Revised Selected Papers*. Ed. by S. Hammoudi, J. Cordeiro, A. L. Maciaszek, and J. Filipe. Springer International Publishing, 2014, pp. 280–295. DOI: 10.1007/978-3-319-09492-2_17 (cit. on p. 13).

[BPM04] BPMI.org. *Business Process Modeling Notation (BPMN) v 1.0*. Tech. rep. BPMI.org, 2004, p. 296 (cit. on p. 67).

[Bra+10] L. Braubach, A. Pokahr, K. Jander, W. Lamersdorf, and B. Burmeister. “Go4Flex: Goal-Oriented Process Modelling”. In: *Intelligent Distributed Computing IV: Proceedings of the 4th International Symposium on Intelligent Distributed Computing - IDC 2010, Tangier, Morocco, September 2010*. Ed. by M. Essaaidi, M. Malgeri, and C. Badica. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 77–87. DOI: 10.1007/978-3-642-15211-5_9 (cit. on p. 23).

[Bri+95] L. C. Briand, Kh. E. Emam, and S. Morasca. *Theoretical and Empirical Validation of Software Product Measures*. Tech. rep. International Software Engineering Research Network, 1995, pp. 1–22 (cit. on pp. 107, 155, 344).

[Bri+96] L. C. Briand, S. Morasca, and V. R. Basili. “Property-based Software Engineering Measurement”. In: *IEEE Transactions on Software Engineering* 22.1 (1996), pp. 68–86. DOI: 10.1109/32.481535 (cit. on pp. 4, 107, 124, 131, 135, 138, 143, 144, 147, 149, 151, 156, 157, 196, 200, 317).

[Bri96] S. Brinkkemper. “Method Engineering: Engineering of Information Systems Development Methods and Tools”. In: *Information and Software Technology* 38.4 (1996), pp. 275–280. DOI: 10.1016/0950-5849(95)01059-9 (cit. on p. 265).

[Bru16] G. Bruno. “Tasks and Assignments in Case Management Models”. In: *Procedia Computer Science* 100 (2016), pp. 156–163. DOI: 10.1016/j.procs.2016.09.135 (cit. on p. 78).

[Buc+11] A. Bucchiarone, M. Pistore, H. Raik, and R. Kazhamiakin. “Adaptation of Service-based Business Processes by Context-aware Replanning”. In: *2011 IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*. Dec. 2011, pp. 1–8. DOI: 10.1109/SOCA.2011.6166209 (cit. on p. 24).

[Buc+12] A. Bucchiarone, A. Marconi, M. Pistore, and H. Raik. "CAptEvo: Context-Aware Adaptation and Evolution of Business Processes". In: *Service-Oriented Computing - ICSOC 2011 Workshops: ICSOC 2011, International Workshops WESOA, NFPSLAM-SOC, and Satellite Events, Paphos, Cyprus, December 5-8, 2011. Revised Selected Papers.* Ed. by G. Pallis, M. Jmaiel, A. Charfi, S. Graupner, Y. Karabulut, S. Guinea, F. Rosenberg, Q. Z. Sheng, C. Pautasso, and S. Ben Mokhtar. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 252–254. DOI: 10.1007/978-3-642-31875-7_35 (cit. on p. 24).

[Buc+15] T. Bucher, D. Raber, and R. Winter. "A Taxonomy of Business Process Management Approaches". In: *Handbook on Business Process Management 2: Strategic Alignment, Governance, People and Culture.* Ed. by J. vom Brocke and M. Rosemann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 203–225. DOI: 10.1007/978-3-642-45103-4_8 (cit. on p. 20).

[Bur+12] A. Burattin, F. M. Maggi, W. M. P. van der Aalst, and A. Sperduti. "Techniques for a Posteriori Analysis of Declarative Processes". In: *Enterprise Distributed Object Computing Conference (EDOC), 2012 IEEE 16th International.* 2012, pp. 41–50. DOI: 10.1109/EDOC.2012.15 (cit. on pp. 1, 21).

[Bur11] E. V. Burns. "Case Management 101: 10 Things You Must Know About Case Management". In: *Taming the Umpredictable, Real-world Adaptive Case Management:Case Studies and Practical Guidance.* Ed. by L. Fischer. Excellence in Practice Series. Lighthouse Point, FL, USA: Future Strategies Inc., 2011. Chap. Case Management 101: 10 Things You Must Know About Case Management, pp. 17–26 (cit. on p. 28).

[Cae+12] A. Caetano, C. Pereira, and P. Sousa. "Generation of Business Process Model Views". In: *4th Conference of ENTERprise Information Systems - aligning technology, organizations and people (CENTERIS 2012)* 5 (2012), pp. 378–387. DOI: 10.1016/j.protcy.2012.09.042 (cit. on pp. 18, 19, 194).

[Cal+01] C. Calero, M. Piattini, and M. Genero. "Empirical Validation of Referential Integrity Metrics". In: *Information and Software Technology* 43.15 (2001), pp. 949–957. DOI: 10.1016/S0950-5849(01)00202-6 (cit. on pp. 106, 107, 131).

[CG08] M. Calisti and D. Greenwood. "Goal-Oriented Autonomic Process Modeling and Execution for Next Generation Networks". In: *Modelling Autonomic Communications Environments: Third IEEE International Workshop, MACE 2008, Samos Island, Greece, September 22-26, 2008. Proceedings.* Ed. by

S. van der Meer, M. Burgess, and S. Denazis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 38–49. DOI: 10.1007/978-3-540-87355-6_4 (cit. on p. 23).

[Cal+13] D. Calvanese, G. de Giacomo, and M. Montali. "Foundations of Data-aware Process Analysis: A Database Theory Perspective". In: *Proceedings of the 32Nd Symposium on Principles of Database Systems*. PODS '13. New York, New York, USA: ACM, 2013, pp. 1–12. DOI: 10.1145/2463664.2467796 (cit. on p. 23).

[Cal+15] D. Calvanese, M. Montali, F. Patrizi, and A. Rivkin. "Implementing Data-Centric Dynamic Systems over a Relational DBMS". In: *Alberto Mendelzon International Workshop on Foundations of Data Management*. 2015, p. 209 (cit. on pp. 36, 40, 78, 259).

[Can+05] G. Canfora, F. García, M. Piattini, F. Ruíz, and C. A. Visaggio. "A Family of Experiments to Validate Metrics for Software Process Models". In: *The Journal of systems and software* 77.2 (2005), pp. 113–129. DOI: 10.1016/j.jss.2004.11.007 (cit. on pp. 313, 315).

[Can+14] J. Cano, G. Delaval, and E. Rutten. "Coordination of ECA Rules by Verification and Control". In: *Coordination Models and Languages: 16th IFIP WG 6.1 International Conference, COORDINATION 2014, Held as Part of the 9th International Federated Conferences on Distributed Computing Techniques, DisCoTec 2014, Berlin, Germany, June 3-5, 2014, Proceedings*. Ed. by E. Kühn and R. Pugliese. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 33–48. DOI: 10.1007/978-3-662-43376-8_3 (cit. on p. 261).

[Car05a] J. Cardoso. "About the Data-Flow Complexity of Web Processes". In: *6th International Workshop on Business Process Modeling, Development, and Support: Business Processes and Support Systems: Design for Flexibility*. Porto, Portugal, 2005, pp. 67–74 (cit. on pp. 315, 331).

[Car05b] J. Cardoso. "Control-flow Complexity Measurement of Processes and Weyuker's Properties". In: *Enformatika In Transactions on Engineering, Computing, and Technology* 8 (2005), pp. 213–218 (cit. on pp. xxi, xxiii, 124, 137, 251, 255, 256, 258, 285, 315, 316, 321, 324, 330, 333, 335, 338, 342, 343, 349, 351, 354, 362, 367).

[Car05c] J. Cardoso. "Evaluating the Process Control-Flow Complexity Measure". In: *Proceedings of the IEEE International Conference on Web Services*. ICWS '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 803–804. DOI: 10.1109/ICWS.2005.57 (cit. on p. 315).

[Car05d] J. Cardoso. "How to Measure the Control-flow Complexity of Web Process and Workflows". In: *Workflow Handbook 2005.* Ed. by L. Fischer. Lighthouse Point, Florida: Future Strategies Inc., 2005, pp. 199–212 (cit. on pp. 25, 26, 160, 315, 322, 327, 330, 343).

[Car06a] J. Cardoso. "Approaches to Compute Workflow Complexity". In: *Dagstuhl Seminar: The Role of Business Processes in Service Oriented Architectures.* Dagstuhl, Germany, 2006, p. 15 (cit. on p. 320).

[Car06b] J. Cardoso. "Process Control-flow Complexity Metric: An Empirical Validation". In: *IEEE International Conference on Services Computing.* IEEE, 2006, pp. 167–173. DOI: 10.1109/SCC.2006.82 (cit. on pp. 159, 160, 196, 258, 265, 281, 315, 322, 333, 350, 362).

[Car07a] J. Cardoso. "Business Process Quality Metrics: Log-Based Complexity of Workflow Patterns". In: *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS: OTM Confederated International Conferences CoopIS, DOA, ODBASE, GADA, and IS 2007, Vilamoura, Portugal, November 25-30, 2007, Proceedings, Part I.* Ed. by R. Meersman and Z. Tari. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 427–434. DOI: 10.1007/978-3-540-76848-7_30 (cit. on pp. xxvii, 25, 264, 331, 332, 365).

[Car07b] J. Cardoso. "Complexity Analysis of BPEL Web Processes". In: *Software Process: Improvement and Practice* (2007) (cit. on pp. xxiv, 138, 259, 315, 330, 331, 362).

[Car08] J. Cardoso. "Business Process control-Flow complexity: Metric, Evaluation, and Validation". In: *International Journal of web Services research* 5.June (2008), pp. 49–76 (cit. on pp. xxiv, 5, 124, 160, 251, 284, 322, 333, 355, 357, 362, 384).

[Car+06] J. Cardoso, J. Mendling, G. Neumann, and H. A. Reijers. "A Discourse on Complexity of Process Models". In: *BPM 2006 Workshops, Workshop on Business Process Design.* Ed. by J. Eder and S. Dustdar. Vol. pp. Vienna, Austria: Springer-Verlag, 2006, pp. 115–126 (cit. on pp. xxiv, xxv, xxviii, xxxi, xxxiv, xxxviii, 24, 25, 137, 256, 258, 264, 266, 267, 282, 284, 320, 321, 327, 335, 342, 352–354, 356, 360, 363, 367, 368, 382, 384).

[CV15] F. Caron and J. Vanthienen. "Exploring Business Process Modelling Paradigms and Design-time to Run-time Transitions". In: *Enterprise Information Systems* 0.0 (2015), pp. 1–24. DOI: 10.1080/17517575.2014.986291 (cit. on p. 20).

[CN05] N. S. Caswell and A. Nigam. "Agility = change + coordination". In: *Seventh IEEE International Conference on E-Commerce Technology Workshops*. 2005, pp. 131–139. DOI: 10.1109/CECW.2005.3 (cit. on p. 30).

[Cha+09] T. Chao, D. Cohn, A. Flatgard, S. Hahn, M. Linehan, P. Nandi, A. Nigam, F. Pinel, J. Vergo, and F. Wu. "Artifact-Based Transformation of IBM Global Financing". In: *Business Process Management*. Ed. by U. Dayal, J. Eder, J. Koehler, and H. A. Reijers. Vol. 5701. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pp. 261–277. DOI: 10.1007/978-3-642-03848-8_18 (cit. on pp. 47, 263).

[CP08] Y. C. Chen and V. Prabhu. "Complexity Model for Business Process Analysis". In: *IIE Annual Conference and Expo 2008*. 2008 (cit. on pp. xxiv, 288, 336, 386).

[Che08] C. Cheng. "Complexity and Usability Models for Business Process Analysis". PhD thesis. Penn State, 2008 (cit. on pp. xxvii, 285, 335, 336, 385).

[Chi+13] C. M. Chiao, V. Künzle, and M. Reichert. "Schema Evolution in Object and Process-Aware Information Systems: Issues and Challenges". In: *Revised Papers Business Process Management Workshops: BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012*. Ed. by M. La Rosa and P. Soffer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 328–339. DOI: 10.1007/978-3-642-36285-9_37 (cit. on pp. 21, 34).

[Chi+14] C. M. Chiao, V. Kuenzle, and M. Reichert. "A Tool for Supporting Object-Aware Processes". In: *2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations*. 2014, pp. 410–413. DOI: 10.1109/EDOCW.2014.69 (cit. on pp. 22, 23).

[CK94] S. R. Chidamber and C. F. Kemerer. "A Metrics Suite for Object Oriented Design". In: *IEEE Transactions on Software Engineering* 20.6 (1994), pp. 476–493. DOI: 10.1109/32.295895 (cit. on p. 311).

[CH09] D. Cohn and R. Hull. "Business Artifacts: A Data-centric Approach to Modeling Business Operations and Processes". In: *IEEE Data Engineering Bulletin* 32.3 (2009), pp. 3–9 (cit. on p. 29).

[Con+86] S. D. Conte, H. E. Dunsmore, and V. Y. Shen. *Software Engineering Metrics and models*. Benjamin/Cummings Publishing Company, Inc., 1986 (cit. on pp. 25, 285).

[Çoş14] E. Çoşkun. "A New Complexity Metric for Business Process Models". Master. Atilim University, 2014, p. 80 (cit. on pp. xxii, 5, 106, 124, 125, 136, 137, 142, 257, 355, 360).

[Cur+92] B. Curtis, M. I. Kellner, and J. Over. “Process Modeling”. In: *Communications of the ACM* 35.9 (1992), pp. 75–90. DOI: 10.1145/130994.130998 (cit. on p. 16).

[Dam+11] E. Damaggio, R. Hull, and R. Vaculín. “On the Equivalence of Incremental and Fixpoint Semantics for Business Artifacts with Guard-Stage-Milestone Lifecycles”. In: *Proceedings of the 9th international conference on Business Process Management*. Ed. by S. Rinderle-Ma, F. Toumani, and K. Wolf. Vol. 6896. Lecture Notes in Computer Science. Clermont-Ferrand: Springer, 2011, pp. 396–412 (cit. on pp. 36, 37, 40, 41, 46, 78, 195, 255, 280).

[Dan+96] M. Daneva, R. Heib, and A. W. Scheer. *Benchmarking Business Process Models*. Research Report 136. Institute for Information Systems, University of the Saarland, 1996 (cit. on pp. xxi, 3, 24, 124, 252, 253, 305–307, 359).

[DN94] T. Davenport and N. Nohria. “Case Management and the Integration of Labor”. In: *MIT Sloan Management Review* 35.2 (1994), pp. 11–23 (cit. on pp. 26–28).

[dD10] T. de Bruin and G. Doebeli. “An Organizational Approach to BPM: The Experience of an Australian Transport Provider”. In: *Handbook on Business Process Management 2: Strategic Alignment, Governance, People and Culture*. Ed. by J. vom Brocke and M. Rosemann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 559–577. DOI: 10.1007/978-3-642-01982-1_26 (cit. on p. 12).

[de +16] R. M. de Carvalho, H. Mili, J. Gonzalez-Huerta, A. Boubaker, and A. Leshob. “Comparing ConDec to CMMN - Towards a Common Language for Flexible Processes”. In: *Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development - Volume 1: MODELSWARD,* 2016, pp. 233–240. DOI: 10.5220/0005688002330240 (cit. on p. 38).

[de +15a] G. de Giacomo, M. Dumas, F. M. Maggi, and M. Montali. “Declarative Process Modeling in BPMN”. In: *Advanced Information Systems Engineering: 27th International Conference, CAiSE 2015, Stockholm, Sweden, June 8-12, 2015, Proceedings*. Ed. by J. Zdravkovic, M. Kirikova, and P. Johannesson. Springer International Publishing, 2015. Chap. Declarative Process Modeling in BPMN, pp. 84–100. DOI: 10.1007/978-3-319-19069-3_6 (cit. on pp. 1, 21).

[de 09] H. de Man. *Case Management: Cordys Approach*. Accessed: October 06, 2016. BPTrends. 2009. URL: http://www.bptrends.com/publicationfiles/02%2D09%2DART%

2DBPTrends%20%2D%20Case%20Management%2DDeMan%20%2Dfinal.doc.pdf (cit. on p. 37).

[de +10] H. de Man, Sh. Prasad, and Th. van Donge. "Innovation Management". In: *Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done.* Ed. by K. D. Swenson. 1st ed. Tampa, Florida, USA: Meghan-Kiffer Press, 2010. Chap. Innovation Management, pp. 211–255 (cit. on p. 28).

[de +15b] R. de Masellis, D. Lembo, M. Montali, and D. Solomakhin. "Semantic Enrichment of GSM-Based Artifact-Centric Models". In: *Journal on Data Semantics* 4.1 (2015), pp. 3–27. DOI: 10.1007/s13740-014-0036-6 (cit. on pp. 36, 40, 46, 280).

[de +14] R. M. de Morais, S. Kazan, S. I. D. D. Pádua, and A. L. Costa. "An Analysis of BPM Lifecycles: From a Literature Review to a Framework Proposal". In: *Business Process Management Journal* 20.3 (2014), pp. 412–432. DOI: 10.1108/BPMJ-03-2013-0035 (cit. on p. 13).

[Deb+10] N. Debnath, C. Salgado, M. Peralta, D. Riesco, and G. Montejano. "Optimization of the Business Process Metrics Definition According to the BPDM Standard and Its Formal Definition in OCL". In: *Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications - AICCSA 2010.* AICCSA '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 1–8. DOI: 10.1109/AICCSA.2010.5586986 (cit. on p. 347).

[di +14] C. di Ciccio, A. Marrella, and A. Russo. "Knowledge-Intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches". In: *Journal on Data Semantics* 4.1 (2014), pp. 29–57. DOI: 10.1007/s13740-014-0038-4 (cit. on pp. 13, 19, 20, 75, 264).

[di +15] C. di Ciccio, F. M. Maggi, M. Montali, and J. Mendling. "Ensuring Model Consistency in Declarative Process Discovery". In: *Business Process Management: 13th International Conference, BPM 2015, Innsbruck, Austria, August 31 – September 3, 2015, Proceedings.* Ed. by R. H. Motahari-Nezhad, J. Recker, and M. Weidlich. Springer International Publishing, 2015. Chap. Ensuring Model Consistency in Declarative Process Discovery, pp. 144–159. DOI: 10.1007/978-3-319-23063-4_9 (cit. on pp. 1, 21).

[DT01] M. Dumas and A. H. M. Ter Hofstede. "UML Activity Diagrams As a Workflow Specification Language". In: *Proceedings of the 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts, and Tools.* London, UK, UK: Springer-Verlag, 2001, pp. 76–90 (cit. on p. 67).

[Dum+13] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers. *Fundamentals of Business Process Management*. 2013. DOI: 10.1007/978-3-642-33143-5 (cit. on pp. 12, 13, 282).

[Dum+16] M. Dumas, R. Hull, and F. Patrizi. "Guest Editorial: Special Issue on Data and Artifact-centric Business Processes". In: *Computing* 98.4 (2016), pp. 343–344. DOI: 10.1007/s00607-016-0486-9 (cit. on pp. 21, 23).

[DD08] T. Dybå and T. Dingsøyr. "Empirical Studies of Agile Software Development: A Systematic Review". In: *Information and Software Technology* 50.9-10 (2008), pp. 833–859. DOI: 10.1016/j.infsof.2008.01.006 (cit. on pp. 111, 133).

[Egg06] L. Egghe. "Theory and Practise of the G-index". In: *Scientometrics* 69.1 (2006), pp. 131–152 (cit. on p. 354).

[Eik+11] T. R. Eikebrokk, J. Iden, D. H. Olsen, and A. L. Opdahl. "Understanding the Determinants of Business Process Modelling in Organisations". In: *Business Process Management Journal* 17.4 (2011), pp. 639–662. DOI: 10.1108/14637151111149465 (cit. on p. 255).

[Ev15] R. Eshuis and P. van Gorp. "Synthesizing Data-centric Models from Business Process Models". In: *Computing* 98.4 (2015), pp. 345–373. DOI: 10.1007/s00607-015-0442-0 (cit. on pp. 5, 20–22, 197).

[Esh+13] R. Eshuis, R. Hull, Y. Sun, and R. Vaculín. "Splitting GSM Schemas: A Framework for Outsourcing of Declarative Artifact Systems". In: *Business Process Management*. Ed. by F. Daniel, J. Wang, and B. Weber. Vol. 8094. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 259–274. DOI: 10.1007/978-3-642-40176-3_22 (cit. on pp. 36, 41, 47–49, 82, 195).

[Esh+16] R. Eshuis, S. Debois, T. Slaats, and T. Hildebrandt. "Deriving Consistent GSM Schemas from DCR Graphs". In: *Service-Oriented Computing: 14th International Conference, ICSOC 2016, Banff, AB, Canada, October 10-13, 2016, Proceedings*. Ed. by Q. Z. Sheng, E. Stroulia, S. Tata, and S. Bhiri. Springer International Publishing, 2016, pp. 467–482. DOI: 10.1007/978-3-319-46295-0_29 (cit. on p. 78).

[eXe15] eXeLearning Development Team. *eXeLearning version 2.0.4*. 2015 (cit. on pp. 179, 394).

[Eys12] G. Eysenbach. "Improving the Quality of Web Surveys: the Checklist for Reporting Results of Internet E-Surveys (CHERRIES)". In: *Journal of Medical Internet Research* 14 (1 2012), p. 34. DOI: 10.2196/jmir.6.3.e34 (cit. on pp. 158, 256, 391).

[Fab+13] S. Fabbri, E. Hernandes, A. di Thommazo, A. Belgamo, A. Zamboni, and C. Silva. "Using Information Visualization and Text Mining to Facilitate the Conduction of Systematic Literature Reviews". In: *Enterprise Information Systems: 14th International Conference, ICEIS 2012, Wroclaw, Poland, June 28 - July 1, 2012, Revised Selected Papers.* Ed. by J. Cordeiro, A. L. Maciaszek, and J. Filipe. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 243–256. DOI: 10.1007/978-3-642-40654-6_15 (cit. on p. 104).

[Fab+16] S. Fabbri, C. Silva, E. Hernandes, F. Octaviano, A. di Thommazo, and A. Belgamo. "Improvements in the StArt Tool to Better Support the Systematic Review Process". In: *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering.* EASE '16. Limerick, Ireland: ACM, 2016, pp. 211–215. DOI: 10.1145/2915970.2916013 (cit. on pp. 24, 104, 285, 392).

[Fah+09] D. Fahland, D. Lübke, J. Mendling, H. Reijers, B. Weber, M. Weidlich, and S. Zugal. "Declarative versus Imperative Process Modeling Languages: The Issue of Understandability". In: *Enterprise, Business-Process and Information Systems Modeling: 10th International Workshop, BPMDS 2009, and 14th International Conference, EMMSAD 2009.* Ed. by T. Halpin, J. Krogstie, S. Nurcan, E. Proper, R. Schmidt, P. Soffer, and R. Ukor. Springer Berlin Heidelberg, 2009, pp. 353–366. DOI: 10.1007/978-3-642-01862-6_29 (cit. on pp. 2, 21).

[Fah+10] D. Fahland, J. Mendling, H. A. Reijers, B. Weber, M. Weidlich, and S. Zugal. "Declarative versus Imperative Process Modeling Languages: The Issue of Maintainability". In: *Business Process Management Workshops: BPM 2009 International Workshops, Ulm, Germany, September 7, 2009.* Ed. by S. Rinderle-Ma, Sh. Sadiq, and F. Leymann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 477–488. DOI: 10.1007/978-3-642-12186-9_45 (cit. on pp. 1, 2).

[Fau+07] F. Faul, E. Erdfelder, A. Lang, and A. Buchner. "G*Power 3: a Flexible Statistical Power Analysis Program for the Social, Behavioral, and Biomedical Sciences." In: *Behavior research methods* 39.2 (2007), pp. 175–91 (cit. on pp. 165, 167, 169, 170, 176, 393).

[Fen90] N. E. Fenton. "Software Metrics: Theory, Tools and Validation". In: *Software Engineering Journal* 5.1 (1990), pp. 65–78 (cit. on p. 305).

[FP98] N. E. Fenton and Sh. L. Pfleeger. *Software Metrics: A Rigorous and Practical Approach.* 2nd ed. Boston, MA, USA: PWS Publishing Co., 1998 (cit. on pp. 125, 159).

[Fig+10] K. Figl, J. Mendling, M. Strembeck, and J. C. Recker. "On the Cognitive Effectiveness of Routing Symbols in Process Modeling Languages". In: *Proceedings of the 13th International Conference on Business Information Systems*. Ed. by W. Abramowicz and R. Tolksdorf. Berlin: Springer, 2010, pp. 230–241 (cit. on pp. 136, 142).

[Fig+13] K. Figl, J. Mendling, and M. Strembeck. "The Influence of Notational Deficiencies on Process Model Comprehension". In: *Journal of the Association for Information Systems* 14.6 (2013) (cit. on p. 15).

[Fle94] A. Fleischmann. *Distributed Systems: Software Design and Implementation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994. DOI: 10.1007/978-3-642-78612-9_1 (cit. on p. 23).

[Fle10] A. Fleischmann. "What Is S-BPM?" In: *S-BPM ONE – Setting the Stage for Subject-Oriented Business Process Management: First International Workshop, Karlsruhe, Germany, October 22, 2009. Revised Selected Papers*. Ed. by H. Buchwald, A. Fleischmann, D. Seese, and C. Stary. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 85–106. DOI: 10.1007/978-3-642-15915-2_7 (cit. on p. 286).

[Fle+12] A. Fleischmann, W. Schmidt, Ch. Stary, S. Obermeier, and E. Börger. *Subject-Oriented Business Process Management*. Springer Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-32392-8 (cit. on p. 23).

[Fle+13] A. Fleischmann, U. Kannengiesser, W. Schmidt, and Ch. Stary. "Subject-Oriented Modeling and Execution of Multi-agent Business Processes". In: *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 02*. WI-IAT '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 138–145. DOI: 10.1109/WI-IAT.2013.102 (cit. on p. 23).

[Fle+14] A. Fleischmann, W. Schmidt, and Ch. Stary. "Tangible or Not Tangible – A Comparative Study of Interaction Types for Process Modeling Support". In: *Proceedings, Part II, Human-Computer Interaction. Advanced Interaction Modalities and Techniques: 16th International Conference, HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014*. Ed. by M. Kurosu. Springer International Publishing, 2014. Chap. Tangible or Not Tangible – A Comparative Study of Interaction Types for Process Modeling Support, pp. 544–555. DOI: 10.1007/978-3-319-07230-2_52 (cit. on p. 14).

[FS15] M. Fleming and J. Silverstein. *Worldwide Business Process Management Software Market Shares, 2014: Tactical Spending Drives Growth*. Marketing report August. IDC, 2015, pp. 1–21 (cit. on p. 11).

[Fu+10] X. Fu, P. Zou, Y. Ma, Y. Jiang, and K. Yue. “A Control-Flow Complexity Measure of Web Service Composition Process”. In: *Services Computing Conference (APSCC), 2010 IEEE Asia-Pacific*. 2010, pp. 712–716. DOI: 10.1109/APSCC.2010.27 (cit. on pp. xxii, 124, 284, 343, 385).

[Gar+03] F. García, F. Ruíz, J. A. Cruz, and M. Piattini. “Integrated Measurement for the Evaluation and Improvement of Software Processes”. In: *Software Process Technology: 9th European Workshop, EWSPT 2003, Helsinki, Finland, September 1-2, 2003. Proceedings*. Ed. by F. Oquendo. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 94–111. DOI: 10.1007/978-3-540-45189-1_8 (cit. on pp. xxii, xxix, xxxii–xxxiv, 137, 251, 266, 267, 269, 274–276, 278, 279, 283, 286, 287, 311–313, 316–322, 328, 342, 348–350, 358, 367, 377, 378, 380, 381, 383, 384).

[Gar+04a] F. García, F. Ruíz, and M. Piattini. “An Experimental Replica to Validate a Set of Metrics for Software Process Models”. In: *11th European conference Software process improvement*. Trondheim, Norway: Springer, 2004, pp. 79–90 (cit. on pp. xxx, xxxiv, 25, 128, 129, 157, 196, 269, 273, 276, 277, 283, 311–313, 315, 316, 318, 319, 328, 349, 350, 358, 367, 370, 374, 377, 378, 381, 383, 384).

[Gar+04b] F. García, F. Ruíz, and M. Piattini. “Definition and Empirical Validation of Metrics for Software Process Models”. In: *Product Focused Software Process Improvement: 5th International Conference, PROFES 2004, Kansai Science City, Japan, April 5-8, 2004. Proceedings*. Ed. by F. Bomarius and H. Iida. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 146–158. DOI: 10.1007/978-3-540-24659-6_11 (cit. on pp. 120, 125, 128, 158, 196, 312, 358, 367, 370, 374, 377, 378, 381, 383, 384).

[Gar+04c] F. García, M. Piattini, F. Ruíz, G. Canfora, and C. A. Visaggio. “FMESP: Framework for the Modeling and Evaluation of Software Processes”. In: *Proceedings of the 2004 Workshop on Quantitative Techniques for Software Agile Process*. QUTE-SWAP ’04. New York, NY, USA: ACM, 2004, pp. 5–13. DOI: 10.1145/1151433.1151435 (cit. on pp. 128, 196, 312, 358, 367, 370, 374, 377, 378, 381, 383, 384).

[Gar+04d] F. García, F. Ruíz, and M. Piattini. “Proposal of Metrics for Software Process Models”. In: *Software Measurement European Forum (SMEF 2004)*. Rome, 2004, pp. 406–416 (cit. on pp. 120, 128, 196, 312, 358, 367, 370, 374, 377, 378, 381, 383, 384).

[Gar+06] F. García, M. Piattini, F. Ruíz, G. Canfora, and C. A. Visaggio. “FMESP: Framework for the Modeling and Evaluation of Software Processes”. In:

Journal of Systems Architecture 52.11 (2006), pp. 627–639. DOI: 10.1016/j.sysarc.2006.06.007 (cit. on p. 312).

[Gia01] G. M. Giaglis. "A Taxonomy of Business Process Modeling and Information Systems Modeling Techniques". In: *International Journal of Flexible Manufacturing Systems* 13.2 (2001), pp. 209–228. DOI: 10.1023/A:1011139719773 (cit. on pp. 16, 18, 19, 194).

[GM05] R. J. Glushko and T. McGrath. *Document Engineering:Analyzing and Designing Documents for Business Informatics and Web Services.* MIT Press, Cambridge, MA, 2005 (cit. on pp. 31, 32).

[God15] N. Godha. "Improving the Understandability of Artifact-centric Workflows using BPMN with Extensions". MA thesis. Unitec Institute of Technology, 2015 (cit. on p. 39).

[Gon+15] P. Gonzalez, A. Griesmayer, and A. Lomuscio. "Verification of GSM-Based Artifact-Centric Systems by Predicate Abstraction". In: A. P. Barros, D. Grigori, C. N. Narendra, and K. H. Dam. *Service-Oriented Computing: 13th International Conference.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. Chap. Verification of GSM-Based Artifact-Centric Systems by Predicate Abstraction, pp. 253–268. DOI: 10.1007/978-3-662-48616-0_16 (cit. on pp. 1, 78, 195).

[Gro17] Object Management Group. *About OMG.* Accessed: January 30, 2017. 2017. URL: http://www.omg.org/gettingstarted/gettingstartedindex.htm (cit. on pp. 1, 37, 48, 66, 199, 280).

[GL06a] V. Gruhn and R. Laue. "Adopting the Cognitive Complexity Measure for Business Process Models". In: *5th IEEE International Conference on Cognitive Informatics, 2006. ICCI 2006.* Vol. 1. 2006, pp. 236–241. DOI: 10.1109/COGINF.2006.365702 (cit. on pp. xxiii, 25, 136, 137, 142, 257, 323, 325, 327, 354, 360).

[GL06b] V. Gruhn and R. Laue. "Complexity Metrics for Business Process Models". In: *9th international conference on business information systems.* Ed. by W. Abramowicz and H. C. Mayer. Klagenfurt, Austria, 2006, pp. 1–12 (cit. on pp. xxv, xxvii–xxx, 2, 24, 25, 136, 137, 142, 257, 262, 264–266, 272, 322, 325, 327, 344, 345, 349, 360, 365–367, 373).

[GL07] V. Gruhn and R. Laue. "Approaches for Business Process Model Complexity Metrics". In: *Technologies for business information systems.* Springer Netherlands, 2007, pp. 13–24. DOI: 10.1007/1-4020-5634-6_2 (cit. on pp. 24, 325).

[GD05] A. S. Guceglioglu and O. Demirors. "Using Software Quality Characteristics to Measure Business Process Quality". In: *Business Process Management: 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005. Proceedings.* Ed. by W. M. P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 374–379. DOI: 10.1007/11538394_26 (cit. on p. 24).

[Gui05] G. Guizzardi. "Ontological Foundations for Structural Conceptual Models". PhD thesis. Enschede: University of Twente, 2005 (cit. on pp. 14, 15, 18).

[Hab+08] D. Habich, S. Richly, S. Preissler, M. Grasselt, W. Lehner, and A. Maier. "BPEL^{DT} Data-Aware Extension for Data-Intensive Service Applications". In: *Emerging Web Services Technology, Volume II.* Ed. by T. Gschwind and C. Pautasso. Basel: Birkhäuser Basel, 2008. Chap. BPEL DT Data-Aware Extension for Data-Intensive Service Applications, pp. 111–128. DOI: 10.1007/978-3-7643-8864-5_8 (cit. on p. 22).

[Hai+13] C. Haisjackl, S. Zugal, P. Soffer, I. Hadar, M. Reichert, J. Pinggera, and B. Weber. "Making Sense of Declarative Process Models: Common Strategies and Typical Pitfalls". In: *Proceedings Enterprise, Business-Process and Information Systems Modeling: 14th International Conference, BPMDS 2013, 18th International Conference, EMMSAD 2013, Held at CAiSE 2013, Valencia, Spain, June 17-18, 2013.* Ed. by S. Nurcan, H. A. Proper, P. Soffer, J. Krogstie, R. Schmidt, T. Halpin, and I. Bider. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 2–17. DOI: 10.1007/978-3-642-38484-4_2 (cit. on pp. 1, 5, 20).

[Hai+16] C. Haisjackl, I. Barba, S. Zugal, P. Soffer, I. Hadar, M. Reichert, J. Pinggera, and B. Weber. "Understanding Declare Models: Strategies, Pitfalls, Empirical Results". In: *Software & Systems Modeling* 15.2 (2016), pp. 325–352. DOI: 10.1007/s10270-014-0435-z (cit. on p. 20).

[Hal87] M. H. Halstead. *Elements of Software Science.* New York, NY, USA: Elsevier Science Inc., 1987 (cit. on pp. 24, 321, 336).

[HZ09] Zh. Han and L. Zhang. "Evaluating Cognitive Complexity Measure of Processes with Weyuker Properties". In: *Proceedings of the 2nd International Conference on Advanced Computer Theory and Engineering (ICACTE 2009).* Cairo, Egypt, 2009 (cit. on pp. 124, 325, 360).

[Har87] D. Harel. "Statecharts: a Visual Formalism for Complex Systems". In: *Science of Computer Programming* 8.3 (1987), pp. 231–274. DOI: 10.1016/0167-6423(87)90035-9 (cit. on pp. 88, 89).

[Hau+15] M. Hauder, R. Kazman, and F. Matthes. "Empowering End-Users to Collaboratively Structure Processes for Knowledge Work". In: *Proceedings of the 18^{th} International Conference on Business Information Systems (BIS).* 2015 (cit. on pp. 5, 197).

[Hei+15] B. Heinrich, M. Klier, and S. Zimmermann. "Automated Planning of Process Models: Design of a Novel Approach to Construct Exclusive Choices". In: *Decision Support Systems* 78 (2015), pp. 1–14. DOI: `10.1016/j.dss.2015.07.005` (cit. on p. 24).

[HL07] J. Heiser and R. J. D. Lotto. *Introduction to Investigative Case Management Products.* Tech. rep. April. Stamford, CT: Gartner, 2007, p. 6 (cit. on p. 28).

[HB09] M. Held and W. Blochinger. "Structured Collaborative Workflow Design". In: *Future Generation Computer Systems* 25.6 (2009), pp. 638–653. DOI: `10.1016/j.future.2008.12.005` (cit. on pp. xxi, xxiii, xxv, xxvi, xxviii, xxxi, xxxv, 124, 137, 252, 255, 259, 260, 265–267, 269, 285, 287, 341, 342, 356, 358, 362, 363, 367, 373, 385).

[Hen12] B. Henderson-Sellers. *On the Mathematics of Modelling, Metamodelling, Ontologies and Modelling Languages.* SpringerBriefs in Computer Science. Springer Berlin Heidelberg, 2012. DOI: `10.1007/978-3-642-29825-7` (cit. on pp. 14, 15, 18).

[HK81] S. Henry and D. Kafura. "Software Structure Metrics Based on Information Flow". In: *IEEE Transactions on Software Engineering* 7.5 (1981), pp. 510–518. DOI: `10.1109/TSE.1981.231113` (cit. on pp. 323, 327).

[Her14] V. C. T. Hermans. "A Hybrid Process Modeling Approach". Master. Eindhoven University of Technology, 2014, p. 85 (cit. on p. 21).

[Her+12] E. Hernandes, A. Zamboni, S. Fabbri, and A. D. Thommazo. "Using GQM and TAM to Evaluate StArt - A Tool that Supports Systematic Review". In: *CLEI Electronic Journal* 15 (Apr. 2012), pp. 3–3 (cit. on pp. 104, 108).

[Hir05] J. E. Hirsch. "An Index to Quantify an Individual's Scientific Research Output". In: *Proceedings of the National Academy of Sciences of the United States of America* 102.46 (2005), pp. 16569–16572. DOI: `10.1073/pnas.0507655102`. arXiv: `0508025v5` (cit. on p. 354).

[Hop+01] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* 2nd ed. Addison-Wesley, 2001, p. 537 (cit. on p. 261).

[Hul08] R. Hull. “Artifact-Centric Business Process Models: Brief Survey of Research Results and Challenges”. In: *On the Move to Meaningful Internet Systems: OTM 2008.* Ed. by R. Meersman and Z. Tari. Vol. 5332. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 1152–1163. DOI: 10.1007/978-3-540-88873-4_17 (cit. on pp. 29, 40, 41, 198, 253).

[Hul+99] R. Hull, F. Llirbat, E. Siman, J. Su, G. Dong, Bh. Kumar, and G. Zhou. “Declarative Workflows That Support Easy Modification and Dynamic Browsing”. In: *SIGSOFT Software Engineering Notes* 24.2 (1999), pp. 69–78. DOI: 10.1145/295666.295674 (cit. on pp. 35, 36).

[Hul+11a] R. Hull, E. Damaggio, R. de Masellis, F. Fournier, M. Gupta, F. T. Heath III, S. Hobson, M. Linehan, S. Maradugu, A. Nigam, P. N. Sukaviriya, R. Vaculín, and R. de Masellis. “Business Artifacts with Guard-Stage-Milestone Lifecycles: Managing Artifact Interactions with Conditions and Events”. In: *Proceedings of the 5th ACM international conference on Distributed event-based system.* DEBS ’11. New York, NY, USA: ACM, 2011, pp. 51–62. DOI: 10.1145/2002259.2002270 (cit. on pp. 3, 36, 37, 40, 41, 44–46, 48, 195, 280).

[Hul+11b] R. Hull, E. Damaggio, F. Fournier, M. Gupta, F. T. Heath III, S. Hobson, M. Linehan, S. Maradugu, A. Nigam, P. Sukaviriya, and R. Vaculín. “Introducing the Guard-Stage-Milestone Approach for Specifying Business Entity Lifecycles”. In: *Web Services and Formal Methods.* Ed. by M. Bravetti and T. Bultan. Vol. 6551. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 1–24. DOI: 10.1007/978-3-642-19589-1_1 (cit. on pp. 2, 3, 21, 29, 36, 37, 39–41, 48, 62, 77, 193, 254, 262, 392).

[Hul+16] R. Hull, V. S. Batra, Y.-M. Chen, A. Deutsch, F. T. Heath III, and V. Vianu. “Towards a Shared Ledger Business Collaboration Language Based on Data-Aware Processes”. In: *Service-Oriented Computing: 14th International Conference, ICSOC 2016, Banff, AB, Canada, October 10-13, 2016, Proceedings.* Ed. by Q. Z. Sheng, E. Stroulia, S. Tata, and S. Bhiri. Springer International Publishing, 2016, pp. 18–36. DOI: 10.1007/978-3-319-46295-0_2 (cit. on p. 254).

[I+13] Ö. Işik, W. Mertens, and J. van den Bergh. “Practices of Knowledge Intensive Process Management: Quantitative Insights”. In: *Business Process Management Journal* 19.3 (2013), pp. 515–534. DOI: 10.1108/14637151311319932 (cit. on p. 75).

[IEE90] IEEE. *IEEE Standard Glossary of Software Engineering Terminology.* Std 610.12. Vol. 121990. New York (NY): IEEE, 1990, p. 84 (cit. on pp. 26, 320).

[IEE10] IEEE. *ISO/IEC/IEEE 24765:2010(E) Systems and Software Engineering – Vocabulary*. 2010. DOI: 10.1109/IEEESTD.2010.5733835 (cit. on p. 20).

[Ind+09a] M. Indulska, P. Green, J. Recker, and M. Rosemann. "Business Process Modeling: Perceived Benefits". In: *ProceedingsConceptual Modeling - ER 2009: 28th International Conference on Conceptual Modeling, Gramado, Brazil, November 9-12, 2009*. Ed. by A. H. F. Laender, S. Castano, U. Dayal, F. Casati, and J. P. M. Oliveira. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 458–471. DOI: 10.1007/978-3-642-04840-1_34 (cit. on p. 2).

[Ind+09b] M. Indulska, M. zur Muehlen, and J. C. Recker. *Measuring Method Complexity: The Case of the Business Process Modeling Notation*. BPM Center Report BPM-09-03. Business Process Management Center, 2009, p. 14 (cit. on pp. 5, 26, 65, 66, 68–71, 73–75, 194, 197).

[Jai+16] H. Jain, P. Radha Krishna, and K. Karlapalem. "Context-Aware Workflow Execution Engine for E-Contract Enactment". In: *Conceptual Modeling: 35th International Conference, ER 2016, Gifu, Japan, November 14-17, 2016, Proceedings*. Ed. by I. Comyn-Wattiau, K. Tanaka, I.-Y. Song, S. Yamamoto, and M. Saeki. Springer International Publishing, 2016, pp. 293–301. DOI: 10.1007/978-3-319-46397-1_23 (cit. on p. 24).

[Jan15] P. Jansen. "Case Management Model and Notation, Analysis of a Practical Application". MA thesis. Open Universiteit, faculteit Management, Science & Technology, 2015 (cit. on p. 78).

[Kaa+06] K. Kaan, H. A. Reijers, and P. V. D. Molen. "Introducing Case Management: Opening Workflow Management's Black Box". In: *BPM 2006, LNCS*. Ed. by S. Dustdar, J.L. Fiadeiro, and A. Sheth. Vol. 4102. 2006, pp. 358–367 (cit. on p. 27).

[Kai74] R. A. Kaimann. "Coefficient of Network Complexity". In: *Management Science* 21.2 (1974), pp. 172–177. DOI: 10.1287/mnsc.21.2.172 (cit. on p. 310).

[KB04] C. Kaner and W. P. Bond. "Software Engineering Metrics: What Do They Measure and How Do We Know?" In: *10th International Software Metrics Symposium, METRICS 2004*. 2004 (cit. on pp. 107, 124).

[Kan+16] U. Kannengiesser, M. Neubauer, and R. Heininger. "Integrating Business Processes and Manufacturing Operations Based on S-BPM and B2MML". In: *Proceedings of the 8th International Conference on Subject-oriented Business Process Management*. S-BPM '16. Erlangen, Germany: ACM, 2016, pp. 121–1210. DOI: 10.1145/2882879.2882881 (cit. on p. 23).

[Kap+13] M. Kapuruge, J. Han, and A. Colman. "Supporting Adaptation Patterns in the Event-Driven Business Process Modeling Paradigm". In: *Web Information Systems Engineering – WISE 2013: 14th International Conference, Nanjing, China, October 13-15, 2013, Proceedings, Part II.* Ed. by X. Lin, Y. Manolopoulos, D. Srivastava, and G. Huang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 299–308. DOI: 10.1007/978-3-642-41154-0_22 (cit. on p. 24).

[Kel+92] G. Keller, M. Nüttgens, and A. W. Scheer. *Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK).* German. Tech. rep. 89. Saarbrücken, Germany: Universität des Saarlandes, Germany, 1992 (cit. on p. 67).

[Kem16] S. Kemsley. *Take Mike Marin's CMMN Survey: Learn Something and Help CMMN Research.* Accessed: October 06, 2016. 2016. URL: https://column2.com/2016/07/take-mike-marins-cmmn-survey-learn-something-and-help-cmmn-research/ (cit. on pp. 178, 391).

[Ker08] M. Kerremans. *Case Management Is a Challenging BPMS Use Case.* Gartner report G00162739. Stamford, CT: Gartner, 2008, p. 7 (cit. on pp. 28, 255, 256).

[Khl+09] W. Khlif, L. Makni, N. Zaaboub, and H. Ben-Abdallah. "Quality Metrics for Business Process Modeling". In: *Proceedings of the 9th WSEAS International Conference on Applied Computer Science.* ACS'09. Genova, Italy: World Scientific, Engineering Academy, and Society (WSEAS), 2009, pp. 195–200 (cit. on pp. 24, 25).

[Kho+09] A. Khoshkbarforoushha, P. Jamshidi, A. Nikravesh, S. Khoshnevis, and F. Shams. "A Metric for Measuring BPEL Process Context-independency". In: *2009 IEEE International Conference on Service-Oriented Computing and Applications (SOCA).* 2009, pp. 1–8. DOI: 10.1109/SOCA.2009.5410260 (cit. on pp. xxxiii, 124, 281, 339, 340, 382).

[Kit04] B. Kitchenham. *Procedures for Performing Systematic Reviews.* Tech. rep. Technical Report TR/SE-0401. Keele University, 2004 (cit. on pp. 103, 132, 133, 198).

[KC07] B. Kitchenham and S. Charters. *Guidelines for Performing Systematic Literature Reviews in Software Engineering.* Technical Report EBSE-2007-01. Software Eng. Group, Keele Univ. and Dept. of Computer Science, Univ. of Durham, UK, 2007 (cit. on pp. 103, 108, 132, 133, 198, 286).

[Kit+04] B. A. Kitchenham, T. Dyba, and M. Jorgensen. "Evidence-based Software Engineering". In: *Proceedings. 26th International Conference on Software Engineering.* May 2004, pp. 273–281. DOI: 10.1109/ICSE.2004.1317449 (cit. on pp. 103, 132, 133, 198).

[Klu15] K. Kluza. "Measuring Complexity of Business Process Models Integrated with Rules". In: *Artificial Intelligence and Soft Computing: 14th International Conference, ICAISC 2015, Zakopane, Poland, June 14-18, 2015, Proceedings, Part II.* Ed. by L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, A. L. Zadeh, and M. J. Zurada. Springer International Publishing, 2015, pp. 649–659. DOI: 10.1007/978-3-319-19369-4_57 (cit. on pp. xxiv, 3, 24, 106, 258, 356, 361).

[KN12] K. Kluza and G. J. Nalepa. "Proposal of Square Metrics for Measuring Business Process Model Complexity". In: *Proceedings of the Federated Conference on Computer Science and Information Systems.* Ed. by IEEE. 2012, pp. 919–922 (cit. on pp. xxvi, xxxiv, 25, 124, 136, 137, 261, 280, 353, 354, 364, 381).

[Klu+14] K. Kluza, G. J. Nalepa, and J. Lisiecki. "Square Complexity Metrics for Business Process Models". In: *Advances in Business ICT.* Ed. by M. Mach-Król and T. Pełech-Pilichowski. Springer International Publishing, 2014, pp. 89–107. DOI: 10.1007/978-3-319-03677-9_6 (cit. on pp. 125, 354, 364, 381).

[Ko09] R. K. L. Ko. "A Computer Scientist's Introductory Guide to Business Process Management (BPM)". In: *Crossroads* 15.4 (2009), pp. 11–18. DOI: 10.1145/1558897.1558901 (cit. on pp. 2, 12).

[Kor08] B. Korherr. "Business Process Modelling - Languages, Goals, and Variabilities". PhD. Vienna University of Technology, 2008, p. 127 (cit. on pp. 16, 74).

[Kre+08] M. Kreimeyer, M. Gurtler, and U. Lindemann. "Structural Metrics for Decision Points within Multiple-domain Matrices Representing Design Processes". In: *2008 IEEE International Conference on Industrial Engineering and Engineering Management.* 2008, pp. 435–439. DOI: 10.1109/IEEM.2008.4737906 (cit. on p. 335).

[Kre+10] M. Kreimeyer, N. Bradford, and U. Lindemann. "Process Analysis using Structural Metrics: A Comprehensive Case Study". In: *Proceedings of DESIGN 2010, the 11th International Design Conference, Dubrovnik, Croatia.* 2010, pp. 295–306 (cit. on p. 348).

[Kre10] M. F. Kreimeyer. “A Structural Measurement System for Engineering Design Processes”. PhD thesis. Technische Universität München, 2010 (cit. on pp. xxii, xxvii, xxix–xxxi, xxxiii, xxxiv, xxxviii, 137, 253, 262, 268, 270–272, 275, 279, 284, 288, 347, 348, 360, 365, 367, 368, 370, 372, 373, 380, 384, 386).

[Kru+14] J. Krumeich, B. Weis, D. Werth, and P. Loos. “Event-Driven Business Process Management: Where are We Now?: A Comprehensive Synthesis and Analysis of Literature”. In: *Business Process Management Journal* 20.4 (2014), pp. 615–633. DOI: 10.1108/BPMJ-07-2013-0092 (cit. on pp. 24, 257).

[Kru+15] J. Krumeich, N. Mehdiyev, D. Werth, and P. Loos. “Towards an Extended Metamodel of Event-Driven Process Chains to Model Complex Event Patterns”. In: *Advances in Conceptual Modeling: ER 2015 Workshops AHA, CMS, EMoV, MoBID, MORE-BI, MReBA, QMMQ, and SCME, Stockholm, Sweden, October 19-22, 2015, Proceedings.* Ed. by M. A. Jeusfeld and K. Karlapalem. Springer International Publishing, 2015, pp. 119–130. DOI: 10.1007/978-3-319-25747-1_12 (cit. on p. 24).

[KS11] E. Kucukoguz and J. Su. “On Lifecycle Constraints of Artifact-Centric Workflows”. In: *Web Services and Formal Methods: 7th International Workshop, WS-FM 2010, Hoboken, NJ, USA, September 16-17, 2010. Revised Selected Papers.* Ed. by M. Bravetti and T. Bultan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 71–85. DOI: 10.1007/978-3-642-19589-1_5 (cit. on p. 60).

[Kum+03] S. Kumaran, P. Nandi, F. T. Heath III, J. Chung, K. Bhaskaran, and R. Das. “ADoc-Oriented Programming”. In: *Proceedings. 2003 Symposium on Applications and the Internet, 2003.* 2003, pp. 334–341. DOI: 10.1109/SAINT.2003.1183067 (cit. on pp. 1, 29–32, 39, 40, 251).

[Kum+07] S. Kumaran, P. Bishop, T. Chao, P. Dhoolia, P. Jain, R. Jaluka, H. Ludwig, A. Moyer, and A. Nigam. “Using a Model-driven Transformational Approach and Service-oriented Architecture for Service Delivery Management”. In: *IBM Systems Journal* 46.3 (2007), pp. 513–529. DOI: 10.1147/sj.463.0513 (cit. on pp. 34, 265).

[Kum+08] S. Kumaran, R. Liu, and F. Y. Wu. “On the Duality of Information-Centric and Activity-Centric Models of Business Processes”. In: *Advanced Information Systems Engineering: 20th International Conference, CAiSE 2008 Montpellier, France, June 16-20, 2008 Proceedings.* Ed. by Z. Bellahsène and M. Léonard. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 32–47. DOI: 10.1007/978-3-540-69534-9_3 (cit. on pp. 32–34, 254).

[Kun+15] J. Kunchala, J. Yu, and S. Yongchareon. “A Survey on Approaches to Modeling Artifact-Centric Business Processes”. In: *Web Information Systems Engineering – WISE 2014 Workshops.* Vol. 9051. Lecture Notes in Computer Science. Springer International Publishing, 2015, pp. 117–132. DOI: 10.1007/978-3-319-20370-6 (cit. on pp. 2, 40, 41, 60).

[KR11a] V. Künzle and M. Reichert. “A Modeling Paradigm for Integrating Processes and Data at the Micro Level”. In: *Enterprise, Business-Process and Information Systems Modeling: 12th International Conference, BPMDS 2011, and 16th International Conference, EMMSAD 2011, held at CAiSE 2011, London, UK, June 20-21, 2011. Proceedings.* Ed. by T. Halpin, S. Nurcan, J. Krogstie, P. Soffer, E. Proper, R. Schmidt, and I. Bider. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 201–215. DOI: 10.1007/978-3-642-21759-3_15 (cit. on p. 34).

[KR11b] V. Künzle and M. Reichert. “PHILharmonicFlows: Towards a Framework for Object-aware Process Management”. In: *Journal of Software Maintenance and Evolution: Research and Practice* 23.4 (2011), pp. 205–244. DOI: 10.1002/smr.524 (cit. on p. 34).

[KR12] V. Künzle and M. Reichert. “Striving for Object-Aware Process Support: How Existing Approaches Fit Together”. In: *Data-Driven Process Discovery and Analysis: First International Symposium, SIMPDA 2011, Campione d’Italia, Italy, June 29 – July 1, 2011, Revised Selected Papers.* Ed. by K. Aberer, E. Damiani, and T. Dillon. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 169–188. DOI: 10.1007/978-3-642-34044-4_10 (cit. on p. 34).

[Kur+15] M. Kurz, W. Schmidt, A. Fleischmann, and M. Lederer. “Leveraging CMMN for ACM: Examining the Applicability of a New OMG Standard for Adaptive Case Management”. In: *Proceedings of the 7th International Conference on Subject-Oriented Business Process Management.* S-BPM ONE ’15. Kiel, Germany: ACM, 2015. DOI: 10.1145/2723839.2723843 (cit. on pp. 5, 197).

[Küs+14] T. Küster, A. Heßler, and S. Albayrak. “Towards Process-Oriented Modelling and Creation of Multi-Agent Systems”. In: *Engineering Multi-Agent Systems: Second International Workshop, EMAS 2014, Paris, France, May 5-6, 2014, Revised Selected Papers.* Ed. by F. Dalpiaz, J. Dix, and M. B. van Riemsdijk. Springer International Publishing, 2014, pp. 163–180. DOI: 10.1007/978-3-319-14484-9_9 (cit. on p. 23).

[La +11a] M. La Rosa, M. Dumas, A. H. M. Ter Hofstede, and J. Mendling. “Configurable Multi-perspective Business Process Models”. In: *Information*

Systems 36.2 (2011), pp. 313–340. DOI: 10.1016/j.is.2010.07.001 (cit. on p. 19).

[La +11b] M. La Rosa, P. Wohed, J. Mendling, A. H. M. Ter Hofstede, H. A. Reijers, and W. M. P. van der Aalst. "Managing Process Model Complexity Via Abstract Syntax Modifications". In: *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS* 7.4 (2011), pp. 614–629. DOI: 10.1109/TII.2011.2166795 (cit. on pp. xxi, xxv, xxviii, xxxv, 137, 253, 260, 261, 265, 285, 351, 359, 364, 366, 373, 386).

[La +11c] M. La Rosa, A. H. M. Ter Hofstede, P. Wohed, H. A. Reijers, J. Mendling, and W. M. P. van der Aalst. "Managing Process Model Complexity via Concrete Syntax Modifications". In: *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS* 7.2 (2011), pp. 255–265. DOI: 10.1109/TII.2011.2124467 (cit. on p. 351).

[Lv09] K. B. Lassen and W. M. P. van der Aalst. "Complexity Metrics for Workflow Nets". In: *Information and Software Technology* 51.3 (2009), pp. 610–626. DOI: 10.1016/j.infsof.2008.08.005 (cit. on pp. xxvi, xxxv, 2, 124, 159, 252, 261, 262, 264, 286, 337, 338, 341, 358, 364, 386).

[Lat01] A. M. Latva-Koivisto. *Finding a Complexity Measure for Business Process Models*. Tech. rep. Helsinki University of Technology, Systems Analysis Laboratory, 2001, pp. 1–26 (cit. on pp. xxiv, xxxiv, xxxvi, 25, 124, 256, 257, 259, 264, 265, 279, 284, 309, 310, 312, 321, 329, 342, 346, 349, 356, 360, 363, 380, 384).

[LG11] R. Laue and A. Gadatsch. "Measuring the Understandability of Business Process Models - Are We Asking the Right Questions?" In: *Business Process Management Workshops*. Ed. by M. zur Muehlen and J. Su. Vol. 66. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2011, pp. 37–48. DOI: 10.1007/978-3-642-20511-8_4 (cit. on p. 164).

[LM10] R. Laue and J. Mendling. "Structuredness and Its Significance for Correctness of Process Models". In: *Information Systems and e-Business Management* 8.3 (2010), pp. 287–307. DOI: 10.1007/s10257-009-0120-x (cit. on p. 351).

[Le +09] C. Le Clair, C. Moore, and R. Vitti. *Dynamic Case Management - An Old Idea Catches New Fire*. Tech. rep. Cambridge, MA: Forrester, 2009, p. 18 (cit. on pp. 28, 67, 75, 261).

[Lea09] N. Leavitt. "Complex-Event Processing Poised for Growth". In: *Computer* 42.4 (2009), pp. 17–20. DOI: 10.1109/MC.2009.109 (cit. on p. 24).

[Lim16] LimeSurvey Development Team. *LimeSurvey version 2.06 LTS*. 2016 (cit. on pp. 179, 395).

[Lin+03] A. Lindsay, D. Downs, and K. Lunn. “Business Processes - Attempts to Find a Definition”. In: *Information and Software Technology* 45.15 (2003), pp. 1015–1019. DOI: `10.1016/S0950-5849(03)00129-0` (cit. on p. 254).

[Liu+07] R. Liu, K. Bhattacharya, and F. Y. Wu. “Modeling Business Contexture and Behavior Using Business Artifacts”. In: *Advanced Information Systems Engineering: 19th International Conference, CAiSE 2007, Trondheim, Norway, June 11-15, 2007. Proceedings*. Ed. by J. Krogstie, A. Opdahl, and G. Sindre. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 324–339. DOI: `10.1007/978-3-540-72988-4_23` (cit. on p. 32).

[LN12] N. Lohmann and M. Nyolt. “Artifact-Centric Modeling Using BPMN”. In: *Service-Oriented Computing - ICSOC 2011 Workshops: ICSOC 2011, International Workshops WESOA, NFPSLAM-SOC, and Satellite Events, Paphos, Cyprus, December 5-8, 2011. Revised Selected Papers*. Ed. by G. Pallis, M. Jmaiel, A. Charfi, S. Graupner, Y. Karabulut, S. Guinea, F. Rosenberg, Q. Z. Sheng, C. Pautasso, and S. Ben Mokhtar. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 54–65. DOI: `10.1007/978-3-642-31875-7_7` (cit. on p. 60).

[LÏ5] D. Lübke. “Using Metric Time-lines for Identifying Architecture Shortcomings in Process Execution Architectures”. In: *Proceedings of the Second International Workshop on Software Architecture and Metrics*. SAM ’15. Florence, Italy: IEEE Press, 2015, pp. 55–58 (cit. on pp. xxii, xxvi, xxxvii, 137, 251, 253, 262, 286, 287, 355, 356, 358, 360, 364, 365, 380, 385).

[Mäe11] R. Mäesalu. “Complexity and Understandability Comparison between Unstructured and Structured Business Process Models”. MA thesis. University of Tartu, 2011 (cit. on pp. 124, 125, 351, 359, 362, 364, 367).

[Mag+13] F. M. Maggi, M. Dumas, L. García-Bañuelos, and M. Montali. “Discovering Data-Aware Declarative Process Models from Event Logs”. In: *Business Process Management: 11th International Conference, BPM 2013, Beijing, China, August 26-30, 2013. Proceedings*. Ed. by F. Daniel, J. Wang, and B. Weber. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 81–96. DOI: `10.1007/978-3-642-40176-3_8` (cit. on p. 1).

[MM15] M. Malinova and J. Mendling. “Why is BPMN not Appropriate for Process Maps?” In: *Proceedings of the International Conference on Information Systems - Exploring the Information Frontier, ICIS 2015, Fort Worth, Texas, USA, December 13-16, 2015*. 2015 (cit. on p. 11).

[Mao10a] C. Mao. “Complexity Analysis for Petri Net-Based Business Process in Web Service Composition”. In: *Service Oriented System Engineering (SOSE), 2010 Fifth IEEE International Symposium on*. 2010, pp. 193–196. DOI: 10.1109/SOSE.2010.24 (cit. on pp. xxi, xxiii, xxxii, xxxiii, xxxvii, 252, 253, 265, 276, 277, 279, 287, 346, 347, 357, 359, 363, 377, 378, 380, 386).

[Mao10b] Ch. Mao. “Control Flow Complexity Metrics for Petri Net-based Web Service Composition”. In: *Journal of Software* 5 (11 2010), pp. 1292–1299. DOI: 10.4304/jsw.5.11.1292-1299 (cit. on pp. xxi, 253, 347, 359, 363, 377, 378, 380, 386).

[Mar+14a] A. Margara, G. Cugola, and G. Tamburrelli. “Learning from the Past: Automated Rule Generation for Complex Event Processing”. In: *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*. DEBS ’14. Mumbai, India: ACM, 2014, pp. 47–58. DOI: 10.1145/2611286.2611289 (cit. on p. 257).

[Mar16a] M. A. Marin. “Introduction to the Case Management Model and Notation (CMMN)”. In: *Computing Research Repository* (2016). arXiv: abs/1608.05011 (cit. on p. 8).

[Mar16b] M. A. Marin. *Looking at the Case Management Modeling and Notation (CMMN) from a Research Perspective*. Accessed: October 06, 2016. 2016. URL: http://www.bptrends.com/looking-at-the-case-management-modeling-and-notation-cmmn-from-a-research-perspective/ (cit. on pp. 178, 391).

[Mar16c] M. A. Marin. *The Case Management Model and Notation (CMMN) version 1.0 Tutorial*. Accessed: August 08, 2016. 2016. URL: http://cmmn.byethost4.com (cit. on p. 8).

[MB15] M. A. Marin and J. A. Brown. “Implementing a Case Management Modeling and Notation (CMMN) System using a Content Management Interoperability Services (CMIS) compliant repository”. In: *Computing Research Repository* (2015). arXiv: abs/1504.06778 (cit. on pp. 9, 48).

[MH15] M. A. Marin and M. Hauder. “Case Management: A Data Set of Definitions”. In: *Computing Research Repository* (2015). arXiv: abs/1507.04004 (cit. on p. 8).

[Mar+13] M. A. Marin, R. Hull, and R. Vaculín. “Data Centric BPM and the Emerging Case Management Standard: A Short Survey”. In: *Business Process Management Workshops*. Ed. by M. La Rosa and P. Soffer. Vol. 132. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2013, pp. 24–30. DOI: 10.1007/978-3-642-36285-9_4 (cit. on pp. 1, 3, 8, 11, 37, 38, 48, 66, 77).

[Mar+14b] M. A. Marin, H. Lotriet, and J. A. van der Poll. "Measuring Method Complexity of the Case Management Modeling and Notation (CMMN)". In: *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference (SAICSIT'14)*. Ed. by J. P. van Deventer, M. C. Matthee, H. Gelderblom, and A. Gerber. Centurion, Gauteng, South Africa: ACM Press, New York, USA, 2014, pp. 209–216 (cit. on pp. 8, 11, 38, 65, 75).

[Mar+15a] M. A. Marin, M. Hauder, and F. Matthes. "Case Management: An Evaluation of Existing Approaches for Knowledge-Intensive Processes". In: *4th International Workshop on Adaptive Case Management and other non-workflow approaches to BPM (AdaptiveCM)*. Innsbruck, Austria, 2015, pp. 1–12 (cit. on pp. 6, 11, 29, 38).

[Mar+15b] M. A. Marin, H. Lotriet, and J. A. van der Poll. "Metrics for the Case Management Modeling and Notation (CMMN) Specification". In: *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference (SAICSIT'15)*. Ed. by R. J. Barnett, L. Cleophas, D. G. Kourie, D. B. le Roux, and B. W. Watson. Stellenbosch, South Africa: ACM Press, New York, USA, 2015 (cit. on pp. 6, 39, 63, 135, 156, 174).

[Mar+16] M. A. Marin, H. Lotriet, and J. A. van der Poll. "Implementing Deterministic Finite State Machines using Guard-Stage-Milestone". In: *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference (SAICSIT'16)*. Johannesburg, South Africa: ACM Press, New York, USA, 2016 (cit. on pp. 6, 39, 63, 77, 102, 392).

[MF11] O. Marjanovic and R. Freeze. "Knowledge Intensive Business Processes: Theoretical Foundations and Research Challenges". In: *2011 44th Hawaii International Conference on System Sciences (HICSS)*. 2011, pp. 1–10. DOI: 10.1109/HICSS.2011.271 (cit. on pp. 20, 23, 264).

[Mar+15c] A. Marrella, M. Mecella, A. Russo, S. Steinau, K. Andrews, and M. Reichert. "Data in Business Process Models, A Preliminary Empirical Study (Short Paper)". In: *2015 IEEE 8th International Conference on Service-Oriented Computing and Applications (SOCA)*. 2015, pp. 116–122. DOI: 10.1109/SOCA.2015.19 (cit. on pp. 22, 36).

[Mar+14c] Ch. Marshall, P. Brereton, and B. Kitchenham. "Tools to Support Systematic Reviews in Software Engineering: A Feature Analysis". In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software*

Engineering. EASE '14. London, England, United Kingdom: ACM, 2014, pp. 131–1310. DOI: 10.1145/2601248.2601270 (cit. on pp. 104, 108).

[Mar+15d] R. Martinho, R. Rijo, and A. Nunes. "Complexity Analysis of a Business Process Automation: Case Study on a Healthcare Organization". In: *Procedia Computer Science* 64 (2015), pp. 1226–1231. DOI: 10.1016/j.procs.2015.08.510 (cit. on p. 355).

[May+92] R. J. Mayer, M. K. Painter, and P. S. DeWitte. *IDEF Family of Methods for Concurrent Engineering and Business Re-engineering Applications*. Tech. rep. College Station, TX.: Knowledge Based Systems, Inc., 1992 (cit. on pp. 29, 263).

[MH11] S. Mazanek and M. Hanus. "Constructing a Bidirectional Transformation between BPMN and BPEL with a Functional Logic Programming Language". In: *Journal of Visual Languages & Computing* 22.1 (2011), pp. 66–89. DOI: 10.1016/j.jvlc.2010.11.005 (cit. on p. 78).

[McC76] T. J. McCabe. "A Complexity Measure". In: *IEEE Transactions on Software Engineering* SE-2.4 (1976), pp. 308–320. DOI: 10.1109/TSE.1976.233837 (cit. on pp. 24, 336, 338).

[McC10] D. McCauley. "Achieving Agility". In: *Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done*. Ed. by K. D. Swenson. 1st ed. Tampa, Florida, USA: Meghan-Kiffer Press, 2010. Chap. Achieving Agility, pp. 257–275 (cit. on p. 28).

[Mej+15] A. Mejri, S. A. Ghanouchi, and R. Martinho. "Evaluation of Process Modeling Paradigms Enabling Flexibility". In: *Conference on ENTERprise Information Systems/International Conference on Project MANagement/Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN / HCist 2015 October 7-9, 2015*. Vol. 64. 2015, pp. 1043–1050. DOI: 10.1016/j.procs.2015.08.514 (cit. on p. 20).

[Mel+10] J. Melcher, J. Mendling, H. A. Reijers, and D. Seese. "On Measuring the Understandability of Process Models". In: *Lecture Notes in Business Information Processing* 43 LNBIP (2010), pp. 465–474. DOI: 10.1007/978-3-642-12186-9_44 (cit. on p. 164).

[Men06] J. Mendling. *Testing Density as a Complexity Metric for EPCs*. Tech. rep. Wien, Austria: Vienna University of Economics and Business Administration, 2006, p. 7 (cit. on pp. xxv, xxvi, 25, 125, 260, 324, 351, 363).

[Men07] J. Mendling. “Detection and Prediction of Errors in EPC Business Process Models”. PhD. Vienna University of Economics and Business Administration, 2007, p. 491 (cit. on pp. xxiv–xxviii, xxxiii, xxxv, xxxvii, xxxviii, 25, 26, 137, 252, 253, 257–261, 264, 267–272, 275, 277, 279, 284, 286, 326–329, 332, 334, 350, 351, 359–361, 363, 364, 366–368, 370–373, 376, 379, 381, 385, 386).

[Men08] J. Mendling. *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness.* Springer Publishing Company, Incorporated, 2008 (cit. on pp. 13, 67, 327).

[MN07] J. Mendling and G. Neumann. “Error Metrics for Business Process Models”. In: *19th International Conference on Advanced Information Systems Engineering.* Trondheim, Norway, 2007, pp. 53–56 (cit. on pp. 327, 363, 367, 385, 386).

[MS08] J. Mendling and M. Strembeck. “Influence Factors of Understanding Business Process Models”. In: *11th International Conference on Business Information Systems.* Ed. by W. Abramowicz and D. Fensel. Innsbruck, Austria: Springer Berlin Heidelberg, 2008, pp. 142–153 (cit. on pp. 125, 160, 332, 361, 363, 364, 367, 385, 386).

[Men+06] J. Mendling, M. Moser, G. Neumann, B. F. van Dongen, and W. M. P. van der Aalst. *A Quantitative Analysis of Faulty EPCs in the SAP Reference Model.* Tech. rep. Eindhoven, The Netherlands: Eindhoven University of Technology, 2006, p. 48 (cit. on pp. xxi, xxii, xxv–xxvii, xxxiii, xxxv, xxxviii, 259, 264, 265, 267, 268, 270, 272, 274, 275, 277, 279, 280, 285, 323, 327, 328, 348, 351, 362, 365, 368, 371, 373, 375, 376, 379, 381).

[Men+07a] J. Mendling, G. Neumann, and W. M. P. van der Aalst. “On the Correlation between Process Model Metrics and Errors”. In: *26th international conference on Conceptual modeling.* Auckland, New Zealand: Australian Computer Society, Inc., 2007, pp. 173–178 (cit. on pp. 159, 327).

[Men+07b] J. Mendling, G. Neumann, and W. M. P. van der Aalst. “Understanding the Occurrence of Errors in Process Models based on Metrics”. In: *15th International Conference on Cooperative Information Systems.* Ed. by R. Meersman and Z. Tari. Vilamoura, Portugal: Springer, 2007, pp. 113–130 (cit. on p. 327).

[Men+07c] J. Mendling, H. A. Reijers, and J. Cardoso. “What Makes Process Models Understandable?” In: *5th International Conference on Business Process Management.* Ed. by G. Alonso, P. Dadam, and M. Rosemann. Brisbane, Australia: Springer, 2007, pp. 1–16 (cit. on pp. 125, 158, 160, 164, 334, 359–361, 363, 364, 366–368, 370–373, 376, 379, 381, 385, 386).

[MW13] A. Meyer and M. Weske. "Activity-Centric and Artifact-Centric Process Model Roundtrip". In: *Business Process Management Workshops - BPM 2013 International Workshops, Beijing, China, August 26, 2013, Revised Papers.* 2013, pp. 167–181. DOI: 10.1007/978-3-319-06257-0_14 (cit. on p. 78).

[Mey+11] A. Meyer, S. Smirnov, and M. Weske. *Data in Business Processes.* Technical Report 50. Hasso-Plattner-Institut for IT Systems Engineering, Universität Potsdam, 2011 (cit. on p. 22).

[MK07] S. Misra and H. Kilic. "Measurement Theory and Validation Criteria for Software Complexity Measures". In: *SIGSOFT Software Engineering Notes* 32.2 (2007), pp. 1–3. DOI: 10.1145/1234741.1234760 (cit. on p. 144).

[Mis+12] S. Misra, I. Akman, and R. Colomo-Palacios. "Framework for Evaluation and Validation of Software Complexity Measures". In: *Software, IET* 6.4 (2012), pp. 323–334. DOI: 10.1049/iet-sen.2011.0206 (cit. on pp. 155, 158, 159).

[MC16] M. Montali and D. Calvanese. "Soundness of data-aware, Case-centric Processes". In: *International Journal on Software Tools for Technology Transfer* (2016), p. 1. DOI: 10.1007/s10009-016-0417-2 (cit. on p. 23).

[Moo09] D. Moody. "The 'Physics' of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering". In: *IEEE Transactions on Software Engineering* 35.6 (2009), pp. 756–779. DOI: 10.1109/TSE.2009.67 (cit. on p. 15).

[Mor99] S. Morasca. "Measuring Attributes of Concurrent Software Specifications in Petri Nets". In: *Proceedings Sixth International Software Metrics Symposium (Cat. No.PR00403).* 1999, pp. 100–110. DOI: 10.1109/METRIC.1999.809731 (cit. on pp. 25, 124, 352).

[Mor08] S. Morasca. "Refining the Axiomatic Definition of Internal Software Attributes". In: *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement.* ESEM '08. Kaiserslautern, Germany: ACM, 2008, pp. 188–197. DOI: 10.1145/1414004.1414035 (cit. on pp. 25, 124, 352).

[Mor+15] I. Moreno-Montes de Oca, M. Snoeck, H. A. Reijers, and A. Rodríguez-Morffi. "A Systematic Literature Review of Studies on Business Process Modeling Quality". In: *Information and Software Technology* 58 (2015), pp. 187–205. DOI: 10.1016/j.infsof.2014.07.011 (cit. on pp. 104–106).

[MS13] H. R. Motahari-Nezhad and K. D. Swenson. "Adaptive Case Management: Overview and Research Challenges". In: *Proceedings of the 2013 IEEE 15th Conference on Business Informatics*. CBI '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 264–269. DOI: 10.1109/CBI.2013.44 (cit. on pp. 28, 251, 282).

[Muk+10a] G. M. Muketha, A. A. A. Ghani, M. H. Selamat, and R. Atan. "A Survey of Business Process Complexity Metrics". In: *Information Technology Journal* 9.7 (2010), pp. 1336–1344 (cit. on pp. 5, 104–107, 131, 138, 151, 157, 158).

[Muk+10b] G. M. Muketha, A. A. A. Ghani, M. H. Selamat, and R. Atan. "Complexity Metrics for Executable Business Processes". In: *Information Technology Journal* 9.7 (2010), pp. 1317–1326 (cit. on pp. xxi–xxiii, xxvii, xxxi, xxxii, xxxv, 124, 137, 159, 252, 253, 257, 263, 268, 278, 285, 344, 355, 358–360, 365, 373, 380, 385).

[Nan+10] P. Nandi, D. Koenig, S. Moser, R. Hull, V. Klicnik, Sh. Claussen, M. Kloppmann, and J. Vergo. *Data4BPM, Part 1: Introducing Business Entities and the Business Entity Definition Language (BEDL)*. Accessed: July 07, 2016. IBM DeveloperWorks. 2010. URL: http://www.ibm.com/developerworks/websphere/library/techarticles/1004_nandi/1004_nandi.html (cit. on p. 34).

[Net+07] N. Nethercote, P. J. Stuckey, R. Becket, S. Brand, G. Duck, and G. Tack. "MiniZinc: Towards a Standard CP Modelling Language". In: *Principles and Practice of Constraint Programming – CP 2007*. Ed. by C. Bessière. Vol. 4741. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, pp. 529–543. DOI: 10.1007/978-3-540-74970-7_38 (cit. on p. 395).

[NC03] A. Nigam and N. S. Caswell. "Business Artifacts: An Approach to Operational Specification". In: *IBM Systems Journal* 42.3 (2003), pp. 428–445 (cit. on pp. 1, 29–31, 39, 40, 253, 280).

[NR02] M. Nüttgens and F. J. Rump. "Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK)". German. In: *Promise 2002 - Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen, Proceedings des GI-Workshops und Fachgruppentreffens*. Potsdam, 2002, pp. 64–77 (cit. on p. 68).

[OAS07] OASIS. *Web Services Business Process Execution Language Version 2 . 0*. Standard April. OASIS, 2007, p. 264 (cit. on pp. 2, 16, 78, 254, 315).

[OAS13] OASIS. *Content Management Interoperability Services (CMIS), Version 1.1, OASIS Standard*. Standard. OASIS, 2013, p. 331 (cit. on pp. 9, 258).

[OMG01] OMG. *OMG Unified Modeling Language Specification, version 1.4*. Standard. Document formal/2001-09-67. OMG, 2001, p. 1034 (cit. on p. 67).

[OMG02] OMG. *Software Process Engineering Metamodel Specification (SPEM), version 1.0*. Standard. Document formal/02-11-14. OMG, 2002 (cit. on pp. 285, 311).

[OMG09a] OMG. *Business Process Model and Notation (BPMN), version 1.2*. Standard. Document formal/2009-01-03. OMG, 2009, p. 316 (cit. on p. 66).

[OMG09b] OMG. *Case Management Process Modeling (CMPM) Request for Proposal*. Request for Proposal. Document bmi/09-09-23. Needham, MA: OMG, 2009, pp. 1–43 (cit. on pp. 37, 66, 256).

[OMG09c] OMG. *OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.2*. Standard. Document formal/2009-02-02. OMG, 2009, p. 740 (cit. on pp. 2, 16, 22, 29, 65, 194, 287, 288, 322).

[OMG11] OMG. *OMG Unified Modeling Language (OMG UML), Infrastructure, version 2.4.1*. Standard. Document formal/2011-08-05. OMG, 2011, p. 230 (cit. on p. 66).

[OMG13] OMG. *Business Process Model and Notation (BPMN), version 2.0.2*. Standard. Document formal/2013-12-09. OMG, 2013, p. 532 (cit. on pp. 2, 16, 60, 65, 66, 78, 123, 178, 194, 255, 316).

[OMG14a] OMG. *Case Management Model and Notation, version 1.0*. Standard. Document formal/2014-05-05. OMG, 2014, p. 96 (cit. on pp. 1, 3, 11, 37, 39, 48, 54, 62, 65, 73, 75, 77–79, 89, 90, 92, 93, 97, 103, 135, 157, 193, 194, 256, 391).

[OMG14b] OMG. *Object Constraint Language (OCL) version 2.4*. Standard. Document formal/2014-02-03. OMG, 2014, p. 244 (cit. on pp. 45, 79, 280).

[Ozc13] Y. Ozcelik. “Effects of Business Process Reengineering on Firm Performance: An Econometric Analysis”. In: *Business Process Management: Theory and Applications*. Ed. by M. Glykas. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 99–110. DOI: 10.1007/978-3-642-28409-0_4 (cit. on p. 255).

[Pal10] N. Palmer. “Introduction”. In: *Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done*. Ed. by K. D. Swenson. 1st ed. Tampa, Florida, USA: Meghan-Kiffer Press, 2010. Chap. Introduction, pp. 1–4 (cit. on p. 28).

[PG08] R. M. Parizi and A. A. A. Ghani. "An Ensemble of Complexity Metrics for BPEL Web Processes". In: *9th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing.* IEEE, 2008, pp. 753–758. DOI: 10.1109/SNPD.2008.152 (cit. on p. 331).

[Par+13] L. Parody, M. T. Gòmez-Lòpez, R. M. Gasca, and A. J. Varela-Vaca. "CombiS-BP Editor: Combining Declarative and Imperative Languages in BP Modelling: Demonstration Paper". In: *IEEE 7th International Conference on Research Challenges in Information Science (RCIS).* 2013. DOI: 10.1109/RCIS.2013.6577745 (cit. on pp. 1, 20, 21).

[Pas66] T. L. Pascoe. "Allocation of Resources - CPM." In: *Revue Francaise de Recherche Opérationelle* 38 (1966), pp. 31–38 (cit. on p. 310).

[Per+11a] C. M. Pereira, A. Caetano, and P. Sousa. "Ontology-Driven Business Process Design". In: *Building the e-World Ecosystem: 11th IFIP WG 6.11 Conference on e-Business, e-Services, and e-Society, I3E 2011, Kaunas, Lithuania, October 12-14, 2011, Revised Selected Papers.* Ed. by T. Skersys, R. Butleris, L. Nemuraite, and R. Suomi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 153–162. DOI: 10.1007/978-3-642-27260-8_12 (cit. on pp. 18, 19, 194).

[Per+11b] C. M. Pereira, A. Caetano, and P. Sousa. "Using a Controlled Vocabulary to Support Business Process Design". In: *Enterprise and Organizational Modeling and Simulation: 7th International Workshop, EOMAS 2011, held at CAiSE 2011, London, UK, June 20-21, 2011. Selected Papers.* Ed. by J. Barjis, T. Eldabi, and A. Gupta. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 74–84. DOI: 10.1007/978-3-642-24175-8_6 (cit. on pp. 18, 19, 194).

[Pv06] M. Pesic and W. M. P. van der Aalst. "A Declarative Approach for Flexible Business Processes Management". In: *Business Process Management Workshops: BPM 2006 International Workshops, BPD, BPI, ENEI, GPWW, DPM, semantics4ws.* Ed. by J. Eder and S. Dustdar. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. Chap. A Declarative Approach for Flexible Business Processes Management, pp. 169–180. DOI: 10.1007/11837862_18 (cit. on pp. 21, 38).

[Pes+07] M. Pesic, H. Schonenberg, and W. M. P. van der Aalst. "DECLARE: Full Support for Loosely-Structured Processes". In: *Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International.* 2007, pp. 287–287. DOI: 10.1109/EDOC.2007.14 (cit. on pp. 2, 21, 260).

[Pfa03] B. Pfaffenberger. *Webster's New World Computer Dictionary.* Wiley Pub., 2003 (cit. on p. 263).

[Pic+12] P. Pichler, B. Weber, S. Zugal, J. Pinggera, J. Mendling, and H. A. Reijers. "Imperative versus Declarative Process Modeling Languages: An Empirical Investigation". In: *Business Process Management Workshops: BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I.* Ed. by F. Daniel, K. Barkaoui, and S. Dustdar. Springer Berlin Heidelberg, 2012, pp. 383–394. DOI: 10.1007/978-3-642-28108-2_37 (cit. on pp. 2, 5, 20).

[Pin+12a] J. Pinggera, P. Soffer, S. Zugal, B. Weber, M. Weidlich, D. Fahland, H. A. Reijers, and J. Mendling. "Modeling Styles in Business Process Modeling". In: *Enterprise, Business-Process and Information Systems Modeling: 13th International Conference, BPMDS 2012, 17th International Conference, EMMSAD 2012, and 5th EuroSymposium, held at CAiSE 2012, Gdańsk, Poland, June 25-26, 2012. Proceedings.* Ed. by I. Bider, T. Halpin, J. Krogstie, S. Nurcan, E. Proper, R. Schmidt, P. Soffer, and S. Wrycza. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 151–166. DOI: 10.1007/978-3-642-31072-0_11 (cit. on p. 15).

[Pin+12b] J. Pinggera, S. Zugal, M. Weidlich, D. Fahland, B. Weber, J. Mendling, and H. A. Reijers. "Tracing the Process of Process Modeling with Modeling Phase Diagrams". In: *Business Process Management Workshops: BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I.* Ed. by F. Daniel, K. Barkaoui, and S. Dustdar. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 370–382. DOI: 10.1007/978-3-642-28108-2_36 (cit. on p. 15).

[Pit15] V. F. Pittke. "Linguistic Refactoring of Business Process Models". PhD thesis. Wirtschaftsuniversität Wien, Vienna University of Economics and Business, 2015 (cit. on pp. 11, 12, 15).

[PD00] G. Poels and G. Dedene. "Distance-based Software Measurement: Necessary and Sufficient Properties for Software Measures". In: *Information and Software Technology* 42.1 (2000), pp. 35–46. DOI: 10.1016/S0950-5849(99)00053-1 (cit. on pp. 107, 124).

[PC16] G. Polančič and B. Cegnar. "Complexity Metrics for Process Models - A Systematic Literature Review". In: *Computer Standards & Interfaces* (2016). DOI: 10.1016/j.csi.2016.12.003 (cit. on pp. 104–107, 130–132, 195).

[Pol13] Š. Poljak. "Conceptual Modeling of Business Artifacts and their Implementation as Active XML". MA thesis. Charles University in Prague, Faculty of Mathematics and Physics, 2013 (cit. on p. 29).

[PD13] V. Popova and M. Dumas. "From Petri Nets to Guard-Stage-Milestone Models". In: *Business Process Management Workshops: BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers*. Ed. by M. La Rosa and P. Soffer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. Chap. From Petri Nets to Guard-Stage-Milestone Models, pp. 340–351. DOI: 10.1007/978-3-642-36285-9_38 (cit. on p. 78).

[Pop+15] V. Popova, D. Fahland, and M. Dumas. "Artifact Lifecycle Discovery". In: *International Journal of Cooperative Information Systems* 24.01 (2015), p. 1550001. DOI: 10.1142/S021884301550001X (cit. on p. 78).

[Por10] R. H. D. B. V. D. Porciúncula. "Governança e Modelação de Processos de TI". Portuguese. MA thesis. Universidade Nova de Lisboa, 2010 (cit. on pp. 106, 348).

[Pre+14] J. Prescher, C. di Ciccio, and J. Mendling. "From Declarative Processes to Imperative Models". In: *Proceedings of the 4^{th} International Symposium on Data-driven Process Discovery and Analysis (SIMPDA 2014)*. 2014, pp. 162–173 (cit. on pp. 5, 20, 21).

[Puc10] M. J. Pucher. *The Difference between DYNAMIC and ADAPTIVE*. Accessed: April 29, 2015. ISIS Papyrus. 2010. URL: https://acmisis.wordpress.com/2010/11/18/the-difference-between-dynamic-and-adaptive/ (cit. on pp. 28, 29).

[R C16] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2016 (cit. on pp. 109, 113, 119, 179, 182, 393).

[Rec+09] J. C. Recker, M. zur Muehlen, K. Siau, J. Erickson, and M. Indulska. "Measuring Method Complexity: UML versus BPMN". In: *15^{th} Americas Conference on Information Systems*. San Francisco, 2009, pp. 6–9 (cit. on pp. 5, 26, 65, 66, 68–70, 75, 197).

[Red+08] G. Redding, M. Dumas, A. H. M. Ter Hofstede, and A. Iordachescu. "Transforming Object-Oriented Models to Process-Oriented Models". In: *Business Process Management Workshops: BPM 2007 International Workshops, BPI, BPD, CBP, ProHealth, RefMod, semantics4ws, Brisbane, Australia, September 24, 2007, Revised Selected Papers*. Ed. by A. Ter Hofstede, B. Benatallah, and H.-Y. Paik. Berlin, Heidelberg: Springer Berlin

Heidelberg, 2008, pp. 132–143. DOI: 10.1007/978-3-540-78238-4_15 (cit. on p. 34).

[Red+09] G. Redding, M. Dumas, A. H. M. Ter Hofstede, and A. Iordachescu. "Modelling Flexible Processes with Business Objects". In: *Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing.* CEC '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 41–48. DOI: 10.1109/CEC.2009.39 (cit. on p. 34).

[Reg+06] G. Regev, P. Soffer, and R. Schmidt. "Taxonomy of Flexibility in Business Processes". In: *Proceedings of the CAISE*06 Workshop on Business Process Modelling, Development, and Support BPMDS'06.* Luxemburg, 2006 (cit. on p. 20).

[Rei12] M. Reichert. "Process and Data: Two Sides of the Same Coin?" In: *On the Move to Meaningful Internet Systems: OTM 2012: Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE 2012, Rome, Italy, September 10-14, 2012. Proceedings, Part I.* Ed. by R. Meersman, H. Panetto, T. Dillon, S. Rinderle-Ma, P. Dadam, X. Zhou, S. Pearson, A. Ferscha, S. Bergamaschi, and I. F. Cruz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 2–19. DOI: 10.1007/978-3-642-33606-5_2 (cit. on pp. 22, 34).

[RW12] M. Reichert and B. Weber. "A Framework for Object-Aware Processes". In: *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 405–438. DOI: 10.1007/978-3-642-30409-5_14 (cit. on p. 34).

[Rei03] H. A. Reijers. "A Cohesion Metric for the Definition of Activities in a Workflow Process". In: *Proceedings EMMSAD.* 2003, pp. 116–125 (cit. on pp. 311, 313).

[RV04] H. A. Reijers and I. Vanderfeesten. "Cohesion and Coupling Metrics for Workflow Process Design". In: *2nd International Conference on Business Process Management.* Berlin: Springer Verlag, 2004, pp. 290–305 (cit. on pp. xxii, xxvii, xxxiv, 24, 25, 124, 281, 282, 313, 314, 337, 381, 382).

[Rei+03] H. A. Reijers, J. Rigter, and W. M. P. van der Aalst. "The Case Handling Case". In: *International Journal of Cooperative Information Systems* 12.3 (2003), pp. 365–391 (cit. on pp. 27, 48, 67, 75).

[Rei+10a] H. A. Reijers, S. van Wijk, B. Mutschler, and M. Leurs. "BPM in Practice: Who Is Doing What?" In: *Business Process Management: 8th International Conference, BPM 2010, Hoboken, NJ, USA, September 13-16, 2010. Proceedings.* Ed. by R. Hull, J. Mendling, and S. Tai. Berlin, Heidelberg:

Springer Berlin Heidelberg, 2010, pp. 45–60. DOI: 10.1007/978-3-642-15618-2_6 (cit. on p. 12).

[Rei+10b] H. A. Reijers, J. Mendling, and J. Recker. “Business Process Quality Management”. In: *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems.* Ed. by J. vom Brocke and M. Rosemann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 167–185. DOI: 10.1007/978-3-642-00416-2_8 (cit. on p. 14).

[Rei+13] H. A. Reijers, T. Slaats, and Ch. Stahl. “Declarative Modeling – An Academic Dream or the Future for BPM?” In: *Business Process Management: 11*th *International Conference, BPM 2013, Beijing, China, August 26-30, 2013. Proceedings.* Ed. by F. Daniel, J. Wang, and B. Weber. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 307–322. DOI: 10.1007/978-3-642-40176-3_26 (cit. on p. 20).

[Rie+16] D. M. Riehle, S. Jannaber, A. Karhof, O. Thomas, P. Delfmann, and J. Becker. “On the de-facto Standard of Event-driven Process Chains: How EPC is Defined in Literature.” In: *Modellierung.* 2016, pp. 61–76 (cit. on p. 261).

[Rol09] E. A. Rolón. “Medidas para Asegurar la Calidad de los Modelos de Proceso de Negocio”. Spanish. PhD thesis. Universidad de Castilla-La Mancha, 2009, p. 424 (cit. on pp. 124, 317, 361, 367–381, 383, 384).

[Rol+05] E. A. Rolón, F. Ruíz, F. García, and M. Piattini. “Aplicación de Métricas Software en la Evaluación de Modelos de Procesos de Negocio”. Spanish. In: *Revista Electrónica de la Sociedad Chilena de Ciencia de la Computacion* 6.1 (2005), p. 10 (cit. on pp. 25, 316, 343, 347, 351).

[Rol+06a] E. A. Rolón, F. Ruíz, F. García, and M. Piattini. “Applying Software Metrics to Evaluate Business Process Models”. In: *Clei Electronic Journal* 9 (1 2006) (cit. on p. 316).

[Rol+06b] E. A. Rolón, F. Ruíz, F. García, and M. Piattini. “Applying Software Process Metrics in Bussiness Process Model”. In: *Revista de Processos y Metricas* 3.2 (2006), pp. 45–61 (cit. on pp. xxiv, xxix–xxxiv, xxxvii, 124, 136, 137, 157, 258, 266, 268–278, 280, 283, 286, 287, 316, 317, 323, 328, 343, 344, 349, 350, 356, 361, 367–381, 383, 384).

[Rol+06c] E. A. Rolón, F. Ruíz, F. García, and M. Piattini. “Evaluation Measures for Business Process Models”. In: *Proceedings of the 2006 ACM symposium on Applied computing - SAC ’06.* New York: ACM Press, 2006, pp. 1567–1568. DOI: 10.1145/1141277.1141641 (cit. on p. 316).

[Rol+06d] E. A. Rolón, F. Ruíz, F. García, and M. Piattini. “Towards a Suite of Metrics for Business Process Models in BPMN”. In: *Proceedings of the Eighth International Conference on Enterprise Information Systems (ICEIS 2006).* Ed. by Y. Manolopoulos, J. Filipe, P. Constantopoulos, and J. Cordeiro. 2006, pp. 440–443 (cit. on p. 316).

[Rol+07a] E. A. Rolón, F. García, F. Ruíz, and M. Piattini. “An Exploratory Experiment to Validate Measures for Business Process Models.” In: *Proceedings of the First International Conference on Research Challenges in Information Science, RCIS 2007.* Ed. by C. Rolland, O. Pastor, and J.-L. Cavarero. 2007, pp. 271–280 (cit. on pp. 128, 196, 316, 361, 367–381, 383, 384).

[Rol+07b] E. A. Rolón, F. García, F. Ruíz, and M. Piattini. “Familia de Experimentos para Validar Medidas para Modelos de Procesos de Negocio con BPMN”. Spanish. In: *Taller sobre Procesos de Negocio e Ingeniería del Software.* Vol. 1(2). 2007, pp. 41–48 (cit. on p. 316).

[Rol+08] E. A. Rolón, F. García, F. Ruíz, M. Piattini, C. A. Visaggio, and C. Gerardo. “Evaluation of BPMN Models Quality: a Family of Experiments”. In: *3rd International Conference on Evaluation of Novel Approaches to Software Engineering.* Funchal, Portugal, 2008, pp. 56–63 (cit. on pp. 128, 159, 196, 317, 343, 361, 367–381, 383, 384).

[Rol+09a] E. A. Rolón, J. Cardoso, F. García, F. Ruíz, and M. Piattini. “Analysis and Validation of Control-Flow Complexity Measures with BPMN Process Models”. In: *Enterprise, Business-Process and Information Systems Modeling: 10th International Workshop, BPMDS 2009, and 14th International Conference, EMMSAD 2009, held at CAiSE 2009, Amsterdam, The Netherlands, June 8-9, 2009. Proceedings.* Ed. by T. Halpin, J. Krogstie, S. Nurcan, E. Proper, R. Schmidt, P. Soffer, and R. Ukor. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 58–70. DOI: 10.1007/978-3-642-01862-6_6 (cit. on pp. 128, 129, 196, 343, 362).

[Rol+09b] E. A. Rolón, L. Sánchez-González, F. García, F. Ruíz, M. Piattini, D. Caivano, and G. Visaggio. “Prediction Models for BPMN Usability and Maintainability”. In: *2009 IEEE Conference on Commerce and Enterprise Computing* (2009), pp. 383–390. DOI: 10.1109/CEC.2009.53 (cit. on pp. 128, 196, 343, 361, 367, 369–374, 376–379, 381, 383, 384).

[RV11] W. D. Roover and J. Vanthienen. “Unified Patterns to Transform Business Rules Into an Event Coordination Mechanism”. In: *Business Process Management Workshops: BPM 2010 International Workshops.* Ed. by M. zur

Muehlen and J. Su. Hoboken, NJ: Springer-Verlag Berlin Heidelberg, 2011, pp. 730–742 (cit. on pp. 260, 281).

[RB96] M. Rossi and S. Brinkkemper. “Complexity Metrics for Systems Development Methods and Techniques”. In: *Information Systems* 21.2 (1996), pp. 209–227 (cit. on pp. xxii, xxviii, xxix, xxxiii, xxxiv, xxxvi, xxxviii, 4, 26, 65–71, 73–75, 194, 195, 198, 199, 265, 280).

[Rus+13] A. Russo, M. Mecella, F. Patrizi, and M. Montali. “Implementing and Running Data-Centric Dynamic Systems”. In: *2013 IEEE 6th International Conference on Service-Oriented Computing and Applications.* Dec. 2013, pp. 225–232. DOI: 10.1109/SOCA.2013.37 (cit. on p. 36).

[Rus+14] A. Russo, M. Mecella, M. Montali, and F. Patrizi. “Towards a Reference Implementation for Data Centric Dynamic Systems”. In: *Business Process Management Workshops: BPM 2013 International Workshops, Beijing, China, August 26, 2013, Revised Papers.* Ed. by N. Lohmann, M. Song, and P. Wohed. Springer International Publishing, 2014. Chap. Towards a Reference Implementation for Data Centric Dynamic Systems, pp. 141–154. DOI: 10.1007/978-3-319-06257-0_12 (cit. on pp. 21, 23, 34).

[Saa80] T. L. Saaty. *The Analytic Hierarchy Process, Planning, Piority Setting, Resource Allocation.* New york: McGraw-Hill, 1980 (cit. on p. 171).

[S+10a] L. Sánchez-González, F. G. Rubio, F. R. González, and M. P. Velthuis. “Measurement in Business Processes: A Systematic Review”. In: *Business Process Management Journal* 16.1 (2010), pp. 114–134. DOI: 10.1108/14637151011017976 (cit. on pp. 25, 104–106, 130, 131, 195).

[S+10b] L. Sánchez-González, F. García, J. Mendling, F. Ruíz, and M. Piattini. “Prediction of Business Process Model Quality Based on Structural Metrics”. In: *Conceptual Modeling – ER 2010: 29th International Conference on Conceptual Modeling, Vancouver, BC, Canada, November 1-4, 2010. Proceedings.* Ed. by J. Parsons, M. Saeki, P. Shoval, C. Woo, and Y. Wand. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 458–463. DOI: 10.1007/978-3-642-16373-9_35 (cit. on pp. 125, 128, 129, 350, 356, 359–361, 363, 364, 366, 367, 385, 386).

[S+11] L. Sánchez-González, F. Ruíz, F. García, and J. Cardoso. “Towards Thresholds of Control Flow Complexity Measures for BPMN Models”. In: *Proceedings of the 2011 ACM Symposium on Applied Computing.* SAC ’11. TaiChung, Taiwan: ACM, 2011, pp. 1445–1450. DOI: 10.1145/1982185.1982496 (cit. on pp. 106, 350, 351).

[Ś+12] L. Sánchez-GonzáLez, F. García, F. Ruíz, and J. Mendling. “Quality Indicators for Business Process Models from a Gateway Complexity Perspective”. In: *Information and Software Technology* 54.11 (2012), pp. 1159–1174. DOI: 10.1016/j.infsof.2012.05.001 (cit. on pp. 106, 350, 351).

[Sán+15] L. Sánchez-González, F. García, F. Ruíz, and M. Piattini. “A Case Study About the Improvement of Business Process Models Driven by Indicators”. In: *Software & Systems Modeling* (2015), pp. 1–30. DOI: 10.1007/s10270-015-0482-0 (cit. on pp. 106, 350).

[Sav08] J. E. Savage. *Models of Computation: Exploring the Power of Computing.* electronic edition, 2008 (cit. on p. 261).

[Sch+13] S. Schönig, M. Zeising, and S. Jablonski. “Supporting Collaborative Work by Learning Process Models and Patterns from Cases”. In: *2013 9th International Conference Conference on Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom).* IEEE. 2013, pp. 60–69 (cit. on pp. 5, 197).

[Sch+15] S. Schulte, Ch. Janiesch, S. Venugopal, I. Weber, and Ph. Hoenisch. “Elastic Business Process Management: State of the Art and Open Challenges for BPM in the Cloud”. In: *Future Generation Computer Systems* 46 (2015), pp. 36–50. DOI: 10.1016/j.future.2014.09.005 (cit. on p. 13).

[Sch95] C. Schwindt. *ProGen/max: A New Problem Generator for Different Resource-constrained Project Scheduling Problems with Minimal and Maximal Time Lags. Report WIOR-449.* Report WIOR-449. Karlsruhe, Germany: Institut Für Wirtschaftstheorie Und Operations Research, Universität Karlsruhe, 1995, p. 57 (cit. on p. 310).

[SB91] R. W. Selby and V. R. Basili. “Analyzing Error-prone System Structure”. In: *IEEE Transactions on Software Engineering* 17.2 (1991), pp. 141–152. DOI: 10.1109/32.67595 (cit. on p. 311).

[Ser+02] M. Serrano, C. Calero, and M. Piattini. “Validating Metrics for Data Warehouses”. In: *Software, IEE Proceedings* - 149.5 (2002), pp. 161–166. DOI: 10.1049/ip-sen:20020697 (cit. on pp. 106, 107, 131).

[SS13] M. A. Setiawan and Sh. Sadiq. “Integrated Framework For Business Process Complexity Analysis”. In: *Proceedings of the 21st European Conference on Information Systems (ECIS 2013).* 2013 (cit. on p. 354).

[Sha48] C. E. Shannon. “A Mathematical Theory of Communication”. In: *Bell System Technical Journal* 27.July and October (1948), pp. 379–423,623–656 (cit. on pp. 321, 354).

[SW03] J. Shao and Y. Wang. “A New Measure of Software Complexity Based on Cognitive Weights”. In: *IEEE CCECE 2003. Canadian Conference on Electrical and Computer Engineering, 2003.* Vol. 2. 2003, pp. 1333–1338. DOI: 10.1109/CCECE.2003.1226146 (cit. on pp. 136, 142, 323, 325, 345).

[Shi+16] Zh. Shi, X. Zeng, T. Zhang, S. Huang, Z. Qi, H. Li, B. Hu, Y. Yao, and Sh. Zhong. “Bidirectional Transformation Between BPMN and BPEL with Graph Grammar”. In: *Computers & Electrical Engineering* (2016). DOI: 10.1016/j.compeleceng.2015.12.027 (cit. on p. 78).

[SC02] K. Siau and Q. Cao. “How Complex is the Unified Modeling Language?” In: *Advanced Topics in Database Research Vol. 1.* Ed. by K. Siau. Hershey, PA, USA: IGI Global, 2002, pp. 294–306. DOI: 10.4018/978-1-930708-41-9.ch015 (cit. on pp. 5, 65, 68, 70, 74, 75, 194, 197).

[SR98] K. Siau and M. Rossi. “Evaluation of Information Modeling Methods – A Review”. In: *Proceedings of the Thirty-First Hawaii International Conference on System Sciences, 1998.* Vol. 5. IEEE Comput. Soc, 1998, pp. 314–322. DOI: 10.1109/HICSS.1998.648327 (cit. on p. 67).

[Sjo+07] D. I. K. Sjoberg, T. Dyba, and M. Jorgensen. “The Future of Empirical Methods in Software Engineering Research”. In: *Future of Software Engineering, 2007. FOSE ’07.* May 2007, pp. 358–378. DOI: 10.1109/FOSE.2007.30 (cit. on pp. 103, 132, 133, 198).

[Smo+91] K. Smolander, K. Lyytinen, V. Tahvanainen, and P. Marttiin. “MetaEdit – A Flexible Graphical Environment for Methodology Modelling”. In: *Advanced Information Systems Engineering.* Ed. by R. Andersen, J. A. Bubenko Jr., and A. Solvberg. Vol. 498. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1991, pp. 168–193. DOI: 10.1007/3-540-54059-8_85 (cit. on pp. 70, 199).

[Sob99] F. G. Sobrinho. “Complexity Measures for Process Evolution”. In: *Journal of Systems Integration* 9.2 (1999), pp. 141–165. DOI: 10.1023/A:1008470510343 (cit. on pp. xxvi–xxviii, 262, 265, 308, 364–366).

[Sof12] Software Engineering Research Laboratory (LaPES). *StArt – State of the Art through Systematic Review Tool.* 2012 (cit. on pp. 24, 285).

[Sol+13a] I. Solichah, M. Hamilton, P. Mursanto, C. Ryan, and M. Perepletchikov. “Exploration on Software Complexity Metrics for Business Process Model and Notation”. In: *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS).* 2013, pp. 31–37. DOI: 10.1109/ICACSIS.2013.6761549 (cit. on pp. 106, 124, 354, 360, 362, 382).

[Sol+12] D. Solomakhin, M. Montali, and S. Tessaris. *Formalizing Guard-Stage-Milestone Meta-Models as Data-centric Dynamic Systems.* Tech. rep. Technical Report KRDB12-4. KRDB Research Centre, Faculty of Computer Science, Free University of Bozen-Bolzano, 2012, p. 34 (cit. on pp. 5, 78, 197).

[Sol+13b] D. Solomakhin, M. Montali, S. Tessaris, and R. de Masellis. “Verification of Artifact-Centric Systems: Decidability and Modeling Issues”. In: *Service-Oriented Computing: 11th International Conference, ICSOC 2013, Berlin, Germany, December 2-5, 2013, Proceedings.* Ed. by S. Basu, C. Pautasso, L. Zhang, and X. Fu. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. Chap. Verification of Artifact-Centric Systems: Decidability and Modeling Issues, pp. 252–266. DOI: 10.1007/978-3-642-45005-1_18 (cit. on pp. 2, 29, 46, 78, 195).

[Son09] D. Soni. “A Framework for Validation of Object-Oriented Design Metrics”. In: *International Journal of Computer Science and Information Security* 6.3 (2009), pp. 46–52 (cit. on pp. 106, 107, 131).

[Sou+07] P. Sousa, C. Pereira, R. Vendeirinho, A. Caetano, and J. Tribolet. “Applying the Zachman Framework Dimensions to Support Business Process Modeling”. In: *Digital Enterprise Technology: Perspectives and Future Challenges.* Ed. by P. F. Cunha and P. G. Maropoulos. Boston, MA: Springer US, 2007, pp. 359–366. DOI: 10.1007/978-0-387-49864-5_42 (cit. on pp. 18, 19, 194).

[SZ92] J. F. Sowa and J. A. Zachman. “Extending and Formalizing the Framework for Information Systems Architecture”. In: *IBM Systems Journal* 31.3 (1992), pp. 590–616. DOI: 10.1147/sj.313.0590 (cit. on pp. 16, 18–20, 22–24, 194).

[SD14] K. P. Srinivasan and T. Devi. “Software Metrics Validation Methodologies In Software Engineering”. In: *International Journal of Software Engineering & Applications* 5.6 (2014), pp. 87–102. DOI: 10.5121/ijsea.2014.5606 (cit. on pp. 152, 157, 158).

[Sta73] H. Stachowiak. *Allgemeine Modelltheorie.* German. Wien, New York, 1973 (cit. on p. 15).

[Ste+74] W. P. Stevens, G. J. Myers, and L. L. Constantine. “Structured Design”. In: *IBM Systems Journal* 13.2 (1974), pp. 115–139. DOI: 10.1147/sj.132.0115 (cit. on p. 311).

[Str+08] J. K. Strosnider, P. Nandi, S. Kumaran, S. Ghosh, and A. Arsanjani. “Model-driven Synthesis of SOA Solutions”. In: *IBM Systems Journal* 47.3 (2008), pp. 415–432. DOI: 10.1147/sj.473.0415 (cit. on pp. 34, 254).

[Sue+90] C. Y. Suen, P. D. Grogono, R. Shinghal, and F. Coallier. "Verifying, Validating, and Measuring the Performance of Expert Systems". In: *Expert Systems with Applications* 1.2 (1990), pp. 93–102. DOI: 10.1016/0957-4174(90)90019-Q (cit. on p. 356).

[SH14] H. Sun and H. Hou. "Study on Complexity Metrics of Business Process". In: *International Conference on Computer Science and Service System (CSSS 2014)*. Atlantis Press, 2014. DOI: 10.2991/csss-14.2014.68 (cit. on pp. xxvii, 124, 125, 263, 354, 355, 365).

[Sun+12] Y. Sun, R. Hull, and R. Vaculín. "Parallel Processing for Business Artifacts with Declarative Lifecycles". In: *On the Move to Meaningful Internet Systems: OTM 2012*. Ed. by R. Meersman, H. Panetto, T. Dillon, S. Rinderle-Ma, P. Dadam, X. Zhou, S. Pearson, A. Ferscha, S. Bergamaschi, and I. F. Cruz. Vol. 7565. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 433–443. DOI: 10.1007/978-3-642-33606-5_27 (cit. on pp. 41, 44–46).

[Swa07] B. R. Swan. "The Effects of Business Process Management Cognitive Resources and Individual Cognitive Differences on Outcomes of User Comprehension". Phd. Virginia Polytechnic Institute and State University, 2007, p. 420 (cit. on pp. 1, 2, 262, 288).

[Swe10a] K. D. Swenson, ed. *Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done*. Landmark books. Meghan-Kiffer Press, 2010 (cit. on pp. 28, 48, 67, 75).

[Swe10b] K. D. Swenson. "The Nature of Knowledge Work". In: *Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done*. Ed. by K. D. Swenson. 1st ed. Tampa, Florida, USA: Meghan-Kiffer Press, 2010. Chap. The Nature of Knowledge Work, pp. 5–28 (cit. on p. 28).

[Swe13] K. D. Swenson. *State of the Art in Case Management*. white paper. Fujitsu, 2013 (cit. on pp. 28, 251, 282).

[Tem81] H. M. V. Temperley. *Graph Theory and Applications*. Chichester, West Sussex, England: Ellis Horwood Limited, 1981, p. 130 (cit. on pp. 310, 311).

[Ter+10] A. H. M. Ter Hofstede, W. M. P. van der Aalst, M. Adams, and N. Russell. *Modern Business Process Automation: YAWL and its Support Environment*. Springer Berlin Heidelberg, 2010. DOI: 10.1007/978-3-642-03121-2 (cit. on pp. 16, 289, 322).

[Tja99] G. S. Tjaden. *Business Process Structural Analysis.* Tech. rep. Georgia Tech Center for Enterprise Systems, 1999 (cit. on pp. xxii, xxvii, xxxv, 252, 262, 263, 284, 309, 358, 365, 385).

[TZ81] D. A. Troy and S. H. Zweben. "Measuring the Quality of Structured Designs". In: *Journal of Systems and Software* 2.2 (1981), pp. 113–120. DOI: 10.1016/0164-1212(81)90031-5 (cit. on p. 25).

[Uni07] University of South Africa. *Policy on Research Ethics.* Tech. rep. University of South Africa, 2007, p. 29 (cit. on p. 178).

[U.S09] U.S. Department of Defense. *Enterprise Architecture based on Design Primitives and Patterns Guidelines for the Design and Development of Event-Trace Descriptions (DoDAF OV-6c) using BPMN.* Tech. rep. Business Transformation Agency, 2009, p. 51 (cit. on pp. 66, 75, 194).

[van95] W. M. P. van der Aalst. *A Class of Petri Nets for Modeling and Analyzing Business Processes.* Tech. rep. No. 95-26. Eindhoven, The Netherlands: Technische Universiteit Eindhoven, 1995, pp. 1–25 (cit. on pp. 21, 288, 322).

[van97] W. M. P. van der Aalst. "Verification of Workflow Nets". In: *Application and Theory of Petri Nets 1997* 1248 (1997), pp. 407–426 (cit. on pp. 21, 324).

[van98] W. M. P. van der Aalst. "The Application of Petri Nets to Workflow Management". In: *The Journal of Circuits, Systems and Computers* 8.1 (1998), pp. 21–66 (cit. on pp. 321, 322).

[van99] W. M. P. van der Aalst. "Formalization and Verification of Event-driven Process Chains". In: *Information and Software Technology* 41.10 (1999), pp. 639–650. DOI: 10.1016/S0950-5849(99)00016-6 (cit. on pp. 4, 16, 65, 123, 194, 261, 305, 324).

[van13] W. M. P. van der Aalst. "Business Process Management: A Comprehensive Survey". In: *ISRN Software Engineering* (2013). DOI: 10.1155/2013/507984 (cit. on pp. 3, 12).

[van15] W. M. P. van der Aalst. "Business Process Management as the 'Killer App' for Petri Nets". In: *Software & Systems Modeling* 14.2 (2015), pp. 685–691. DOI: 10.1007/s10270-014-0424-2 (cit. on p. 12).

[vB01] W. M. P. van der Aalst and P. J. S. Berens. "Beyond Workflow Management: Product-Driven Case Handling". In: *Proceedings of the 2001 International ACM SIGGROUP.* New York: ACM Press, 2001, pp. 42–51. DOI: 10.1145/500286.500296 (cit. on pp. 1, 27, 36, 48, 67, 75).

[vT05] W. M. P. van der Aalst and A. H. M. Ter Hofstede. "YAWL: Yet Another Workflow Language". In: *Information Systems* 30.4 (2005), pp. 245–275. DOI: 10.1016/j.is.2004.02.002 (cit. on pp. 27, 289, 324).

[van+00] W. M. P. van der Aalst, A. P. Barros, A. H. M. Ter Hofstede, and B. Kiepuszewski. "Advanced Workflow Patterns". In: *Cooperative Information Systems: 7th International Conference, CoopIS 2000 Eilat, Israel, September 6-8, 2000. Proceedings.* Ed. by P. Scheuermann and O. Etzion. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 18–29. DOI: 10.1007/10722620_2 (cit. on pp. 331, 332).

[van+03] W. M. P. van der Aalst, B. Kiepuszewski, A. P. Barros, and A. H. M. Ter Hofstede. "Workflow Patterns". In: *Distributed and Parallel Databases* 14.3 (2003), pp. 5–51 (cit. on pp. 327, 331, 332).

[van+05] W. M. P. van der Aalst, M. Weske, and D. Grunbauer. "Case Handling: A New Paradigm for Business Process Support". In: *Data and Knowledge Engineering* 53.2 (2005), pp. 129–162 (cit. on pp. 27, 36, 37, 48, 255).

[van+09] W. M. P. van der Aalst, M. Pesic, and H. Schonenberg. "Declarative Workflows: Balancing Between Flexibility and Support". In: *Computer Science - Research and Development* 23.2 (2009), pp. 99–113. DOI: 10.1007/s00450-009-0057-9 (cit. on pp. 2, 21).

[van+13] J. van Grondelle, M. Zoet, and F. Vermeer. "Characterizing Declarativity across Business Process Formalisms". In: *Proceedings of the Annual Conferences of the International Information Management Association (IIMA 2013).* 2013 (cit. on p. 20).

[Van+07a] I. Vanderfeesten, J. Cardoso, and H. A. Reijers. "A Weighted Coupling Metric for Business Process Models". In: *19th International Conference on Advanced Information Systems Engineering Forum.* Trondheim, Norway, 2007, pp. 41–44 (cit. on pp. xxiv, 25, 259, 331, 333, 362).

[Van+07b] I. Vanderfeesten, J. Cardoso, J. Mendling, H. A. Reijers, and W. M. P. van der Aalst. "Quality Metrics for Business Process Models". In: *BPM and Workflow Handbook 2007.* Ed. by L. Fischer. Lighthouse Point, Florida: Future Strategies Inc., 2007, pp. 1–12 (cit. on pp. 24, 25).

[Van+08a] I. Vanderfeesten, H. A. Reijers, and W. M. P. van der Aalst. "Evaluating Workflow Process Designs Using Cohesion and Coupling Metrics". In: *Computers in Industry* 59.5 (2008), pp. 420–437. DOI: 10.1016/j.compind.2007.12.007 (cit. on pp. xxiv, xxxiv, 24, 25, 124, 125, 158, 281, 282, 336, 337, 381, 382).

[Van+08b] I. Vanderfeesten, H. A. Reijers, J. Mendling, W. M. P. van der Aalst, and J. Cardoso. "On a Quest for Good Process Models: The Cross-Connectivity Metric". In: *The 20th International Conference on Advanced Information Systems Engineering.* Berlin: Springer-Verlag, 2008, pp. 480–494 (cit. on pp. xxiii, 259, 334, 351, 362).

[Van+08c] J. Vanhatalo, H. Völzer, and J. Koehler. "The Refined Process Structure Tree". In: *Proceedings Business Process Management: 6th International Conference, BPM 2008, Milan, Italy, September 2-4, 2008.* Ed. by M. Dumas, M. Reichert, and M.-C. Shan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 100–115. DOI: 10.1007/978-3-540-85758-7_10 (cit. on p. 357).

[Vas+17] F. J. S. Vasconcellos, G. B. Landre, J. A. O. G. Cunha, J. L. Oliveira, R. A. Ferreira, and A. M. R. Vincenzi. "Approaches to Strategic Alignment of Software Process Improvement: A Systematic Literature Review". In: *Journal of Systems and Software* 123 (2017), pp. 45–63. DOI: 10.1016/j.jss.2016.09.030 (cit. on pp. 111, 113, 133).

[VW06] R. Vidgen and X. Wang. "From Business Process Management to Business Process Ecosystem". In: *Journal of Information Technology* 21.4 (2006), pp. 262–271. DOI: 10.1057/palgrave.jit.2000076 (cit. on p. 254).

[WK08] K. Wahler and J. M. Küster. "Predicting Coupling of Object-Centric Business Process Implementations". In: *Business Process Management: 6th International Conference, BPM 2008, Milan, Italy, September 2-4, 2008. Proceedings.* Ed. by M. Dumas, M. Reichert, and M.-C. Shan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 148–163. DOI: 10.1007/978-3-540-85758-7_13 (cit. on p. 23).

[WK05] J. Wang and A. Kumar. "A Framework for Document-Driven Workflow Systems". In: *Business Process Management: 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005. Proceedings.* Ed. by W. M. P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 285–301. DOI: 10.1007/11538394_19 (cit. on pp. 31–33).

[Wan02] Y. Wang. "On Cognitive Informatics". In: *Proceedings. First IEEE International Conference on Cognitive Informatics, 2002.* 2002, pp. 34–42. DOI: 10.1109/COGINF.2002.1039280 (cit. on p. 347).

[Web+09] B. Weber, H. A. Reijers, S. Zugal, and W. Wild. "The Declarative Approach to Business Process Execution: An Empirical Test". In: *Advanced Information Systems Engineering: 21st International Conference, CAiSE 2009.* Ed. by P. Eck, J. Gordijn, and R. Wieringa. Springer Berlin Heidelberg, 2009.

Chap. The Declarative Approach to Business Process Execution: An Empirical Test, pp. 470–485. DOI: 10.1007/978-3-642-02144-2_37 (cit. on p. 21).

[Web+11] B. Weber, M. Reichert, J. Mendling, and H. A. Reijers. “Refactoring Large Process Model Repositories”. In: *Computers in Industry* 62.5 (2011), pp. 467–486. DOI: 10.1016/j.compind.2010.12.012 (cit. on p. 351).

[Wes12] M. Weske. *Business Process Management: Concepts, Languages, Architectures.* Vol. 54. 2012, p. 403. DOI: 10.1007/978-3-540-73522-9 (cit. on pp. 11–14, 254, 280, 288).

[WS13] M. Westergaard and T. Slaats. “Mixing Paradigms for More Comprehensible Models”. In: *Business Process Management: 11th International Conference, BPM 2013, Beijing, China, August 26-30, 2013. Proceedings.* Ed. by F. Daniel, J. Wang, and B. Weber. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 283–290. DOI: 10.1007/978-3-642-40176-3_24 (cit. on pp. 20, 21).

[Wey88] E. J. Weyuker. “Evaluating Software Complexity Measures”. In: *IEEE Transactions on Software Engineering* 14.9 (1988), pp. 1357–1365. DOI: 10.1109/32.6178 (cit. on pp. 4, 107, 124, 131, 135, 144, 151, 152, 155–157, 196, 200, 311, 315, 343, 344).

[WfM12] WfMC. *XML Process Definition Language, v2.2.* Tech. rep. WfMC, 2012, p. 155 (cit. on pp. 16, 288).

[Whi09] M. White. *Case Management: Combining Knowledge With Process.* Tech. rep. BPTrends, 2009 (cit. on p. 28).

[Woh14] C. Wohlin. “Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering”. In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering.* EASE ’14. London, England, United Kingdom: ACM, 2014, pp. 381–3810. DOI: 10.1145/2601248.2601268 (cit. on p. 109).

[Woh16] C. Wohlin. “Second-generation Systematic Literature Studies Using Snowballing”. In: *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering.* EASE ’16. Limerick, Ireland: ACM, 2016, pp. 151–156. DOI: 10.1145/2915970.2916006 (cit. on p. 109).

[Woh+12] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 45–54. DOI: 10.1007/978-3-642-29044-2 (cit. on pp. 103, 112, 132, 133, 180, 181, 198).

[Yon+11] S. Yongchareon, Ch. Liu, and X. Zhao. "An Artifact-Centric View-Based Approach to Modeling Inter-organizational Business Processes". In: *Web Information System Engineering – WISE 2011: 12th International Conference, Sydney, Australia, October 13-14, 2011. Proceedings*. Ed. by A. Bouguettaya, M. Hauswirth, and L. Liu. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 273–281. DOI: 10.1007/978-3-642-24434-6_22 (cit. on p. 60).

[Yu+15] Y. Yu, X. Li, H. Liu, J. Mei, N. Mukhi, V. Ishakian, G. Xie, G. Lakshmanan, and M. A. Marin. "Case Analytics Workbench: Platform for Hybrid Process Model Creation and Evolution". In: *Business Process Management*. Ed. by H. R. Motahari-Nezhad, J. Recker, and M. Weidlich. Vol. 9253. Lecture Notes in Computer Science. Springer International Publishing, 2015, pp. 226–241. DOI: 10.1007/978-3-319-23063-4_16 (cit. on p. 21).

[Zac87] J. A. Zachman. "A Framework for Information Systems Architecture". In: *IBM Systems Journal* 26.3 (1987), pp. 276–292. DOI: 10.1147/sj.263.0276 (cit. on pp. 16, 18–20, 22–24, 29, 194).

[zur04] M. zur Muehlen. *Workflow-based Process Controlling. Foundation, Design, and Application of workflow-driven Process Information Systems*. Advances in Information Systems and Management Science. Logos Verlag Berlin, 2004 (cit. on pp. 13, 14).

[zH08] M. zur Muehlen and D. T. Ho. "Service Process Innovation: A Case Study of BPMN in Practice". In: *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*. Waikoloa, Big Island, Hawaii: IEEE, 2008, pp. 372–372. DOI: 10.1109/HICSS.2008.388 (cit. on pp. 66, 74, 75, 194, 199).

[zI10] M. zur Muehlen and M. Indulska. "Modeling Languages for Business Processes and Business Rules: A Representational Analysis". In: *Information Systems* 35.4 (2010), pp. 379–390. DOI: 10.1016/j.is.2009.02.006 (cit. on pp. 1, 2).

[zR08] M. zur Muehlen and J. C. Recker. "How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation". In: *20th International Conference on Advanced Information Systems Engineering*. Montpellier, France: Springer-Verlag, 2008, pp. 465–479 (cit. on pp. 66, 74, 75, 194, 199).

[zur+07] M. zur Muehlen, J. C. Recker, and M. Indulska. "Sometimes Less is More: Are Process Modeling Languages Overly Complex?" In: *2007 Eleventh International IEEE EDOC Conference Workshop*. IEEE, 2007, pp. 197–204. DOI: 10.1109/EDOCW.2007.30 (cit. on pp. 66, 74, 199).

[Zus93] H. Zuse. “Criteria for Program Comprehension Derived from Software Complexity Metrics”. In: *Program Comprehension, 1993. Proceedings., IEEE Second Workshop on.* 1993, pp. 8–16. DOI: 10.1109/WPC.1993.263911 (cit. on p. 25).

Glossary

Absolute Control-Flow Complexity

Is the metric $\texttt{CFC}_{abs}$ defined by Cardoso [Car08], see Table B.20 (Cardoso's 2008 metrics). This has the same meaning as CFC, which was defined by Cardoso [Car05b], see Table B.8. 333

Activity Complexity

Is the metric AC defined by Cardoso [Car05b], see Table B.8 (Cardoso's 2005 metrics). This has the same meaning as NA, which was defined by García et al. [Gar+03], see Table B.5. 315, 316

Activity Coupling

Is the metric CA defined by García et al. [Gar+03], see Table B.5 (García et al.'s 2003 metrics). 312, 313, 358

Activity Size of Process

Is the metric $\texttt{Size}_A$ defined by Antonini et al. [Ant+11], see Table B.36 (Antonini et al.'s 2011 metrics). 352, 358

Activity Type Count

Is the metric ATC defined by Lübke [LÏ5], see Table B.40 (Lübke's 2015 metrics). 356, 358

Adaptive Case Management (ACM)

Is a case management approach used by knowledge workers to create and modify processes at runtime [MS13; Swe13]. In ACM there is no distinction between design time and execution time [Swe13] . xvii, 28, 29, 178

Adaptive Document (ADoc)

Is a process modeling approach introduced by Kumaran et al. [Kum+03] based on three components: an information model, a collaboration that allows actors to interact with the adaptive document, and the event-driven behavior of the adaptive document . xvii, 29–32, 34, 39, 40

Aggregated Depth Fraction

Is the metric δ_S defined by Lassen and van der Aalst [Lv09], see Table B.25 (Lassen and van der Aalst's 2009 metrics). 339, 358

Artifact-Based Process Model

Is a business process model used for a business artifact-centric [Bha+07] approach . 3

Assign/invoke Ratio

Is the metric `AIR` defined by Held and Blochinger [HB09], see Table B.28 (Held and Blochinger's 2009 metrics). 343, 358

Average Activity Complexity

Is the metric $\bar{\mathtt{C}}_A$ defined by Tjaden [Tja99], see Table B.3 (Tjaden's 1999 metrics). 309, 358

Average Cognitive Complexity of Structured Activity

Is the metric `ACCSA` defined by Muketha et al. [Muk+10b], see Table B.30 (Muketha et al.'s 2010 metrics). 345, 358

Average Connector Cohesion

Is the metric `ACC` defined by Daneva et al. [Dan+96], see Table B.1 (Daneva et al.'s 1996 metrics). 307, 359

Average Degree of Connectors

Is the metric $\overline{\mathtt{d}_C}$ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). 330, 359

Average Degree of Place

Is the metric `ADP` defined by Mao [Mao10a], see Table B.31 (Mao's 2010 metrics). 346, 359

Average Degree of Transition

Is the metric `ADT` defined by Mao [Mao10a], see Table B.31 (Mao's 2010 metrics). 346, 359

Average Event Cohesion

Is the metric `AEC` defined by Daneva et al. [Dan+96], see Table B.1 (Daneva et al.'s 1996 metrics). 306, 359

Average Execution Path Complexity

Is the metric AEPC defined by Mao [Mao10a], see Table B.31 (Mao's 2010 metrics). 346, 359

Average Execution Path Complexity Based on Cognitive Informatics

Is the metric $AEPC_{CI}$ defined by Mao [Mao10b], see Table B.32 (Mao's 2010 second set of metrics). 347, 359

Average Function Cohesion

Is the metric AFC defined by Daneva et al. [Dan+96], see Table B.1 (Daneva et al.'s 1996 metrics). 306, 359

Average Gateway Degree

Is the metric AGD defined by La Rosa et al. [La +11b], see Table B.35 (La Rosa et al.'s 2011 metrics). This has the same meaning as $\overline{d_C}$, which was defined by Mendling [Men07], see Table B.16. 351

Average Length of Structured Activity

Is the metric ALSA defined by Muketha et al. [Muk+10b], see Table B.30 (Muketha et al.'s 2010 metrics). 345, 359

Average Path Length

Is the metric APL defined by Kreimeyer [Kre10], see Table B.33 (Kreimeyer's 2010 metrics). 348, 360

Basic Activity Distribution

Is the metric BAD defined by Lübke [Lï5], see Table B.40 (Lübke's 2015 metrics). 356, 360

Business Artifact (BA)

Is a concrete, identifiable, self-describing piece of information that can be used by a business person to run a business [NC03]. Is different from the object-oriented notion of object, because it is always an instance and does not provide encapsulation . xvii, 1–3, 5, 8, 11, 19, 23, 29–41, 47, 48, 59–61, 78, 87, 102, 193, 194, 198, 254

Business Artifacts with Lifecycle Services and Associations (BALSA)

Is a framework that uses four dimensions (i.e., artifact, life cycle, services or tasks, and associations) to describe artifact-centric approaches to business process models Multiple artifact-centric business process model approaches can be obtained by changing the four BALSA dimensions [Hul08] . xvii, 40, 48, 60, 193, 194, 198

Business Entities with Lifecycle (BEL)

This is synonymous with business artifacts [Hul+11b]. See BA . xvii, 29

Business Entity

This is an information entity (e.g., a payment, a claim, etc.) that provides context for business activities and processes and its behavior is modeled using a state machine [Kum+08]. A business entity is synonymous with BA . 33

Business Entity Lifecycle Analytics (BELA)

Is a methodology used to define models utilizing BAs [Hul+16; Str+08] . xvii, 34

Business Process

This comprises "a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly realize a business goal" [Wes12]. A business processes contains a set of planned activities that deal with known goals, but also encompass elements that deal with unpredictable and unknowable conditions [Lin+03] . 1, 2, 12–16, 18, 21–23, 25, 27, 29, 32, 34, 40, 42, 48, 66–68, 105, 151, 158, 178, 281, 309, 310, 313, 315, 322, 357

Business Process Execution Language (BPEL)

This defines a grammar for describing a business process based on Web Services interactions between the process and other Web Services [OAS07]. Is based on several XML and Web Services standard specifications, but it does not provide a graphical notation. It is also referred to as Web Services Business Process Execution Language or WS-BPEL . xvii, 2, 16, 21, 22, 34, 78, 315, 330, 331, 339, 342, 344, 354, 355

Business Process Management (BPM)

"[Includes] concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business processes" [Wes12]. Is a self-documenting technology in which processes are modeled and then executed in a business process manager server. The technology allows processes to be controlled, monitored, and changed in real time [VW06] . xvii, 2–5, 11–14, 20–23, 34, 36, 38, 104, 106, 109, 110, 114, 142, 178, 194, 197, 325, 327

Business Process Management System (BPMS)

Is a generic software system driven by process definitions that enact and execute business processes [Wes12] . xvii, 13, 14

Business Process Model

It, is an abstract description of a process that can be enacted by a human or a machine. Is described in a visual manner and represents the way that business representatives conduct the operation of a business [BR05]. Is also referred to as a process model . 1, 14, 24, 193, 198, 252, 255, 282, 316, 340, 354

Business Process Model and Notation (BPMN)

Is a standard graphical notation used to describe a business process. The notation is designed to facilitate communication between business users and technical developers in charge of implementing processes [OMG13] . xvii, 2, 4, 16, 21, 26, 34, 60, 65–68, 70, 74, 75, 78, 123, 178, 194, 195, 199, 262, 265, 316, 322, 331, 332, 334, 338, 343, 348, 351–355

Business Process Modeling

Is the act of creating and analyzing an organization's existing or planned business process models. Is often collaborative and uses business process models to facilitate communication [Eik+11] . 11, 13, 14, 24, 105, 106, 194, 198

Business Process Re-engineering (BPR)

Is a management practice that implements a radical redesign of organizational processes to gain improvements in cost, quality, and service [Ozc13]. Processes do not need to be automated to accomplish the goals of a business process re-engineering project . xvii, 12, 305, 308, 309

Business step (B-step)

Describes how a GSM instance reacts to an incoming external event [Dam+11] . xvii, 36, 46, 79, 90, 280

Cardoso's Control Flow Complexity

Is the metric `CFC` defined by Held and Blochinger [HB09], see Table B.28 (Held and Blochinger's 2009 metrics). This has the same meaning as `CFC`, which was defined by Cardoso [Car05b], see Table B.8. 342

Case Handling

Is an evolution of workflow technology where the key concept is the case and not the routing of activities [van+05]. Case handling focuses on what should be done to achieve a business goal and targets knowledge intensive processes [van+05] . 1, 27, 28, 36

Case Management

Is a particular type of business process that is collaborative and departs from traditional structured, sequential, predefined processes. Case management work depends more on human decision making and content than other processes [Ker08]. The central concept of case management is the case, rather than the activities or the routing . 1, 6, 11, 23, 26–29, 35, 38, 48, 67, 178

Case Management Model and Notation (CMMN)

Is a standard specification that intends to capture the essential elements that a case management product should provide [OMG14a] . xiii, xviii, 1, 3–6, 9, 11, 16, 21, 26, 29, 37–39, 48–50, 52–54, 57–63, 65–75, 77–79, 81–88, 90–93, 95–98, 100, 102–104, 106–108, 122, 125, 130, 132, 133, 135, 136, 138, 141, 142, 148, 151, 155–157, 159, 161, 163, 164, 171, 173, 177–179, 181, 190–200, 391, 394, 395

Case Management Process Modeling (CMPM)

This was a request for a proposal of a modeling notation for case management applications, which require the concept of a case file to maintain information about the case, including history, documents, and notes. "[Case] management processes include knowledge encoded as rules that provides guidance, prompts, constraints and planning support for the human decision-maker" [OMG09b] . xviii, 37

Case Worker

Is a participant in a case management process who oversees the case. The case worker is considered a knowledge worker. A case worker is expected to make decisions that affect the process applied to the particular case instance [Ker08] . 26, 27, 37, 38, 60–62, 87, 88, 93–96

CFC

Is the metric `CFC` defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as `CFC`, which was defined by Cardoso [Car05b], see Table B.8. 349

Checklist for Reporting Results of Internet E-Surveys (CHERRIES)

It is a checklist developed by the Journal of Medical Internet Research for authors to provide complete description of online surveys and give readers a good understanding of the sample selection [Eys12] . xvii, 158, 391

CNC

Is the metric `CNC` defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as $\texttt{CNC}_P$, which was defined by Latva-Koivisto [Lat01], see Table B.4. 349

Coefficient of Network Complexity

Is the metric `CNC` defined by Cardoso et al. [Car+06], see Table B.10 (Cardoso et al.'s 2006 metrics). This has the same meaning as $\texttt{CNC}_P$, which was defined by Latva-Koivisto [Lat01], see Table B.4. 321

Coefficient of Network Complexity (Kaimann)

Is the metric $\mathtt{CNC}_K$ defined by Latva-Koivisto [Lat01], see Table B.4 (Latva-Koivisto's 2001 metrics). 310, 360

Coefficient of Network Complexity (Pascoe)

Is the metric $\mathtt{CNC}_P$ defined by Latva-Koivisto [Lat01], see Table B.4 (Latva-Koivisto's 2001 metrics). 310, 360

Coefficient of Network Connectivity

Is the metric `CNC` defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). This has the same meaning as $\mathtt{CNC}_P$, which was defined by Latva-Koivisto [Lat01], see Table B.4. 329

Cognitive Activity Depth Arc Control Flow

Is the metric `CADAC` defined by Çoşkun [Çoş14], see Table B.39 (Çoşkun's 2014 metrics). 355, 360

Cognitive Complexity

Is the metric `CCBP` defined by Muketha et al. [Muk+10b], see Table B.30 (Muketha et al.'s 2010 metrics). 345, 360

Cognitive Complexity for YAWL

Is the metric $\mathtt{CC}_{\mathtt{YAWL}}$ defined by Gruhn and Laue [GL06a], see Table B.14 (Gruhn and Laue's 2006 second set of metrics). This has the same meaning as `CW`, which was defined by Gruhn and Laue [GL06b], see Table B.11. 325

Cognitive Weight

Is the metric `CW` defined by Gruhn and Laue [GL06b], see Table B.11 (Gruhn and Laue's 2006 metrics). 323, 325–327, 345, 360

Complex Event Processing (CEP)

Is a system that processes large flows of events to timeously detect situations of interest. The events can be simple events generated by a monitoring system or complex events created by aggregating simple events [Mar+14a]. The complex event system is driven by user-defined rules used to map the observed events to the phenomena of interest [Mar+14a]. It provides the basis for event-driven BPM [Kru+14] . xvii, 24

Complexity Index

Is the metric `CI` defined by Latva-Koivisto [Lat01], see Table B.4 (Latva-Koivisto's 2001 metrics). 310, 360

Complexity Metric

Is a calculation used in a business process model to measure the degree to which the processes are difficult to analyze, understand, or explain to others [Car06b] . 1–5, 24–26, 39, 69, 70, 103–110, 113, 114, 120–123, 125, 128–133, 135, 136, 141–144, 152, 155–161, 164, 166, 169, 174, 177, 191, 193–197, 199, 200, 305, 308–311, 315, 316, 320, 324, 325, 327, 331, 333, 335–338, 343, 345, 352–357, 395

Complexity of Process Integrated with Rules

Is the metric `Complexity` defined by Kluza [Klu15], see Table B.41 (Kluza's 2015 metrics). 356, 361

Connectivity Level Between Activities

Is the metric `CLA` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 361

Connectivity Level Between Pools

Is the metric `CLP` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 361

Connector Heterogeneity

Is the metric `CH` defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). 330, 361

Connector Mismatch

Is the metric `MM` defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). 330, 361

Content Management Interoperability Services (CMIS)

Is a standard programmatic interface that allows applications to interoperate with one or more content management systems or repositories [OAS13] . xvii, 9

Control Flow Complexity

Is the metric $\texttt{Complexity}_{CF}$ defined by Antonini et al. [Ant+11], see Table B.36 (Antonini et al.'s 2011 metrics). This has the same meaning as `CFC`, which was defined by Cardoso et al. [Car+06], see Table B.10. 352

Control-Flow Complexity

Is the metric `CFC` defined by Cardoso [Car05b], see Table B.8 (Cardoso's 2005 metrics). xiv, 315, 316, 321, 322, 324, 325, 327, 330, 333, 335, 343, 362

Control-Flow Complexity for BPEL Process

Is the metric $\texttt{CFC}^{\texttt{BPEL}}_{\texttt{Process}}$ defined by Cardoso [Car07b], see Table B.17 (Cardoso's 2007 metrics). 331, 362

Copy/assign Ratio

Is the metric `CAR` defined by Held and Blochinger [HB09], see Table B.28 (Held and Blochinger's 2009 metrics). 343, 362

Coupling

Is the metric `CP` defined by Vanderfeesten et al. [Van+07a], see Table B.19 (Vanderfeesten et al.'s 2007 metrics). 333, 362

Cross-Connectivity

Is the metric `CC` defined by Vanderfeesten et al. [Van+08b], see Table B.21 (Vanderfeesten et al.'s 2008 second set of metrics). 334, 362

Cycle

Is the metric `Cycle` defined by Mendling et al. [Men+06], see Table B.12 (Mendling et al.'s 2006 metrics). 323, 362

Cyclicity

Is the metric `CYC` defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). 330, 363

Cyclomatic Number

Is the metric `S` defined by Latva-Koivisto [Lat01], see Table B.4 (Latva-Koivisto's 2001 metrics). 310, 363

Data Flow Complexity

Is the metric $\texttt{Complexity}_{DF}$ defined by Antonini et al. [Ant+11], see Table B.36 (Antonini et al.'s 2011 metrics). 353, 363

Data Flow Intensity

Is the metric `DFI` defined by Held and Blochinger [HB09], see Table B.28 (Held and Blochinger's 2009 metrics). 343, 363

Data-Centric Dynamic Systems (DCDS)

Is a process approach where the focus is on both the data and the process controlling the data [Bag+13]. These systems consist of two layers, the data layer and the process layer [Cal+15] . xviii, 23, 36, 39, 78, 197

Declarative Process

See declarative process model . 1–3, 5, 20–22, 196, 197

Declarative Process Model

Is a process model that avoids the use of control-flow to determine a valid execution scenario [RV11]. The process is normally described using rules . 1–3, 21, 106, 108, 122, 130, 131, 135, 136, 193, 195, 199, 200, 260, 263

DECLARE

Is a constraint-based process modeling language that describes loosely structured processes [Pes+07]. Is a declarative process modeling language . 2, 21

Degree of Parallelism

Is the metric `DOP` defined by Held and Blochinger [HB09], see Table B.28 (Held and Blochinger's 2009 metrics). 342, 363

Dehmer's Graph Entropy

Is the metric $\mathtt{I}_{fV}$ defined by Borgert et al. [Bor+09a], see Table B.27 (Borgert et al.'s 2009 metrics). 341, 363

Density of a Workflow-Net

Is the metric $\mathtt{d}_w$ defined by Mendling [Men06], see Table B.13 (Mendling's 2006 second set of metrics). 324, 363

Density of a YWAL Model

Is the metric $\mathtt{d}_y$ defined by Mendling [Men06], see Table B.13 (Mendling's 2006 second set of metrics). 324, 363

Density of an EPC

Is the metric $\mathtt{d}_e$ defined by Mendling [Men06], see Table B.13 (Mendling's 2006 second set of metrics). 324, 363

Density of the Process Graph

Is the metric Δ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). 330, 364

Depth

Is the metric `Depth` defined by La Rosa et al. [La +11b], see Table B.35 (La Rosa et al.'s 2011 metrics). 351, 364

Deterministic Finite State Machine (DFSM)

Is a machine with memory that executes steps by moving from state to state based on external input, but when in a state an input can only transition it to one and only one state [Hop+01; Sav08] . xviii, 77, 84, 87, 88, 90, 392

Diameter

Is the metric `diam` defined by La Rosa et al. [La +11b], see Table B.35 (La Rosa et al.'s 2011 metrics). This has the same meaning as `diam`, which was defined by Mendling [Men07], see Table B.16. 351

Diameter

Is the metric `diam` defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). 328, 364

Different Modeling Concepts

Is the metric `DMC` defined by La Rosa et al. [La +11b], see Table B.35 (La Rosa et al.'s 2011 metrics). 352, 364

Durfee Square Metric

Is the metric `DSM` defined by Kluza and Nalepa [KN12], see Table B.37 (Kluza and Nalepa's 2012 metrics). 354, 364

Dynamic Case Management

Is an information intensive process type that is driven by external events and is highly collaborative and structured with a case folder collecting all of the required information to solve a particular case [Le +09] . 28, 29

Event-Condition-Action (ECA)

Is a rule that executes an action when an event is detected and a condition or set of conditions evaluates to *true* [Can+14] . xviii, 36, 41, 49, 60, 280

Event-driven Process Chain (EPC)

Is a process modeling technique that uses functions, events, and logical connectors as building blocks to model a process [van99]. Is a modeling language commonly used to create business process models [Rie+16] . xviii, 4, 16, 21, 24, 65–68, 70, 74, 75, 123, 194, 195, 199, 305, 307, 320, 322, 323, 326, 327, 331, 332, 334, 335, 338, 351

Extended Cardoso

Is the metric `ECaM` defined by Lassen and van der Aalst [Lv09], see Table B.25 (Lassen and van der Aalst's 2009 metrics). 338, 364

Extended Cyclomatic

Is the metric `ECyM` defined by Lassen and van der Aalst [Lv09], see Table B.25 (Lassen and van der Aalst's 2009 metrics). 338, 364

Extended Measure

Is the metric `EM` defined by Sobrinho [Sob99], see Table B.2 (Sobrinho's 1999 metrics). 308, 364

Extension Activity Count

Is the metric `EAC` defined by Lübke [LÏ5], see Table B.40 (Lübke's 2015 metrics). 356, 364

Extension Activity Distribution

Is the metric `EDB` defined by Lübke [LÏ5], see Table B.40 (Lübke's 2015 metrics). 356, 365

Fan-In / Fan-Out

Is the metric $\mathtt{F}_i\mathtt{F}_o$ defined by Gruhn and Laue [GL06b], see Table B.11 (Gruhn and Laue's 2006 metrics). 323, 327, 365

Flexibility

Is the metric `Flexibility` defined by Tjaden [Tja99], see Table B.3 (Tjaden's 1999 metrics). 309, 365

Global Ripple Effect Measure

Is the metric `GREM` defined by Sobrinho [Sob99], see Table B.2 (Sobrinho's 1999 metrics). 308, 365

Graphical Process Model (GPM)

Is a notation, based on formal graph theory, used to represent a business process model [Swa07]. Several types of GPMs exist, including UML and BPMN [Swa07] . xviii, 1

Guard-Stage-Milestone (GSM)

Is a declarative approach used to define the life cycle of business artifacts, using guards that are conditions that enable entry into a stage. The stage contains one or more activities needed to achieve a milestone [Hul+11b] . xviii, 2–5, 21–23, 34, 36, 37, 39–49, 52, 53, 59–63, 77–82, 84–98, 100–102, 193–195, 197–199, 255, 392

Height of Hierarchy

Is the metric `HH` defined by Kreimeyer [Kre10], see Table B.33 (Kreimeyer's 2010 metrics). 348, 365

Henry and Kafura Metric

Is the metric `HKM` defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 349, 365

Hybrid Process Model

Is a process model that combines the flexibility of the declarative process models with the structure of imperative process models in an attempt to achieve a balance . 1, 20–22

IBM Global Financing (IGF)

Is a division of IBM and one of the largest IT financiers in the world with more than 25 years of experience and 125,000 customers in 50 countries [Cha+09] . xviii, 47, 83

Imperative Process

See procedural process model . 5, 16, 20, 21, 67, 197

Imperative Process Model

See procedural process model . 2, 3, 21, 122, 130, 131, 136, 160, 193, 199, 263

Information Flow Complexity

Is the metric `IF` defined by Sun and Hou [SH14], see Table B.38 (Sun and Hou's 2014 metrics). This has the same meaning as `IF4BP`, which was defined by Muketha et al. [Muk+10b], see Table B.30. 355

Information Flow Complexity

Is the metric `IF4BP` defined by Muketha et al. [Muk+10b], see Table B.30 (Muketha et al.'s 2010 metrics). 344, 365

Information System

Is a system that brings data, computers, processes, and people together to manage the information of an organization [Pfa03] . 11–14, 26, 29

Information Technology (IT)

Is a generic term used for all aspects of computer networking and information systems technology [Pfa03] . xviii, 2, 11–14, 26, 29, 30, 47, 348

Integrated Definition for Function Modeling (IDEF)

Is a family of modeling and descriptive languages [May+92] . xviii, 29

Integration

Is the metric `Integration` defined by Tjaden [Tja99], see Table B.3 (Tjaden's 1999 metrics). 309, 365

Join Complexity

Is the metric `JC` defined by Mendling et al. [Men+06], see Table B.12 (Mendling et al.'s 2006 metrics). 323, 365

Join-Split-Ratio

Is the metric `JSR` defined by Mendling et al. [Men+06], see Table B.12 (Mendling et al.'s 2006 metrics). 324, 365

Knowledge Intensive Processes (KiP)

Is a human-centered process used by knowledge workers that combines the fields of knowledge management and business process management [di +14; MF11]. It was first introduced by Marjanovic and Freeze [MF11] as a theoretical framework that combines research work in knowledge management and business process management. Is also referred to as knowledge intensive business process . xviii, 6, 19, 22, 23, 29, 75

Log-Based Complexity

Is the metric $\mathtt{LBC}_T$ defined by Cardoso [Car07a], see Table B.18 (Cardoso's 2007 second set of metrics). 332, 365

Maximum Depth of all Nodes

Is the metric Λ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). 329, 366

Maximum Nesting Depth

Is the metric `MaxND` defined by Gruhn and Laue [GL06b], see Table B.11 (Gruhn and Laue's 2006 metrics). 322, 366

Maximun Degree of a Connector

Is the metric $\widehat{\mathtt{d}_C}$ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). 329, 366

McCabe's Cyclomatic Number

Is the metric `C` defined by Borgert et al. [Bor+09a], see Table B.27 (Borgert et al.'s 2009 metrics). This has the same meaning as `ECyM`, which was defined by Lassen and van der Aalst [Lv09], see Table B.25. 341

McCabe's Cyclomatic Number

Is the metric `MCC` defined by Cardoso et al. [Car+06], see Table B.10 (Cardoso et al.'s 2006 metrics). This has the same meaning as `S`, which was defined by Latva-Koivisto [Lat01], see Table B.4. 321

McCabe's Cyclomatic Number

Is the metric `MCC` defined by Held and Blochinger [HB09], see Table B.28 (Held and Blochinger's 2009 metrics). This has the same meaning as `S`, which was defined by Latva-Koivisto [Lat01], see Table B.4. 342

McCabe's Cyclomatic Number

Is the metric `CC` defined by Mao [Mao10a], see Table B.31 (Mao's 2010 metrics). This has the same meaning as `S`, which was defined by Latva-Koivisto [Lat01], see Table B.4. 346

Mean Nesting Depth

Is the metric `MeanND` defined by Gruhn and Laue [GL06b], see Table B.11 (Gruhn and Laue's 2006 metrics). 322, 366

Method Complexity

Is an approach used to perform development in a systematic way and is composed of techniques or notations [Bri96]. Method complexity is used as a way of evaluating the complexity of a particular method [RB96]. Common methods include BPMN and UML which are composed of techniques that correspond to their different diagrams . xiii, 3–5, 26, 65–70, 73–75, 135, 194, 197–199

Model Complexity

Refers to the level of difficulty in analyzing, understanding, or explaining processes [Car06b] . 4, 5, 164, 191–193, 196, 197, 200, 311, 351, 353

Model Size

Is the metric `MS` defined by La Rosa et al. [La +11b], see Table B.35 (La Rosa et al.'s 2011 metrics). This has the same meaning as `F`, which was defined by Mendling et al. [Men+06], see Table B.12. 351

Model-Driven Business Transformation (MDBT)

Is a business transformation methodology, developed by IBM Research that describes the life cycle of a business problem to the technological implementation [Kum+07] . xviii, 34

Modularization Measure

Is the metric `MM` defined by Sobrinho [Sob99], see Table B.2 (Sobrinho's 1999 metrics). 308, 366

Modules Overhead

Is the metric `MO` defined by La Rosa et al. [La +11b], see Table B.35 (La Rosa et al.'s 2011 metrics). 351, 366

Nesting Depth

Is the metric ND defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as MaxND, which was defined by Gruhn and Laue [GL06b], see Table B.11. 349

Number Complex Gateways

Is the metric NCG defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as NCD, which was defined by Rolón et al. [Rol+06b], see Table B.9. 349

Number Gateway Data Based Exclusive

Is the metric NGDE defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 349, 366

Number Gateway Data Based Inclusive

Is the metric NGDI defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 349, 366

Number Gateway Event Based Exclusive

Is the metric NGEE defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 349, 366

Number of Activities

Is the metric NA defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as NA, which was defined by García et al. [Gar+03], see Table B.5. 349

Number of Activities

Is the metric NOA defined by Cardoso et al. [Car+06], see Table B.10 (Cardoso et al.'s 2006 metrics). This has the same meaning as NA, which was defined by García et al. [Gar+03], see Table B.5. 321, 345

Number of Activities

Is the metric NA defined by Gruhn and Laue [GL06b], see Table B.11 (Gruhn and Laue's 2006 metrics). This has the same meaning as NA, which was defined by García et al. [Gar+03], see Table B.5. 322

Number of Activities

Is the metric NOA defined by Held and Blochinger [HB09], see Table B.28 (Held and Blochinger's 2009 metrics). This has the same meaning as NA, which was defined by García et al. [Gar+03], see Table B.5. 342

Number of Activities

Is the metric NA defined by García et al. [Gar+03], see Table B.5 (García et al.'s 2003 metrics). 312, 313, 367

Number of Activities and Control

Is the metric NOAC defined by Held and Blochinger [HB09], see Table B.28 (Held and Blochinger's 2009 metrics). This has the same meaning as NOAC, which was defined by Cardoso et al. [Car+06], see Table B.10. 342

Number of Activities and Control Flow Elements

Is the metric NOAC defined by Cardoso et al. [Car+06], see Table B.10 (Cardoso et al.'s 2006 metrics). 321, 367

Number of Activities Joins and Splits

Is the metric NOAJS defined by Cardoso et al. [Car+06], see Table B.10 (Cardoso et al.'s 2006 metrics). 321, 368

Number of Activities, Control Structures, and Copy

Is the metric NOACC defined by Held and Blochinger [HB09], see Table B.28 (Held and Blochinger's 2009 metrics). 342, 367

Number of AND Connectors

Is the metric $\mathtt{S}_{C_{and}}$ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). 328, 368

Number of AND Joins

Is the metric $\mathtt{S}_{j_{and}}$ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). This has the same meaning as $\mathtt{AND}_j$, which was defined by Mendling et al. [Men+06], see Table B.12. 328

Number of AND Joins

Is the metric $\mathtt{AND}_j$ defined by Mendling et al. [Men+06], see Table B.12 (Mendling et al.'s 2006 metrics). 323, 368

Number of AND Splits

Is the metric $\mathtt{S}_{S_{and}}$ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). This has the same meaning as $\mathtt{AND}_s$, which was defined by Mendling et al. [Men+06], see Table B.12. 328

Number of AND Splits

Is the metric $\mathtt{AND}_s$ defined by Mendling et al. [Men+06], see Table B.12 (Mendling et al.'s 2006 metrics). 323, 368

Number of Arcs

Is the metric `S`$_A$ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). This has the same meaning as `A`, which was defined by Mendling et al. [Men+06], see Table B.12. 328, 330

Number of Arcs

Is the metric `A` defined by Mendling et al. [Men+06], see Table B.12 (Mendling et al.'s 2006 metrics). 310, 312, 323, 368

Number of Artifacts

Is the metric `NAf` defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 349, 368

Number of Associations

Is the metric `NAS` defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 349, 368

Number of Basic Activities

Is the metric `NOBA` defined by Muketha et al. [Muk+10b], see Table B.30 (Muketha et al.'s 2010 metrics). This has the same meaning as `NT`, which was defined by Rolón et al. [Rol+06b], see Table B.9. 344

Number of Classes

Is the metric `NCl` defined by Kreimeyer [Kre10], see Table B.33 (Kreimeyer's 2010 metrics). 348, 368

Number of Collapsed Ad-Hoc Sub-Process

Is the metric `NCSA` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 368

Number of Collapsed Compensation Sub-Process

Is the metric `NCSC` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 369

Number of Collapsed Looping Sub-Process

Is the metric `NCSL` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 369

Number of Collapsed Multiple Instance Sub-Process

Is the metric `NCSMI` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 369

Number of Collapsed Sub-Process

Is the metric NCS defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 369

Number of Compensation Tasks

Is the metric NTC defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 369

Number of Complex Decision/merge

Is the metric NCD defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 369

Number of Connectors

Is the metric NC defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as NDWP, which was defined by García et al. [Gar+04a], see Table B.6. 349

Number of Connectors

Is the metric S_C defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). This has the same meaning as NDWP, which was defined by García et al. [Gar+04a], see Table B.6. 325, 328

Number of Data Objects

Is the metric NDO defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as NWP, which was defined by García et al. [Gar+03], see Table B.5. 349

Number of Decisions

Is the metric DC defined by Held and Blochinger [HB09], see Table B.28 (Held and Blochinger's 2009 metrics). This has the same meaning as TNG, which was defined by Rolón et al. [Rol+06b], see Table B.9. 343

Number of Dependences

Is the metric NDWP defined by García et al. [Gar+04a], see Table B.6 (García et al.'s 2004 metrics). 313, 370

Number of Different Objects in all Possible Paths

Is the metric NDOp defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 349, 370

Number of Domains

Is the metric `NDos` defined by Kreimeyer [Kre10], see Table B.33 (Kreimeyer's 2010 metrics). 348, 370

Number of Edges

Is the metric `NE` defined by Kreimeyer [Kre10], see Table B.33 (Kreimeyer's 2010 metrics). This has the same meaning as `A`, which was defined by Mendling et al. [Men+06], see Table B.12. 348

Number of Edges Per Node

Is the metric `NEN` defined by Kreimeyer [Kre10], see Table B.33 (Kreimeyer's 2010 metrics). 348, 370

Number of End Cancel Event

Is the metric `NECaE` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 370

Number of End Compensation Event

Is the metric `NECoE` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 370

Number of End Error Event

Is the metric `NEEE` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 371

Number of End Events

Is the metric `NEE` defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as `TNEE`, which was defined by Rolón et al. [Rol+06b], see Table B.9. 349

Number of End Events

Is the metric $\mathtt{S}_{E_E}$ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). This has the same meaning as `TNEE`, which was defined by Rolón et al. [Rol+06b], see Table B.9. 328

Number of End Events

Is the metric $\mathtt{E}_{end}$ defined by Mendling et al. [Men+06], see Table B.12 (Mendling et al.'s 2006 metrics). This has the same meaning as `TNEE`, which was defined by Rolón et al. [Rol+06b], see Table B.9. 323

Number of End Link Event

Is the metric NELE defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 371

Number of End Message Event

Is the metric NEMsE defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 371

Number of End Multiple Event

Is the metric NEMuE defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 371

Number of End None Event

Is the metric NENE defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 371

Number of End Terminate Event

Is the metric NETE defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 372

Number of Events

Is the metric TNE defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as TNE, which was defined by Rolón et al. [Rol+06b], see Table B.9. 350

Number of Events

Is the metric S_E defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). This has the same meaning as TNE, which was defined by Rolón et al. [Rol+06b], see Table B.9. 328

Number of Exclusive Decision/merge Data-Based

Is the metric NEDDB defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 372

Number of Exclusive Decision/merge Event-Based

Is the metric NEDEB defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 372

Number of Feedbacks

Is the metric NF defined by Kreimeyer [Kre10], see Table B.33 (Kreimeyer's 2010 metrics). 348, 372

Number of Flow Objects

Is the metric NFO defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 349, 372

Number of Flow Objects in Smallest Path

Is the metric NFOSP defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 349, 373

Number of Functions

Is the metric S_F defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). This has the same meaning as F, which was defined by Mendling et al. [Men+06], see Table B.12. 328

Number of Functions

Is the metric F defined by Mendling et al. [Men+06], see Table B.12 (Mendling et al.'s 2006 metrics). 323, 373

Number of Gates (Fan-In + Fan-Out)

Is the metric NGa defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 349, 373

Number of Gateways

Is the metric TNG defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as TNG, which was defined by Rolón et al. [Rol+06b], see Table B.9. 350

Number of Groups

Is the metric NG defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 349, 373

Number of Handles

Is the metric NH defined by Gruhn and Laue [GL06b], see Table B.11 (Gruhn and Laue's 2006 metrics). 322, 373

Number of Incusive Decision/merge

Is the metric NID defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 373

Number of Independent Sets

Is the metric NIS defined by Kreimeyer [Kre10], see Table B.33 (Kreimeyer's 2010 metrics). 348, 373

Number of Input Data Objects

Is the metric `NDOIn` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 374

Number of Input Dependences

Is the metric `NDWPIn` defined by García et al. [Gar+04a], see Table B.6 (García et al.'s 2004 metrics). 313, 374

Number of Input Gates (Fan-In)

Is the metric `NIG` defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 349, 374

Number of Intermediate Cancel Event

Is the metric `NICaE` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 374

Number of Intermediate Compensation Event

Is the metric `NICoE` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 374

Number of Intermediate Error Event

Is the metric `NIEE` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 374

Number of Intermediate Events

Is the metric `NIE` defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as `NTIE`, which was defined by Rolón et al. [Rol+06b], see Table B.9. 349

Number of Intermediate Link Event

Is the metric `NILE` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 375

Number of Intermediate Message Event

Is the metric `NIMsE` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 375

Number of Intermediate Multiple Event

Is the metric `NIMuE` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 375

Number of Intermediate None Event

Is the metric NINE defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 375

Number of Intermediate Rule Event

Is the metric NIRE defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 375

Number of Intermediate Timer Event

Is the metric NITE defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 375

Number of Internal Events

Is the metric E_{int} defined by Mendling et al. [Men+06], see Table B.12 (Mendling et al.'s 2006 metrics). 323, 375

Number of Lanes

Is the metric NL defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as NPR, which was defined by García et al. [Gar+03], see Table B.5. 350

Number of Lanes

Is the metric NL defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). This has the same meaning as NPR, which was defined by García et al. [Gar+03], see Table B.5. 318

Number of Looping Tasks

Is the metric NTL defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 375

Number of Message Flows

Is the metric NMF defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as NMF, which was defined by Rolón et al. [Rol+06b], see Table B.9. 350

Number of Message Flows

Is the metric NMF defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 376

Number of Multiple Instances Tasks

Is the metric NTMI defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 376

Number of Nodes

Is the metric NN defined by Kreimeyer [Kre10], see Table B.33 (Kreimeyer's 2010 metrics). This has the same meaning as NA, which was defined by García et al. [Gar+03], see Table B.5. 348

Number of Nodes

Is the metric S_N defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). This has the same meaning as NA, which was defined by García et al. [Gar+03], see Table B.5. 310, 328, 330

Number of Objects in Biggest Path

Is the metric NOBP defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 350, 376

Number of OR Connectors

Is the metric $S_{C_{or}}$ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). 328, 376

Number of OR Joins

Is the metric $S_{j_{or}}$ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). This has the same meaning as OR_j, which was defined by Mendling et al. [Men+06], see Table B.12. 328

Number of OR Joins

Is the metric OR_j defined by Mendling et al. [Men+06], see Table B.12 (Mendling et al.'s 2006 metrics). 324, 376

Number of OR Splits

Is the metric $S_{S_{or}}$ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). This has the same meaning as OR_s, which was defined by Mendling et al. [Men+06], see Table B.12. 328

Number of OR Splits

Is the metric OR_s defined by Mendling et al. [Men+06], see Table B.12 (Mendling et al.'s 2006 metrics). 324, 376

Number of Output Data Objects

Is the metric NDOOut defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 376

Number of Output Dependences

Is the metric NDWPOut defined by García et al. [Gar+04a], see Table B.6 (García et al.'s 2004 metrics). 313, 377

Number of Output Gates(Fan-Out)

Is the metric NOG defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 350, 377

Number of Parallel Fork/join

Is the metric NPF defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 377

Number of Places

Is the metric N_P defined by Mao [Mao10a], see Table B.31 (Mao's 2010 metrics). 346, 377

Number of Pools

Is the metric NP defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as NP, which was defined by Rolón et al. [Rol+06b], see Table B.9. 350

Number of Pools

Is the metric NP defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 377

Number of Possible Paths

Is the metric NPP defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 350, 377

Number of Precedence Dependences Between Activities

Is the metric NDRA defined by García et al. [Gar+03], see Table B.5 (García et al.'s 2003 metrics). 312, 377

Number of Roles

Is the metric NPR defined by García et al. [Gar+03], see Table B.5 (García et al.'s 2003 metrics). 312, 313, 378

Number of Sequence Flows

Is the metric NSF defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as NDWP, which was defined by García et al. [Gar+04a], see Table B.6. 350

Number of Sequence Flows

Is the metric NSF defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). This has the same meaning as NDWP, which was defined by García et al. [Gar+04a], see Table B.6. 318

Number of Sequence Flows Between Activities

Is the metric NSFA defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 378

Number of Sequence Flows From Events

Is the metric NSFE defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 378

Number of Sequence Flows From Gateways

Is the metric NSFG defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 318, 378

Number of Services

Is the metric $\mathtt{N}_S$ defined by Mao [Mao10a], see Table B.31 (Mao's 2010 metrics). 346, 378

Number of Simple Tasks

Is the metric NT defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 380

Number of Start Events

Is the metric NSE defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as NTSE, which was defined by Rolón et al. [Rol+06b], see Table B.9. 350

Number of Start Events

Is the metric $\mathtt{S}_{E_S}$ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). This has the same meaning as NTSE, which was defined by Rolón et al. [Rol+06b], see Table B.9. 328

Number of Start Events

Is the metric $\mathtt{E}_{start}$ defined by Mendling et al. [Men+06], see Table B.12 (Mendling et al.'s 2006 metrics). This has the same meaning as NTSE, which was defined by Rolón et al. [Rol+06b], see Table B.9. 323

Number of Start Link Event

Is the metric NSLE defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 379

Number of Start Message Event

Is the metric NSMsE defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 379

Number of Start Multiple Event

Is the metric NSMuE defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 379

Number of Start None Event

Is the metric NSNE defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 379

Number of Start Rule Event

Is the metric NSRE defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 379

Number of Start Timer Event

Is the metric NSTE defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 317, 380

Number of Steps (Tasks)

Is the metric NSTP defined by García et al. [Gar+03], see Table B.5 (García et al.'s 2003 metrics). 312, 380

Number of Structured Activities

Is the metric NOSA defined by Muketha et al. [Muk+10b], see Table B.30 (Muketha et al.'s 2010 metrics). This has the same meaning as TNG, which was defined by Rolón et al. [Rol+06b], see Table B.9. 344

Number of Sub-Processes

Is the metric NSP defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 350, 380

Number of Swimlanes

Is the metric NSL defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as NP, which was defined by Rolón et al. [Rol+06b], see Table B.9. 350

Number of Tasks

Is the metric `TNT` defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as `TNT`, which was defined by García et al. [Gar+03], see Table B.5. 350

Number of Text Annotations

Is the metric `NTA` defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 350, 380

Number of Transitions

Is the metric $\mathtt{N}_T$ defined by Mao [Mao10a], see Table B.31 (Mao's 2010 metrics). 346, 380

Number of Trees in a Graph

Is the metric `T` defined by Latva-Koivisto [Lat01], see Table B.4 (Latva-Koivisto's 2001 metrics). 311, 380

Number of Unconnected Nodes

Is the metric `NUN` defined by Kreimeyer [Kre10], see Table B.33 (Kreimeyer's 2010 metrics). 348, 380

Number of Work Products

Is the metric `NWP` defined by García et al. [Gar+03], see Table B.5 (García et al.'s 2003 metrics). 312, 313, 381

Number of XOR Connectors

Is the metric $\mathtt{S}_{C_{xor}}$ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). 328, 381

Number of XOR Joins

Is the metric $\mathtt{S}_{j_{xor}}$ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). This has the same meaning as $\mathtt{XOR}_j$, which was defined by Mendling et al. [Men+06], see Table B.12. 328

Number of XOR Joins

Is the metric $\mathtt{XOR}_j$ defined by Mendling et al. [Men+06], see Table B.12 (Mendling et al.'s 2006 metrics). 324, 381

Number of XOR Splits

Is the metric $\mathtt{S}_{S_{xor}}$ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). This has the same meaning as $\mathtt{XOR}_s$, which was defined by Mendling et al. [Men+06], see Table B.12. 328

Number of XOR Splits

Is the metric $\texttt{XOR}_s$ defined by Mendling et al. [Men+06], see Table B.12 (Mendling et al.'s 2006 metrics). 324, 381

Number Parallel Gateways

Is the metric `NPG` defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). This has the same meaning as `NPF`, which was defined by Rolón et al. [Rol+06b], see Table B.9. 350

Object Constraint Language (OCL)

Is a formal language used to specify expressions in UML in a declarative way in order to describe constraints and query expressions [OMG14b] . xviii, 45, 62, 79, 97

Object Management Group (OMG)

Is an international consortium that was founded in 1989, which is dedicated to the development of software technology standards [Gro17] . xviii, 1, 37, 48, 66, 67, 199

Object, Property, Relationship, Role (OPRR)

Is a method modeling language that can be used to describe method meta-models where objects, properties, relationships, and roles are used as meta-types [RB96] . xviii, 69, 70, 75, 195, 199

Operations Specification (OpS)

Is a data-centric process approach, where there is no distinction between the control-flow and the data-flow [NC03] . xviii, 29–33, 39

Perfect Square Metric

Is the metric `PSM` defined by Kluza and Nalepa [KN12], see Table B.37 (Kluza and Nalepa's 2012 metrics). 354, 381

Petri Net

Is a bipartite graph consisting of two types of nodes (places and transitions) and directed arcs such that an arc never connects two places or two transitions [Wes12]. The dynamic of the system is modeled by tokens that reside in places and change their position according to firing rules [Wes12] . 12, 21, 32, 78, 104, 288, 289, 322, 334, 346

Prerequisite-Antecedent-Consequent (PAC)

Is a variation of Event-Condition-Action (ECA) rules used to describe the Business step (B-step) behavior and constraints [Dam+11; de +15b; Hul+11a] . xviii, 46, 90

Procedural Process Model

Is a process model that is explicit about how processes should proceed and so focuses on the control-flow of the process [RV11] . 21, 263

Process

See business process . xi, 1–5, 11–29, 31–36, 38, 40, 46, 47, 56, 58, 66–68, 70, 72, 74, 104–106, 110–114, 124, 143, 158, 163, 175, 196, 197, 305–313, 315, 317, 320–322, 324, 329–333, 337, 339, 340, 343–348, 351–357

Process Cohesion

Is the metric `ch` defined by Vanderfeesten et al. [Van+08a], see Table B.24 (Vanderfeesten et al.'s 2008 metrics). This has the same meaning as `c`, which was defined by Reijers and Vanderfeesten [RV04], see Table B.7. 337

Process Cohesion

Is the metric `c` defined by Reijers and Vanderfeesten [RV04], see Table B.7 (Reijers and Vanderfeesten's 2004 metrics). 314, 381

Process Complexity

Is a process metric designed to capture the degree to which a processes is difficult to analyze, understand, or explain [Car06b] . 25, 105, 315

Process Complexity

Is the metric `PC` defined by Abreu et al. [Abr+10], see Table B.34 (Abreu et al.'s 2010 metrics). 350, 381

Process Context Independency Metric

Is the metric $\mathtt{P}_{CIM}$ defined by Khoshkbarforoushha et al. [Kho+09], see Table B.26 (Khoshkbarforoushha et al.'s 2009 metrics). 340, 382

Process Coupling

Is the metric `cp` defined by Vanderfeesten et al. [Van+08a], see Table B.24 (Vanderfeesten et al.'s 2008 metrics). This has the same meaning as `k`, which was defined by Reijers and Vanderfeesten [RV04], see Table B.7. 337

Process Coupling

Is the metric `k` defined by Reijers and Vanderfeesten [RV04], see Table B.7 (Reijers and Vanderfeesten's 2004 metrics). 315, 382

Process Coupling Metric

Is the metric $\mathtt{P}_{CM}$ defined by Khoshkbarforoushha et al. [Kho+09], see Table B.26 (Khoshkbarforoushha et al.'s 2009 metrics). 340, 382

Process Coupling/cohesion Ratio

Is the metric ρ defined by Vanderfeesten et al. [Van+08a], see Table B.24 (Vanderfeesten et al.'s 2008 metrics). This has the same meaning as ρ, which was defined by Reijers and Vanderfeesten [RV04], see Table B.7. 337

Process Coupling/cohesion Ratio

Is the metric ρ defined by Reijers and Vanderfeesten [RV04], see Table B.7 (Reijers and Vanderfeesten's 2004 metrics). 315, 382

Process Difficulty

Is the metric `D` defined by Cardoso et al. [Car+06], see Table B.10 (Cardoso et al.'s 2006 metrics). 321, 382

Process Length

Is the metric `N` defined by Cardoso et al. [Car+06], see Table B.10 (Cardoso et al.'s 2006 metrics). 321, 382

Process Model

See business process model . 1, 2, 5, 11, 13–15, 18–20, 22–27, 32, 35, 36, 38, 65–67, 78, 103–110, 114, 120–125, 128–133, 135, 136, 155, 157, 159–161, 164, 166–169, 178, 193–199, 281, 309, 311, 315, 320, 322–326, 331, 334–336, 340, 341, 343, 351, 352, 354, 356, 357

Process Modeling Notation

Is a modeling notation composed of a set of graphical symbols used for the visualization of process elements [Dum+13] . 15, 18, 65, 68, 74, 78, 130, 193, 311, 315, 338

Process Volume

Is the metric `V` defined by Cardoso et al. [Car+06], see Table B.10 (Cardoso et al.'s 2006 metrics). 321, 382

Production Case Management (PCM)

Is a case management approach where the processes is created by developers for use by knowledge workers [MS13; Swe13]. In PCM there is a clear separation between design time and execution time [Swe13] . xviii, 28, 29

Randi'c's Connectivity Index

Is the metric `R` defined by Borgert et al. [Bor+09a], see Table B.27 (Borgert et al.'s 2009 metrics). 341, 382

Rate of Input Data Object Over the Total of Data Objects

Is the metric `PDOPIn` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). This has the same meaning as `RDWPIn`, which was defined by García et al. [Gar+04a], see Table B.6. 319

Rate of Output Data Object Over the Total of Data Objects

Is the metric `PDOPOut` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). This has the same meaning as `RDWPOut`, which was defined by García et al. [Gar+04a], see Table B.6. 319

Rate of Output Data Object Over the Total of Tasks

Is the metric `PDOTOut` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). This has the same meaning as `RWPA`, which was defined by García et al. [Gar+03], see Table B.5. 319

Rate of Pools and Lanes Over the Total of Tasks

Is the metric `PLT` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). This has the same meaning as `RPRA`, which was defined by García et al. [Gar+03], see Table B.5. 319

Ratio Between Input Dependencies and Number of Dependencies

Is the metric `RDWPIn` defined by García et al. [Gar+04a], see Table B.6 (García et al.'s 2004 metrics). 313, 383

Ratio Between Output Dependencies and the Number of Dependencies

Is the metric `RDWPOut` defined by García et al. [Gar+04a], see Table B.6 (García et al.'s 2004 metrics). 313, 383

Ratio of Process Roles and Activities

Is the metric `RPRA` defined by García et al. [Gar+03], see Table B.5 (García et al.'s 2003 metrics). 312, 384

Ratio of Steps and Activities

Is the metric `RSTPA` defined by García et al. [Gar+03], see Table B.5 (García et al.'s 2003 metrics). 312, 383

Ratio of Work Products and Activities

Is the metric `RWPA` defined by García et al. [Gar+03], see Table B.5 (García et al.'s 2003 metrics). 312, 313, 384

Relational Density

Is the metric `RD` defined by Kreimeyer [Kre10], see Table B.33 (Kreimeyer's 2010 metrics). 348, 384

Relative Control-Flow Complexity

Is the metric $\texttt{CFC}_{rel}$ defined by Cardoso [Car08], see Table B.20 (Cardoso's 2008 metrics). 333, 334, 384

Resource Size

Is the metric $\texttt{Size}_{R}$ defined by Antonini et al. [Ant+11], see Table B.36 (Antonini et al.'s 2011 metrics). 353, 384

Restrictiveness Estimator

Is the metric `RT` defined by Cardoso et al. [Car+06], see Table B.10 (Cardoso et al.'s 2006 metrics). This has the same meaning as `RT`, which was defined by Latva-Koivisto [Lat01], see Table B.4. 321

Restrictiveness Estimator

Is the metric `RT` defined by Latva-Koivisto [Lat01], see Table B.4 (Latva-Koivisto's 2001 metrics). 310, 384

Separability Ratio

Is the metric $\prod$ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). 329, 385

Sequence Control-Flow Complexity

Is the metric `C` defined by Fu et al. [Fu+10], see Table B.29 (Fu et al.'s 2010 metrics). 343, 385

Sequentiality Ratio

Is the metric Ξ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). 329, 385

Simplicity

Is the metric `Simplicity` defined by Tjaden [Tja99], see Table B.3 (Tjaden's 1999 metrics). 309, 385

Size of Control Flow Graph

Is the metric $\texttt{Size}_{CF}$ defined by Antonini et al. [Ant+11], see Table B.36 (Antonini et al.'s 2011 metrics). This has the same meaning as `NOAC`, which was defined by Cardoso et al. [Car+06], see Table B.10. 353

Size of Data Flow Graph

Is the metric $\texttt{Size}_{DF}$ defined by Antonini et al. [Ant+11], see Table B.36 (Antonini et al.'s 2011 metrics). 353, 385

Software Metric

This is "used to characterize the essential features of software quantitatively so that classification, comparison, and mathematical analysis can be applied" [Con+86] . 106, 112, 144, 151, 157, 315

Software Process Engineering Metamodel (SPEM)

Is a metamodel used to define process models, and was specifically designed to model the software development process [OMG02] . xix, 311, 313, 316

Split Complexity

Is the metric `SC` defined by Mendling et al. [Men+06], see Table B.12 (Mendling et al.'s 2006 metrics). This has the same meaning as `CFC`, which was defined by Cardoso [Car05b], see Table B.8. 324

State of the Art through Systematic Reviews (StArt)

Is a tool that supports the whole systematic review process and count with an online community of users [Fab+16]. It was created in 2010 by the Laboratory of Research on Software Engineering in the Computing Department of the Federal University of São Carlos in Brazil [Sof12] . xix, 104, 108, 110, 113, 114, 117, 118, 392

Structural Complexity

Is the metric `SCBP` defined by Muketha et al. [Muk+10b], see Table B.30 (Muketha et al.'s 2010 metrics). 345, 385

Structural Complexity

Is the metric $\texttt{H}_{SC}$ defined by Cheng [Che08], see Table B.22 (Cheng's 2008 metrics). 336, 385

Structured Activities

Is the metric `SA` defined by Held and Blochinger [HB09], see Table B.28 (Held and Blochinger's 2009 metrics). 343, 385

Structuredness

Is the metric `S` defined by La Rosa et al. [La +11b], see Table B.35 (La Rosa et al.'s 2011 metrics). 351, 386

Structuredness

Is the metric SM defined by Lassen and van der Aalst [Lv09], see Table B.25 (Lassen and van der Aalst's 2009 metrics). 338, 339, 386

Structuredness Ratio

Is the metric Φ defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). 329, 386

Subject-oriented Business Process Management (S-BPM)

Is a BPM approach that focuses on the subjects or actors in the process [Fle10] . xviii, 23

Systematic Literature Review (SLR)

Is a secondary study using a well define methodology and protocol to identify, analyze and interpret the available evidence relevant to a specific research question in an unbiased and to a certain degree repeatable manner [KC07] . xix, 4, 5, 24, 103–108, 112, 114, 119, 120, 122–124, 128–133, 135, 136, 157, 158, 160, 191–200, 305, 311, 316, 343, 347, 358, 391–394

Token Split

Is the metric TS defined by Mendling [Men07], see Table B.16 (Mendling's 2007 metrics). 330, 386

Total Basic Activity Count

Is the metric TBAC defined by Lübke [LÏ5], see Table B.40 (Lübke's 2015 metrics). This has the same meaning as NT, which was defined by Rolón et al. [Rol+06b], see Table B.9. 356

Total Number of Activities

Is the metric TNA defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). This has the same meaning as NA, which was defined by García et al. [Gar+03], see Table B.5. 317

Total Number of Collapsed Sub-Processes

Is the metric TNCS defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 319, 369

Total Number of Data Objects

Is the metric TNDO defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). This has the same meaning as NWP, which was defined by García et al. [Gar+03], see Table B.5. 320

Total Number of End Events

Is the metric `TNEE` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 320, 371

Total Number of Events

Is the metric `TNE` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 320, 372

Total Number of Gateways

Is the metric `TNG` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 320, 373

Total Number of Intermediate Events

Is the metric `NTIE` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 319, 374

Total Number of Start Events

Is the metric `NTSE` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). 319, 379

Total Number of Tasks

Is the metric `TNT` defined by Rolón et al. [Rol+06b], see Table B.9 (Rolón et al.'s 2006 metrics). This has the same meaning as `TNT`, which was defined by García et al. [Gar+03], see Table B.5. 320

Total Structured Activity Count

Is the metric `TSAC` defined by Lübke [LÏ5], see Table B.40 (Lübke's 2015 metrics). This has the same meaning as `SA`, which was defined by Held and Blochinger [HB09], see Table B.28. 356

Transfer Number Per Service

Is the metric `TNS` defined by Mao [Mao10a], see Table B.31 (Mao's 2010 metrics). 346, 386

Unified Modeling Language (UML)

Is a "visual language for specifying, constructing, and documenting the artifacts of systems. It is a general-purpose modeling language that can be used with all major object and component methods, and that can be applied to all application domains" [OMG09c] . xix, 22, 29, 65–68, 70, 71, 74, 75, 194, 195, 197, 199, 262, 265, 280, 288

Unified Modeling Language Activity Diagram (UML AD)

Is one of the many types of diagrams supported by UML. Activity diagrams describe the flow from one activity to another [OMG09c]. They can be used to model business processes . xix, 2, 4, 16, 21, 22, 26, 65–68, 70, 194, 195, 322, 334, 338, 351, 354

User Comprehension

Is an aspect of cognition that involves the mental processes that an individual uses to grasp the meaning of something [Swa07] . 160, 164, 193

Variety-Based Complexity

Is the metric `Complexity` defined by Chen and Prabhu [CP08], see Table B.23 (Chen and Prabhu's 2008 second set of metrics). 336, 386

Width of Hierarchy

Is the metric `WH` defined by Kreimeyer [Kre10], see Table B.33 (Kreimeyer's 2010 metrics). 348, 386

Wiener's Index

Is the metric `W` defined by Borgert et al. [Bor+09a], see Table B.27 (Borgert et al.'s 2009 metrics). 341, 386

Workflow

Is "the automation of a business process, in whole or in part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules" [Wes12] . 12, 13, 24, 27, 28, 32, 35, 67, 109–111, 114, 178, 311, 320, 326, 331, 336–338, 341

Workflow Management (WfM)

Is a model-driven approach that allows for the explicit representation of process models and for the controlled enactment of those models [Wes12] . xix, 12

Workflow Net (WF-Net)

Is a type of Petri Net introduced by van der Aalst [van95] to formalize workflow process models . xix, 21, 320, 322, 337–339

XML Process Definition Language (XPDL)

Is a process modeling language designed for process model interchange [WfM12]. XPDL does not provide a graphical notation . xix, 16, 21

Yet Another Workflow Language (YAWL)

Is a process modeling language inspired on Petri Nets [Ter+10; vT05] . xix, 16, 21, 322, 323, 325, 326, 334, 351, 354

Index

Appendix A

GSM to CMMN Syntax Directed Translation Grammar

This section described the syntax directed translation grammar [Aho+07] used to transform a Case type into an artifact type. The terminology used is described after the grammar.

A.1 Grammar

Production			$\Longrightarrow$ **Semantic rule**	
$\mathcal{T}$	$\rightarrow$	$\langle \mathcal{D} \;,\; \mathcal{B} \rangle \boxdot$	$\Longrightarrow$ $EType_{gen} \leftarrow EType_{gen} \cup standardEvents(\mathcal{T})$; $Emit \quad \Gamma$	
$\mathcal{D}$	$\rightarrow$	$\langle \mathcal{D}_{casefile} \;,\; \mathcal{D}_{discrete} \;,\; \mathcal{D}_{container} \rangle$		▷ Data
$\mathcal{D}_{casefile}$	$\rightarrow$	**cf**	$\Longrightarrow$ $\Gamma \leftarrow \Gamma \cup \Gamma^{data}_{pattern}(\mathbf{cf})$	
$\mathcal{D}_{discrete}$	$\rightarrow$	$\{ {\mathcal{D}_{discrete}}^{element} \}$		
${\mathcal{D}_{discrete}}^{element}$	$\rightarrow$	${\mathcal{D}_{discrete}}^{element}$ **, dd**	$\Longrightarrow$ apply $\downarrow$	
	$\mid$	**dd**	$\Longrightarrow$ $\Gamma \leftarrow \Gamma \cup \Gamma^{data}_{pattern}(\mathbf{dd})$	
$\mathcal{D}_{container}$	$\rightarrow$	$\{ {\mathcal{D}_{container}}^{element} \}$		
${\mathcal{D}_{container}}^{element}$	$\rightarrow$	${\mathcal{D}_{container}}^{element}$ **, dc**	$\Longrightarrow$ apply $\downarrow$	
	$\mid$	**dc**	$\Longrightarrow$ $\Gamma \leftarrow \Gamma \cup \Gamma^{data}_{pattern}(\mathbf{dc})$	
$\mathcal{B}$	$\rightarrow$	$\langle \mathscr{St} \;,\; \mathscr{Ta} \;,\; \mathscr{Mi} \;,\; \mathscr{Ev} \;,\; \mathscr{H} \;,\; \mathscr{R} \rangle$		▷ Behavior
$\mathscr{St}$	$\rightarrow$	$\langle \mathscr{St}_{case} \;,\; \mathscr{St}_{planned} \;,\; \mathscr{St}_{discretionary} \;,\; \mathscr{St}_{fragment} \rangle$		▷ Stages
$\mathscr{St}_{case}$	$\rightarrow$	$\{ \mathbf{c} \}$	$\Longrightarrow$ $\Gamma \leftarrow \Gamma \cup \Gamma^{case}_{pattern}(\mathbf{c})$	

$$\mathcal{St}_{planned} \rightarrow \{\mathcal{St}_{planned}{}^{element}\}$$

$$\mathcal{St}_{planned}{}^{element} \rightarrow \mathcal{St}_{planned}{}^{element}\ ,\ \mathbf{S} \Longmapsto \text{apply} \downarrow$$

$$\mid\ \mathbf{S} \Longmapsto \Gamma \leftarrow \Gamma \cup \Gamma^{stage}_{pattern}(\mathbf{S})$$

$$\mathcal{St}_{discretionary} \rightarrow \{\mathcal{St}_{discretionary}{}^{element}\}$$

$$\mathcal{St}_{discretionary}{}^{element} \rightarrow \mathcal{St}_{discretionary}{}^{element}\ ,\ \mathbf{dS} \Longmapsto \text{apply} \downarrow$$

$$\mid\ \mathbf{dS} \Longmapsto \Gamma \leftarrow \Gamma \cup \Gamma^{parallel}_{pattern}(\Gamma^{discretionary}_{pattern}(\Gamma^{stage}_{pattern}(\mathbf{dS})))$$

$$\mathcal{St}_{fragment} \rightarrow \{\mathcal{St}_{fragment}{}^{element}\}$$

$$\mathcal{St}_{fragment}{}^{element} \rightarrow \mathcal{St}_{fragment}{}^{element}\ ,\ \mathbf{f} \Longmapsto \text{apply} \downarrow$$

$$\mid\ \mathbf{f} \Longmapsto \Gamma \leftarrow \Gamma \cup \Gamma^{parallel}_{pattern}(\Gamma^{discretionary}_{pattern}(\mathbf{f}))$$

$$\mathcal{Ta} \rightarrow \langle \mathcal{Ta}_{planned}\ ,\ \mathcal{Ta}_{discretionary} \rangle$$ ▷ Tasks

$$\mathcal{Ta}_{planned} \rightarrow \{\mathcal{Ta}_{planned}{}^{element}\}$$

$$\mathcal{Ta}_{planned}{}^{element} \rightarrow \mathcal{Ta}_{planned}{}^{element}\ ,\ \mathbf{t} \Longmapsto \text{apply} \downarrow$$

$$\mid\ \mathbf{t} \Longmapsto \Gamma \leftarrow \Gamma \cup \Gamma^{task}_{pattern}(\mathbf{t})$$

$$\mathcal{Ta}_{discretionary} \rightarrow \{\mathcal{Ta}_{discretionary}{}^{element}\}$$

$$\mathcal{Ta}_{discretionary}{}^{element} \rightarrow \mathcal{Ta}_{discretionary}{}^{element}\ ,\ \mathbf{dt} \Longmapsto \text{apply} \downarrow$$

$$\mid\ \mathbf{dt} \Longmapsto \Gamma \leftarrow \Gamma \cup \Gamma^{parallel}_{pattern}(\Gamma^{discretionary}_{pattern}(\Gamma^{task}_{pattern}(\mathbf{dt})))$$

$$\mathcal{Mi} \rightarrow \{\mathcal{Mi}^{element}\}$$ ▷ Milestones

$$\mathcal{Mi}^{element} \rightarrow \mathcal{Mi}^{element}\ ,\ \mathbf{m} \Longmapsto \text{apply} \downarrow$$

$$\mid\ \mathbf{m} \Longmapsto \Gamma \leftarrow \Gamma \cup \Gamma^{milestone}_{pattern}(\mathbf{m})$$

$$\mathcal{Ev} \rightarrow \{\mathcal{Ev}^{element}\}$$ ▷ Event listeners

$$\mathcal{Ev}^{element} \rightarrow \mathcal{Ev}^{element}\ ,\ \mathbf{e} \Longmapsto \text{apply} \downarrow$$

$$\mid\ \mathbf{e} \Longmapsto \Gamma \leftarrow \Gamma \cup \Gamma^{eventlistener}_{pattern}(\mathbf{e})$$

$$\mathcal{H} \rightarrow \{\mathcal{H}^{element}\}$$ ▷ Hierarchy

$$\mathcal{H}^{element} \rightarrow \mathcal{H}^{element}\ ,\ \langle \mathbf{aS}\ ,\ \mathbf{v} \rangle \Longmapsto \text{apply} \downarrow$$

$$\mid\ \langle \mathbf{aS}\ ,\ \mathbf{v} \rangle \Longmapsto Substages^R \hookleftarrow Substages^R$$

$$\sqcup \{\langle \mathbf{aS}\ ,\ findPattern(\mathbf{v}) \rangle \mid \langle \mathbf{aS}\ ,\ \mathbf{v} \rangle \in \mathcal{H}\}$$

$$\mathcal{R} \rightarrow \langle \widehat{\varphi}\ ,\ \widehat{\mathscr{S}}\ ,\ \widehat{\mathscr{E}}\ ,\ \widehat{\mathscr{X}}\ ,\ \widehat{\mathscr{M}}\ ,\ \widehat{\mathscr{R}}\ ,\ \widehat{\mathscr{N}}\ ,\ \widehat{\mathscr{A}} \rangle$$ ▷ Rules

$$\widehat{\varphi} \rightarrow \{\widehat{\varphi}^{element}\}$$ ▷ Rule expressions

$$\widehat{\varphi}^{element} \rightarrow \widehat{\varphi}^{element}\ ,\ \mathbf{b} \Longmapsto \text{apply} \downarrow$$

$$\mid\ \mathbf{b} \Longmapsto Stry \leftarrow Stry \sqcup \{[\mathbf{if}\ exprConvert(\mathbf{b})]\}$$

$$\widehat{\mathscr{S}} \rightarrow \{\widehat{\mathscr{S}}^{element}\} \qquad \triangleright \text{ Sentries}$$

$$\widehat{\mathscr{S}}^{element} \rightarrow \widehat{\mathscr{S}}^{element}\ ,\ [\dot{\xi}\ ,\ \varphi] \quad \Longmapsto \text{ apply } \downarrow$$

$$\mid\ [\dot{\xi}\ ,\ \varphi] \quad \Longmapsto \quad \mathit{Stry} \leftarrow \mathit{Stry} \sqcup \{s \mid s = [\mathbf{on}\ \dot{\xi}\ \mathbf{if}\ \varphi] \ \vee\ [\mathbf{on}\ \dot{\xi}] \ \vee\ [\ \mathbf{if}\ \varphi]\}$$

$$\dot{\xi} \rightarrow \mathbf{l} \quad \Longmapsto \text{ apply } \downarrow \qquad \triangleright \text{ Event expression}$$

$$\mid\ \{\dot{\xi}^{element}\ ,\ \mathbf{l}\} \quad \Longmapsto \text{ apply } \downarrow$$

$$\dot{\xi}^{element} \rightarrow \dot{\xi}^{element}\ ,\ \mathbf{l} \quad \Longmapsto \text{ apply } \downarrow$$

$$\mid\ \mathbf{l} \quad \Longmapsto \quad \mathbf{l} \leftarrow \mathit{eventConvert}(\mathbf{l})$$

$$\varphi \rightarrow \mathbf{b} \quad \Longmapsto \quad \mathbf{b} \leftarrow \mathit{exprConvert}(\mathbf{b}) \qquad \triangleright \text{ Condition expression}$$

$$\mid\ \varnothing$$

$$\widehat{\mathscr{E}} \rightarrow \{\widehat{\mathscr{E}}^{element}\} \qquad \triangleright \text{ Entry criteria}$$

$$\widehat{\mathscr{E}}^{element} \rightarrow \widehat{\mathscr{E}}^{element}\ ,\ \langle \mathbf{s}\ ,\ \mathbf{w} \rangle \quad \Longmapsto \text{ apply } \downarrow$$

$$\mid\ \langle \mathbf{s}\ ,\ \mathbf{w} \rangle \quad \Longmapsto \quad \mathit{Guards}^{R} \hookleftarrow \mathit{Guards}^{R}$$

$$\sqcup \{\langle \Gamma^{splitEntry}_{pattern}(\mathbf{s}\ ,\ \mathit{findPattern}(\mathbf{w}))\rangle \mid \langle x\ ,\ w \rangle \in \widehat{\mathscr{E}}\}$$

$$\widehat{\mathscr{X}} \rightarrow \{\widehat{\mathscr{X}}^{element}\} \qquad \triangleright \text{ Exit criteria}$$

$$\widehat{\mathscr{X}}^{element} \rightarrow \widehat{\mathscr{X}}^{element}\ ,\ \langle \mathbf{s}\ ,\ \mathbf{x} \rangle \quad \Longmapsto \text{ apply } \downarrow$$

$$\mid\ \langle \mathbf{s}\ ,\ \mathbf{x} \rangle \quad \Longmapsto \quad \mathit{Terminators}^{R} \hookleftarrow \mathit{Terminators}^{R}$$

$$\sqcup \{\langle \Gamma^{splitExit}_{pattern}(\mathbf{s}\ ,\ \mathit{findPattern}(\mathbf{x}))\rangle \mid \langle s\ ,\ x \rangle \in \widehat{\mathscr{X}}\}$$

$$\widehat{\mathscr{M}} \rightarrow \{\widehat{\mathscr{M}}^{element}\} \qquad \triangleright \text{ Manual activation}$$

$$\widehat{\mathscr{M}}^{element} \rightarrow \widehat{\mathscr{M}}^{element}\ ,\ \langle \mathbf{b}\ ,\ \mathbf{y} \rangle \quad \Longmapsto \text{ apply } \downarrow$$

$$\mid\ \langle \mathbf{b}\ ,\ \mathbf{y} \rangle \quad \Longmapsto \quad \Gamma \leftarrow \Gamma\ \sqcup\ \Gamma^{manual}_{pattern}(b, \mathit{findPattern}(\mathbf{y}))$$

$$\widehat{\mathscr{R}} \rightarrow \{\widehat{\mathscr{R}}^{element}\} \qquad \triangleright \text{ Required}$$

$$\widehat{\mathscr{R}}^{element} \rightarrow \widehat{\mathscr{R}}^{element}\ ,\ \langle \mathbf{b}\ ,\ \mathbf{z} \rangle \quad \Longmapsto \text{ apply } \downarrow$$

$$\mid\ \langle \mathbf{b}\ ,\ \mathbf{z} \rangle \quad \Longmapsto \quad \Gamma \leftarrow \Gamma\ \sqcup\ \Gamma^{required}_{pattern}(b, \mathit{findPattern}(\mathbf{z}))$$

$$\widehat{\mathscr{N}} \rightarrow \{\widehat{\mathscr{N}}^{element}\} \qquad \triangleright \text{ Repetition}$$

$$\widehat{\mathscr{N}}^{element} \rightarrow \widehat{\mathscr{N}}^{element}\ ,\ \langle \mathbf{b}\ ,\ \mathbf{w} \rangle \quad \Longmapsto \text{ apply } \downarrow$$

$$\mid\ \langle \mathbf{b}\ ,\ \mathbf{w} \rangle \quad \Longmapsto \quad \Gamma \leftarrow \Gamma\ \sqcup\ \Gamma^{parallel}_{pattern}(\Gamma^{repetition}_{pattern}(b, \mathit{findPattern}(w)))$$

$$\widehat{\mathscr{A}} \rightarrow \{\widehat{\mathscr{A}}^{element}\} \qquad \triangleright \text{ Auto-complete}$$

$$\widehat{\mathscr{A}}^{element} \rightarrow \widehat{\mathscr{A}}^{element}\ ,\ \mathbf{eS} \quad \Longmapsto \text{ apply } \downarrow$$

$$\mid\ \mathbf{eS} \quad \Longmapsto \quad \Gamma \leftarrow \Gamma\ \sqcup\ \Gamma^{auto}_{pattern}(\mathit{findPattern}(\mathbf{eS}))$$

A.2 Terminology

Non-terminal symbols include the symbols defined in Definition 3.4, Definition 3.5, Definition 3.6, Definition 3.7, and Definition 3.8, plus the expansion of their elements.

Constant terminal symbols includes

- ⊡ indicates end of input
- ⟨, ⟩ for ordered tuples
- {, } for sets
- [,] for sentries
- comma (,) as separator

Non-Constant terminal symbols are

- **aS** is a stage (case, planned, discretionary, or fragment)
- **b** is a Boolean expression. It could be a rule expression or a sentry condition expression
- **cf** is a case file container
- **c** is a case stage
- **dS** is a discretionary stage
- **dc** is a data container
- **dd** is a discrete data element
- **dt** is a discretionary task
- **eS** is an executable stage (case, planned, discretionary)
- **e** is an event listener
- **f** is a plan fragment
- **l** is a sentry event expression
- **m** is a milestone
- **s** is a sentry
- **S** is a planned stage
- **t** is a planned task
- **v** could be a stage (case, planned, discretionary, or fragment), task (planned or discretionary), milestone, or event listener
- **w** could be a stage (planned or discretionary), task (planned or discretionary), or milestone
- **x** could be a stage (case, planned or discretionary), or task (planned or discretionary)
- **y** could be a stage (planned or discretionary), or task (planned or discretionary)
- **z** could be a planned stage, or planned task

Others include

- $\Longmapsto$ indicates the start of a semantic rule.
- apply $\downarrow$ indicates the semantic rule is identical as the rule in the next line
- $\triangleright$ indicates the rest of the line is a comment
- $\sqcup$ indicates set union without duplicating elements.

Appendix B

Process Modeling Complexity Metrics

This appendix describes the raw data that was extracted during the Systematic Literature Review (SLR). Additional information about the SLR can be found in the supplementary documents provided in Appendix D.3. Additionally, the two spreadsheets contained in files 32 (Analysis.xlsx) and 31 (report(full).xls) can also be found in Appendix D.

B.1 Identified Papers

This section describes all of the primary and secondary papers that were identified during the SLR. In some cases, duplicate and other papers are also included. Primary papers are those that define new metrics, and secondary papers are usually those that validate the metrics. For all of the primary papers, a table describing the proposed metrics has been included. The section presents the primary papers in chronological order, with the secondary and duplicate papers being presented with their corresponding primary papers.

Daneva et al. [Dan+96] (1996) described three cohesion metrics that are to be used as complexity metrics for Event-driven Process Chain (EPC) [van99] models. They refined the metrics based on the expectations of expert users. Their work was done in the context of Business Process Re-engineering (BPR). They presented three cohesion metrics that they considered to be useful for evaluating complexity. Although, these three metrics were not defined at the process level, they also presented the average of these metrics at the process level. In this review the average of these metrics was considered as they are at the process level. The paper does not describe the protocol used or the number of experts consulted. The empirical validation followed Fenton's [Fen90] guidelines.

Table B.1: Daneva et al.'s 1996 metrics

<table>
<tr><th>Metric</th><th>Description</th></tr>
<tr><td>AFC</td><td>Average function cohesion.

$$AFC = \frac{\sum_{i \in F} FC_i}{|F|} \times 100$$

Where,
F is the set of functions in the process.
FC is the structural complexity of the control-flow calculated as

$$FC = \frac{FC_{inp} + FC_{out}}{2}$$

FC_{inp} is the input function cohesion defined as

$$FC_{inp}(n) = K_n \times N_n$$
$$FC_{inp}(i) = K_i \times (FC_{inp}(i+1) + N_i) \qquad i = 1, \ldots, n-1$$

FC_{out} is the output function cohesion defined as

$$FC_{out}(l) = K_l \times M_l$$
$$FC_{out}(j) = K_j \times (FC_{out}(j+1) + M_j) \qquad j = 1, \ldots, l-1$$

i and j are nesting levels
N_i is the number of input events at nesting level i
M_j is the number of output events at nesting level l
k_i and k_j are weights given by the table:

<table>
<tr><th>Logical connector</th><th>Weight</th></tr>
<tr><td>AND</td><td>3</td></tr>
<tr><td>XOR</td><td>3</td></tr>
<tr><td>OR</td><td>2</td></tr>
</table>
</td></tr>
<tr><td>AEC</td><td>Average event cohesion.

$$AEC = \frac{\sum_{i \in E} EC_i}{|E|} \times 100$$

Where,
E is the set of events in the process.
EC is the event cohesion defined as

$$EC = EC_{inp} + EC_{out} + c \times NF$$

EC_{inp} is the input event cohesion defined as

$$EC_{inp}(n) = K_n \times NI_n$$
$$EC_{inp}(i) = K_i \times (FC_{inp}(i+1) + NI_i) \qquad i = 1, \ldots, n-1$$
</td></tr>
</table>

Continued on next page

Table B.1 – *Continued from previous page*

Metric	Description
	EC_{out} is the output event cohesion defined as $EC_{out}(l) = K_l \times NO_l$ $EC_{out}(j) = K_j \times (EC_{out}(j+1) + NO_j) \qquad j = 1, \ldots, l-1$ c is an event cohesion factor that could be any of the following values: 1. The event is produced by one function only. 2. The event is produced by more than one function. 3. The event is produced by only one function and it is an input to more than one function. 4. The event is produced by more than one function, and it is input to more than one function. NF is the number of functions affected by the event. i and j are nesting levels. NI_i is the number of co-events at nesting level i. NO_j is the number of co-events at nesting level l.
`ACC`	Average connector cohesion. $ACC = \frac{\sum_{i \in C} CLC_i}{\lvert C \rvert} \times 100$ Where, C is the set of connectors in the process. CLC is the Cohesion of a Logical Connector defined as $CLC = \sqrt{NO^2 + NA^2}$ NO is the weighted number of objects defined as $NO = NF + NE + 2 \times NC$ NA is the number of arrows between objects. NF is the number of functions. NE is the number of events. NC number of connectors.

Two evaluations were conducted by Daneva et al. [Dan+96]: one consisted of calculating the metrics of 11 EPC models and asking an undisclosed number of experts to provide their opinions of the same models. The second, consisted of selecting a few EPCprocesses as best of class (best practices) and comparing two EPCprocesses against them using the metrics. The paper concluded that the metrics could be used to detect errors and to evaluate the progress of corrective actions. The three average metrics are described in Table B.1.

Sobrinho [Sob99] (1999) was concerned with process evolution, and presented three structural complexity metrics that deal with process evolution. This work was done in the context of BPR. His work tries to improve the maintainability of processes over time by devising a systematic process re-modularization scheme. Table B.2 describes the proposed metrics

Table B.2: Sobrinho's 1999 metrics

Metric	Description
`GREM`	Global ripple effect measure. $$\texttt{GREM} = \frac{\sum_{i=1} N\texttt{LINF}(i)}{\sum_{i=1} N\texttt{GINF}(i)}$$ Where, N is the number of activities in the process. `LINF` is the local influence of a component defined as $$\texttt{LINF}(i) = \sum_{(i,j)\in D} I(i,j)n_{ij}$$ D is the set of directed neighbors of component i. $I(i,j)$ is the incidence matrix, with $I(i,j) = 1$ if component i and j are directly connected, $I(i,j) = 0$ otherwise. n_{ij} is the number of activities affected in component j because a change in component i. `GINF` is the global influence of a component defined as $$\texttt{GINF}(i) = \sum_{j\neq i, j=1} Nn_{ij}$$
`EM`	Extended measure. $$EM = \frac{\texttt{POD}}{N}$$ Where, `POD` is the stability of the overall process design defined as $$\texttt{POD} = \frac{\sum_{i=1}^{N} GINF(i)}{N}$$
`MM`	Modularization measure. $$MM = \frac{\sum_{i=1}^{N}\sum_{j=1} NP_c(i,j)}{N}$$ Where, $P_c(i,j)$ is a cell in the probability dependency matrix, where $P_c(i,j)$ is the probability of changing component (subprocess) i because a change in component (subprocess) j.

Tjaden [Tja99] (1999) described process metrics for process models in the context of BPR. He assumed that the number of people involved in the execution of the process, the number of material flows, and the number of operations executed per activity was known to the modeler. He used a simple graph notation, where the boxes represent activities and the arrows represent the transfer of material or information. Table B.3 describes the proposed metrics.

Table B.3: Tjaden's 1999 metrics

Metric	**Description**
$\bar{C}_A$	Average activity complexity. $$\bar{C}_A = \frac{\sum_{i=1}^{N} C_B}{N}$$ N number of activities in the process. C_B is the basic activity complexity $C_B = a + m + p$ a is the number of sub-activities. m number of material flow. p is number of people.
`Simplicity`	Simplicity. $$\texttt{Simplicity} = \frac{\bar{C}_A - \min \bar{C}_A}{\max \bar{C}_A - \min \bar{C}_A} \times 100$$
`Flexibility`	Flexibility. $$\texttt{Flexibility} = \frac{\sum_{i=1}^{m_{op}} Fm_i}{60} \times 100$$ Where, m_{op} is the number of operational flows. Fm_i is the material flow for the ith operational flow. This is calculated based on a table provided by the author.
`Integration`	Integration. $$\texttt{Integration} = \frac{\sum_{i=1}^{m} I_i}{10} \times 100$$ Where, m is the number of flows. I_i is the integration value for flow i, calculated based on a table provided by the author.

Latva-Koivisto [Lat01] (2001) recognized the importance of complexity metrics in managing business processes. In particular, he associated higher complexity with process failure, management difficulty, and cost. In addition, he proposed six criteria that are to be used to ensure a good complexity measure. He recognized that business processes are normally modeled by process maps or process charts, which are similar to flowcharts in that they are composed of activities and the dependencies between activities. Therefore, business processes can be easily mapped into directed cyclic graphs, which he attempts to do in order to derive complexity metrics from graph theory. He applied six graph complexity metrics to business processes, by mapping the process map or chart into a directed cyclic graph. The six metrics are described in Table B.4.

Table B.4: Latva-Koivisto's 2001 metrics

Metric	**Description**
$\mathtt{CNC}_K$	Coefficient of network complexity (kaimann). [Kai74], $$\mathtt{CNC}_K = \frac{\mathtt{A}^2}{\mathtt{N}}$$ Where, $\mathtt{N}$ is the number of nodes in the graph. $\mathtt{A}$ is the number of arcs in the graph.
$\mathtt{CNC}_P$	Coefficient of network complexity (pascoe). [Pas66], $$\mathtt{CNC}_P = \frac{\mathtt{A}}{\mathtt{N}}$$
S	Cyclomatic number. [Tem81], $$\mathtt{S} = \mathtt{A} - \mathtt{N} + 1$$
CI	Complexity index. [Bei+92]. Defined in terms of node reduction of a two-terminal directed acyclic graph. The minimum number of node reductions sufficient to reduce the graph to a single edge is called the complexity index. $$\mathtt{CI} = \{c \text{ such that } [[[[G]_r v_1]_r v_2] \ldots_r v_c] \text{ is a single edge}\}$$ where, G is a two terminal directed acyclic graph (st-dag). $G_r v$ denote the result of node reduction to graph G. $[G]$ denotes the graph that results when all possible series and parallel reductions have been applied to G.
RT	Restrictiveness estimator. [Sch95], $$\mathtt{RT} = \frac{2\sum r_{ij} - 6(\mathtt{N}-1)}{(\mathtt{N}-2)(\mathtt{N}-3)}$$ Where, R_{ij} is n element of the reachability matrix. The reachability matrix $R = [r_{ij}]$ of a directed graph is a NxN matrix, where $r_{ij} = 1$ if there is path from node n_i to node n_j, otherwise $r_{ij} = 0$

Continued on next page

Table B.4 – *Continued from previous page*

Metric	Description
T	Number of trees in a graph. [Tem81], $$\mathtt{T} = \sum_{i \in \text{ Sink nodes}} \mathtt{D}_{ij}$$ Where, $\mathtt{D}_{ij}$ is a tree-generating determinant defined using the reachability matrix, as $$\mathtt{D} = \begin{vmatrix} \sum_{j\neq 1} r_{1j} & -r_{12} & -r_{13} & \cdots \\ -r21 & \sum_{j\neq 2} r_{2j} & -r_{23} & \cdots \\ -r31 & -r32 & \sum_{j\neq 3} r_{3j} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{vmatrix}$$ Sink nodes are those with no outgoing arcs.

Reijers [Rei03] (2003) introduced a workflow cohesion metric based on the software engineering cohesion metrics defined by Stevens et al. [Ste+74], and provided some heuristics based on the work done by Selby and Basili [SB91] to make decisions about various workflow design alternatives. Theoretical validation was conducted using Chidamber and Kemerer's [CK94] principles based on the work done by Weyuker [Wey88]. Although Reijers [Rei03] does not claim that the metrics are complexity metrics, they were validated against Chidamber and Kemerer [CK94] which includes a principle stating that *interaction increases complexity*. The heuristics were empirically validated by conducting a web survey that was answered by 15 participants with workflow design experience. However, his work only considered metrics at the activity level, which are not relevant to this SLR.

García et al. [Gar+03] (2003) presented a framework for process models and metrics, and implemented a tool for defining and calculating process model metrics. They proposed a set of metrics, nine of which were at the process level. The defined metrics are provided in Table B.5.

García et al. [Gar+04a] (2004) recognized that metrics focus on the project and products, but not on the models. Therefore, they proposed five new metrics to the set proposed by García et al. [Gar+03], which evaluate the influence of model complexity on process maintenance. They specifically focused on the influence of the metrics on the understandability and modifiability of the process models. In order to validate the metrics they used the Software Process Engineering Metamodel (SPEM) [OMG02] notation, and conducted a series of four empirical experiments with human subjects. They claimed that their work using SPEM could be applied to other process modeling notations. The metrics are described in Table B.6.

Table B.5: García et al.'s 2003 metrics

Metric	Description
NA	Number of activities. This is called "A" by Latva-Koivisto [Lat01], but it was not considered a metric.
NWP	Number of work products.
NPR	Number of roles.
NDRA	Number of precedence dependences between activities. It is calculated by counting the number of arcs connecting two activities. Using [Lat01] terminology $$\mathtt{NDRA} = \sum_{\substack{j=0\\i=0}}^{N} r_{ij}$$ which corresponds to counting all of the r_{ij} in the reachability matrix, where the nodes are the activities of the process.
CA	Activity coupling. $$\mathtt{CA} = \frac{\mathtt{NA}}{\mathtt{NDRA}}$$
RPRA	Ratio of process roles and activities. $$\mathtt{RPRA} = \frac{\mathtt{NWP}}{\mathtt{NA}}$$
RWPA	Ratio of work products and activities. $$\mathtt{RWPA} = \frac{\mathtt{NPR}}{\mathtt{NA}}$$
NSTP	Number of steps (tasks). in an activity.
RSTPA	Ratio of steps and activities. $$\mathtt{RSTPA} = \frac{NSTP}{NA}$$

From 2004 to 2006, García et al. presented a series of papers describing a group of five experiments they conducted during that time. García et al. [Gar+04d] (2004) conducted an initial validation of their metrics. Their first and second controlled experiments were based on the subjective ranking of Analysability, Understandability, Modifiability. García et al. [Gar+04b] conducted further validations of their metrics. Their third controlled experiment validated a few metrics that correlated with Understandability. García et al. [Gar+04a] (2004) described their fourth experiment, and García et al. [Gar+04c; Gar+06] described their fifth experiment. They described a framework that manages the modeling and measurements of business processes. The framework used the SPEM notation and the metrics described in García et al. [Gar+04a], and was based on a set of ontologies and meta-models that represent SPEM and the different measurement models. In addition, their papers provide detailed descriptions of the group of five experiments [Can+05].

Table B.6: García et al.'s 2004 metrics

Metric	**Description**
NA	Number of activities. [Gar+03]
NWP	Number of work products. [Gar+03]
NPR	Number of roles. Roles participating in the process [Gar+03]
NDWPIn	Number of input dependences. It is calculated by counting the number of arrows landing in activities coming from work products.
NDWPOut	Number of output dependences. It is calculated by counting the number of arrows landing in work products coming from activities.
NDWP	Number of dependences. $$\mathtt{NDWP} = \mathtt{NDWPIn} + \mathtt{NDWPOut}$$
NDRA	The number of precedence dependencies between activities [Gar+03]
CA	Activity coupling. [Gar+03]
RDWPIn	Ratio between input dependencies and number of dependencies. Calculated as $$\mathtt{RDWPIn} = \frac{\mathtt{NDWPIn}}{\mathtt{NDWP}}$$
RDWPOut	Ratio between output dependencies and the number of dependencies. Calculated as $$\mathtt{RDWPIn} = \frac{\mathtt{NDWPOut}}{\mathtt{NDWP}}$$
RPRA	Ratio of work products and activities. [Gar+03]
RWPA	The ratio of process roles and activities [Gar+03]

Reijers and Vanderfeesten [RV04] (2004) extended Reijers's [Rei03] work which proposed set of process level activity cohesion and coupling metrics. They proposed the following heuristic: "design with the minimum process coupling/cohesion ratio, is the best design" [RV04] and explained the rationality for it. The metrics that they introduced are described in Table B.7. Although not explicitly stated, they assumed that their metric had some relation with complexity, as indicated by their statement that "it appeals to our intuition that very large activity AE is not very attractive, because of its relatively high complexity and in-cohesive structure" [RV04].

Table B.7: Reijers and Vanderfeesten's 2004 metrics

Metric	Description
$\mathtt{c}$	Process cohesion. Defined in an operation structure $(\mathtt{D}, O)$, which is based on Activity cohesion and is defined as $$\mathtt{c} = \frac{\sum_{t\in\mathtt{T}} \mathtt{c}_a(t)}{\lvert\mathtt{T}\rvert}$$ Where, $\mathtt{T}$ is a set of activities, such that $\forall o{\in}O : (\exists t{\in}\mathtt{T} : o{\in}t)$ $\mathtt{c}_a(t)$ is the Activity cohesion for an activity t on an operation structure $(\mathtt{D}, O)$ and it is defined as $$c\,(t) = \lambda(t){\cdot}\mu(t)$$ $\lambda(t)$ is the Activity relation cohesion for an activity t on an operation structure $(\mathtt{D}, O)$ is $$\lambda\,(t) = \begin{cases} \frac{\sum_{(p,cs)\in t} \lvert\{(q,ds)\in t\backslash\{(p,cs)\}\mid(\{p\}\cup cs)\cap(\{q\}\cup ds)\neq\varnothing\}\rvert}{\lvert t\rvert\cdot\lvert t-1\rvert}, & \text{for } \lvert t\rvert > 1 \\ 0, & \text{for } \lvert t\rvert \leq 1 \end{cases}$$ Where operation structure is a tuple $(\mathtt{D}, O)$ such that $\mathtt{D}$: is the set of elements being processed, where each element corresponds to a node in a graph. O: is the set of operations corresponding to arcs in the graph, which is expected to be connected and acyclic $$O = \left\{(p, cs) \in\mathtt{D} \times \prod(\mathtt{D})\right\}$$ Information elements must not be dangling, and not depend on itself $$R = \{(p, c) \in\mathtt{D} \times \mathtt{D} \mid \exists\,(p, cs) \in O : c{\in}cs\}$$ p denotes the output information element. cs represents the input element of an operation. $\mu(t)$ is the Activity information cohesion for an activity t on an operation structure $(\mathtt{D}, O)$ is $$\mu\,(t) = \begin{cases} \frac{\lvert\{d\in\mathtt{D}\mid\exists(p,cs),(q,ds)\in t:d\in(\{p\}\cup cs)\cap(\{q\}\cup ds)\wedge(p,cs)\neq(q,ds)\}\rvert}{\lvert\{d\in\mathtt{D}\mid\exists(p,cs)\in t:d\in(\{p\}\cup cs)\}\rvert}, & \text{for } \lvert t\rvert > 0 \\ 0, & \text{for } \lvert t\rvert = 0 \end{cases}$$

Continued on next page

Table B.7 – *Continued from previous page*

Metric	Description
$\mathtt{k}$	Process coupling. An operation structure $(\mathtt{D}, O)$, is defined as $$\mathtt{k} = \begin{cases} \frac{\sum_{s,t \in \mathtt{T}} connected(s,t)}{\lvert\mathtt{T}\rvert \cdot (\lvert\mathtt{T}\rvert - 1)}, & \text{for } \lvert\mathtt{T}\rvert > 1 \\ 0, & \text{for } \lvert\mathtt{T}\rvert \leq 1 \end{cases}$$ Where, $$connected\,(s,t) = \begin{cases} 1, & \text{if}\,(s \neq t) \wedge (\exists\,(p,cs) \in s \wedge (q,ds) \in t : (\{p\} \cup cs) \cap (\{q\} \cup ds) \neq \varnothing) \\ 0, & \text{otherwise} \end{cases}$$
ρ	Process coupling/cohesion ratio. $$\rho = \frac{\mathtt{k}}{c}$$

Canfora et al. [Can+05] (2005) used the same metrics that were defined by García et al. [Gar+04a] and added an additional empirical experiment that validated the results obtained by García et al. In total, they presented a group of five experiments. They used 224 subjects in total, and each experiment used between 10 and 18 process models, and focused on understandability and modifiability. They used a within-subject experiment design, where all of the subjects were exposed to all of the process models. This was the same procedure used by García et al. [Gar+04a].

Cardoso [Car05b] (2005) introduced a control-flow complexity described in Table B.8. He validated the `CFC` metric using Weyuker's [Wey88] properties for software metrics. Two other papers[Car05c; Car05d] described the same `CFC` metric, with no validation.

Cardoso [Car06b] (2006) and Cardoso [Car07b] defined "process complexity as the degree to which a business process is difficult to analyze, understand or explain". Cardoso classified process complexity into four types: activity complexity, control-flow complexity, data-flow complexity, and resource complexity.

Cardoso [Car05a] (2005) introduced data-flow complexity metrics used for web processes described in Business Process Execution Language (BPEL) [OAS07]. He discussed three metrics, data complexity, interface complexity, and interface integration complexity. His data-flow complexity metrics were not relevant to process models because the data and interface definitions are not modeled in most process modeling notations, or when process models are represented by graphs.

Table B.8: Cardoso's 2005 metrics

Metric	Description
`AC`	Activity complexity. This has the same meaning as `NA`, which was defined by García et al. [Gar+03], see Table B.5.
`CFC`	Control-flow complexity. Calculated as $$\texttt{CFC}(p) = \sum_{i \in \{\text{XOR-splits of p}\}} \texttt{CFC}_{XOR-split}(i) + \sum_{j \in \{\text{OR-splits of p}\}} \texttt{CFC}_{OR-split}(j) + \sum_{k \in \{\text{AND-splits of p}\}} \texttt{CFC}_{AND-split}(k)$$ Where, $\texttt{CFC}_{XOR-split}(XOR\ \ activity) = n$ $\texttt{CFC}_{OR-split}(OR\ \ activity) = 2^n - 1$ $\texttt{CFC}_{AND-split}(AND\ \ activity) = 1$ n is the fan-out of the split, which corresponds to the number of outgoing arcs from the split node.

Rolón et al. [Rol+06b] (2006) based their work on the metrics defined for SPEM by García et al. [Gar+04a] and applied these metrics to Business Process Model and Notation (BPMN) [OMG13]. In addition, to re-base the metrics from SPEM to BPMN, they increased the number of metrics from 12 to 57 by having metrics corresponding to the count of every visual modeling element in BPMN. No validation of the metrics was presented in this paper. Rolón et al. [Rol+06b; Rol+06d] (2006) are shorter versions of Rolón et al. [Rol+05], but published in English. This review used [Rol+06b] which is the same document as [Rol+05], but [Rol+05] was published in Spanish and so cannot form part of the SLR. Rolón et al. [Rol+06a; Rol+06c] (2006) reiterate the metrics presented in Rolón et al. [Rol+05] to evaluate the structural complexity of business process models and propose an experimental plan consisting of a group of experiments that empirically validate the metrics. The independent variable was the described metrics, and the dependent variables corresponded to understandability and modifiability. The notation used was BPMN. Table B.9 presents the metrics.

Rolón et al. [Rol+07a] (2007) described an initial experiment that was designed to test the metrics described in Rolón et al. [Rol+05] (2005). The work produced by Rolón et al. [Rol+07b] (2007), which was published in Spanish (not included in the review) described a set of five experiments used to evaluate the structural complexity of business process models. They used ten models and during the course of the five experiments they tested 109 subjects. The experiments were described by Rolón et al. [Rol+06b; Rol+06c] (2006), and the metrics were described in Rolón et al. [Rol+05] (2005). They used the complexity metrics as the independent variable. The dependent variables were process understandability and process modifiability. The dependent variables were measured by using the time that it took the subjects to complete a set of tasks. For understandability the accuracy of the subjects' answers was used, and for modifiability the accuracy of the model modifications was used. In addition, they received a subjective rating of the ten processes from the subjects. These results were presented again in Rolón et al. [Rol+08] (2008).

Rolón et al. [Rol+08] (2008) described a set of metrics that have been theoretically validated using the Briand et al.'s [Bri+96] framework, but no further information or reference, about the theoretical validation, is given. However, Rolón [Rol09] describes the theoretical validation in Spanish.

Table B.9: Rolón et al.'s 2006 metrics

Type	**Metric**	**Description**
Task counters	NT	Number of simple tasks.
	TNA	Total number of activities. This has the same meaning as NA, which was defined by García et al. [Gar+03], see Table B.5.
	NTC	Number of compensation tasks.
	NTL	Number of looping tasks.
	NTMI	Number of multiple instances tasks.
Sub-process counters	NCS	Number of collapsed sub-process.
	NCSA	Number of collapsed ad-hoc sub-process.
	NCSC	Number of collapsed compensation sub-process.
	NCSL	Number of collapsed looping sub-process.
	NCSMI	Number of collapsed multiple instance sub-process.
Start event counters	NSLE	Number of start link event.
	NSMsE	Number of start message event.
	NSMuE	Number of start multiple event.
	NSNE	Number of start none event.
	NSRE	Number of start rule event.
	NSTE	Number of start timer event.
Intermediate event counters	NICaE	Number of intermediate cancel event.
	NICoE	Number of intermediate compensation event.
	NIEE	Number of intermediate error event.
	NILE	Number of intermediate link event.
	NIMsE	Number of intermediate message event.
	NIMuE	Number of intermediate multiple event.
	NINE	Number of intermediate none event.

Continued on next page

Table B.9 – *Continued from previous page*

Type	Metric	Description
	NIRE	Number of intermediate rule event.
	NITE	Number of intermediate timer event.
End event counters	NECaE	Number of end cancel event.
	NECoE	Number of end compensation event.
	NEEE	Number of end error event.
	NELE	Number of end link event.
	NEMsE	Number of end message event.
	NEMuE	Number of end multiple event.
	NENE	Number of end none event.
	NETE	Number of end terminate event.
Gateway counters	NCD	Number of complex decision/merge.
	NEDDB	Number of exclusive decision/merge data-based.
	NEDEB	Number of exclusive decision/merge event-based.
	NID	Number of incusive decision/merge.
	NPF	Number of parallel fork/join.
Other notation element counters	NDOIn	Number of input data objects.
	NDOOut	Number of output data objects.
	NL	Number of lanes. This has the same meaning as NPR, which was defined by García et al. [Gar+03], see Table B.5.
	NMF	Number of message flows.
	NP	Number of pools.
	NSF	Number of sequence flows. This has the same meaning as NDWP, which was defined by García et al. [Gar+04a], see Table B.6.
	NSFA	Number of sequence flows between activities.
	NSFE	Number of sequence flows from events.
	NSFG	Number of sequence flows from gateways.
Ratios	CLA	Connectivity level between activities. $$\texttt{CLA} = \frac{\texttt{TNT}}{\texttt{NSF}}$$
	CLP	Connectivity level between pools. $$\texttt{CLP} = \frac{\texttt{NMF}}{\texttt{NP}}$$

Continued on next page

Table B.9 – *Continued from previous page*

Type	Metric	Description
	PDOPIn	Rate of input data object over the total of data objects. This has the same meaning as RDWPIn, which was defined by García et al. [Gar+04a], see Table B.6. $$\texttt{PDOPIn} = \frac{\texttt{NDOIn}}{\texttt{TNDO}}$$
	PDOPOut	Rate of output data object over the total of data objects. This has the same meaning as RDWPOut, which was defined by García et al. [Gar+04a], see Table B.6. $$\texttt{PDOPOut} = \frac{\texttt{NDOOut}}{\texttt{TNDO}}$$
	PDOTOut	Rate of output data object over the total of tasks. This has the same meaning as RWPA, which was defined by García et al. [Gar+03], see Table B.5. $$\texttt{PDOTOut} = \frac{\texttt{NDOOut}}{\texttt{TNT}}$$
	PLT	Rate of pools and lanes over the total of tasks. This has the same meaning as RPRA, which was defined by García et al. [Gar+03], see Table B.5. $$\texttt{PLT} = \frac{\texttt{NL}}{\texttt{TNT}}$$
Calculated	NTIE	Total number of intermediate events. $$\texttt{NTIE} = \texttt{NINE} + \texttt{NITE} + \texttt{NIMsE} + \texttt{NIEE} + \texttt{NICaE} + \texttt{NICoE} + \texttt{NIRE} + \texttt{NILE} + \texttt{NIMuE}$$
	NTSE	Total number of start events. $$\texttt{NTSE} = \texttt{NSNE} + \texttt{NSTE} + \texttt{NSMsE} + \texttt{NSRE} + \texttt{NSLE} + \texttt{NSMuE}$$
	TNCS	Total number of collapsed sub-processes. $$\texttt{TNCS} = \texttt{NCS} + \texttt{NCSL} + \texttt{NCSMI} + \texttt{NCSC} + \texttt{NCSA}$$

Continued on next page

Table B.9 – *Continued from previous page*

Type	Metric	Description
	TNDO	Total number of data objects. This has the same meaning as NWP, which was defined by García et al. [Gar+03], see Table B.5. $$\texttt{TNDO} = \texttt{NDOIn} + \texttt{NDOOut}$$
	TNE	Total number of events. $$\texttt{TNE} = \texttt{NTSE} + \texttt{NTIE} + \texttt{TNEE}$$
	TNEE	Total number of end events. $$\texttt{TNEE} = \texttt{NENE} + \texttt{NEMsE} + \texttt{NEEE} + \texttt{NECaE} + \texttt{NECoE} + \texttt{NELE} + \texttt{NEMuE} + \texttt{NETE}$$
	TNG	Total number of gateways. $$\texttt{TNG} = \texttt{NEDDB} + \texttt{NEDEB} + \texttt{NID} + \texttt{NCD} + \texttt{NPF}$$
	TNT	Total number of tasks. This has the same meaning as TNT, which was defined by García et al. [Gar+03], see Table B.5. $$\texttt{TNT} = \texttt{NT} + \texttt{NTL} + \texttt{NTMI} + \texttt{NTC}$$

Cardoso et al. [Car+06] (2006) surveyed complexity metrics in software engineering, cognitive science, and graph theory to derive complexity metrics that were applicable to process models. Although they mentioned EPC and WF-Nets, they did not specify a notation as they wanted to create generic process metrics. They did not provide validation for the proposed metrics. The metrics are described in Table B.10.

Cardoso [Car06a] (2006) evaluated complexity from a few perspectives including information theory, Kolmogorov complexity, cyclomatic complexity, cognitive complexity, and computational complexity for their applicability to process complexity. He used a generic control-flow graph as the modeling notation. He also used the definition of complexity provided in IEEE [IEE90], and defined workflow complexity as the "the degree to which a process is difficult to analyze, understand or explain" [Car06a]. He equated workflows to complex systems. The article describes exploratory research and does not validate the proposed metrics. He proposed and formally described few metrics that did not meet the criteria for this review including:

H which corresponds to the Shannon's entropy of a system [Sha48], but unfortunately is a runtime metric and not applicable to the modeling phase; and K which corresponds to the Kolmogorov complexity, but it requires a repository of processes in order to be calculated.

Table B.10: Cardoso et al.'s 2006 metrics

<table>
<tr><th>Metric</th><th colspan="2">Description</th></tr>
<tr><td>NOA</td><td colspan="2">Number of activities. This has the same meaning as NA, which was defined by García et al. [Gar+03], see Table B.5.</td></tr>
<tr><td>NOAC</td><td colspan="2">Number of activities and control flow elements.This metric requires the process to be well-structured as defined by van der Aalst [van98].</td></tr>
<tr><td>NOAJS</td><td colspan="2">Number of activities joins and splits.This metric can be used with processes that are not well-structured.</td></tr>
<tr><td>MCC</td><td colspan="2">Mccabe's cyclomatic number. This has the same meaning as S, which was defined by Latva-Koivisto [Lat01], see Table B.4.
$$\mathtt{MCC} = e - n + 2$$
where
n is the number of activities (NOA)
e is the number of edges of the graph.</td></tr>
<tr><td>CFC</td><td colspan="2">Control-flow complexity. [Car05b]</td></tr>
<tr><td>N</td><td>Process length.
$$\mathtt{N} = n1 * \log_2(n1) + n2 * \log_2(n2)$$</td><td rowspan="3">Adapted from Halstead [Hal87], where,
$n1$ is the number of unique activities, splits, joins, and control-flow elements.
$n2$ is the number of unique data variables that are manipulated.
$N1$ is the total number of activities, splits, joins, and control-flow elements.
$N2$ is the total number of data variables that are manipulated.</td></tr>
<tr><td>V</td><td>Process volume.
$$\mathtt{V} = (N1 + N2) * \log_2(n1 + n2)$$</td></tr>
<tr><td>D</td><td>Process difficulty.
$$\mathtt{D} = \left(\frac{n1}{2}\right) * \left(\frac{N2}{n2}\right)$$</td></tr>
<tr><td>CNC</td><td colspan="2">Coefficient of network complexity. This has the same meaning as $\mathtt{CNC}_P$, which was defined by Latva-Koivisto [Lat01], see Table B.4.</td></tr>
<tr><td>RT</td><td colspan="2">Restrictiveness estimator. This has the same meaning as RT, which was defined by Latva-Koivisto [Lat01], see Table B.4.</td></tr>
</table>

Cardoso [Car06b] (2006) empirically validated the `CFC` metric by conducting a quantitative experiment, where the subjects were 19 graduate students studying Computer Science, and the instrument contained 22 business processes with known `CFC`. The subjects were asked to rate the control-flow complexity. He used the Spearman's correlation to analyze the data collected from the subjects, and found that `CFC` was highly correlated to the perceived control-flow complexity. This is the same experiment reported by Cardoso [Car08].

Gruhn and Laue [GL06b] (2006) were concerned with the process models for communication between stakeholders and the software developers implementing process technology. This allowed them to focus on metrics in order to measure whether or not a process model is easy or difficult to understand. They used ideas from software complexity to derive process model metrics. This paper tried to describe the metrics in an agnostic way, but some of them are specific to a modeling notation. They expected most metrics to be applicable to EPC, Unified Modeling Language Activity Diagram (UML AD) [OMG09c], BPMN, and Yet Another Workflow Language (YAWL) [Ter+10]. For example, `MaxND` and `MeanND` are natural for EPC, because it requires a well-structured process. Table B.11 provides their derived metrics.

Table B.11: Gruhn and Laue's 2006 metrics

Metric	**Description**
`NA`[1]	Number of activities. This has the same meaning as `NA`, which was defined by García et al. [Gar+03], see Table B.5.
`CFC`	Control-flow complexity. As defined by [Car05d]
`MaxND`[1]	Maximum nesting depth. Nesting depth is defined as the number of decisions in the control-flow that are necessary to reach an activity. Only valid for well-structured processes because it requires proper nesting [van98]. However, counting the number of decisions provides an adequate alternative that is applicable to all type of processes with control-flow
`MeanND`[1]	Mean nesting depth. Calculate the nesting depth for each activity in the process, sum them and divide by `NA`.
`NH`[1]	Number of handles. Count the number of handles in the Petri Net that describes the process (Workflow Net (WF-Net) [van95]). Petri Nets handlers are defined in [van98].

Continued on next page

[1]The author did not provided a symbol or abbreviation for this metric

Table B.11 – *Continued from previous page*

Metric	Description
CW[2]	Cognitive weight. Using the work by Shao and Wang [SW03] they propose the creation of a cognitive weight metric by defining weights for control structures. This paper did not provide the weight values and suggested further research to identify those values. In Gruhn and Laue [GL06a] the metric is formalized for YAWL by providing the corresponding weights (CC_{YAWL}).
F_iF_o[2]	Fan-in / fan-out. Using the work of Henry and Kafura [HK81] they propose $$F_iF_o = ((fan-in) * (fan-out))^2$$

Mendling et al. [Men+06] (2006) converted 604 non-trivial process models from EPC into YAWL models and analyzed the resulting models. They found that 5.6% of the models contained errors. In addition, they defined a set of 15 metrics, mostly counters of model elements, and used regression to find predictors of the errors among the metrics. By using all of the metrics together and using a multivariate logic model they were able to predict a specific set of errors 95.2% of the time using a subset of their metrics. The defined metrics are provided in Table B.12.

Table B.12: Mendling et al.'s 2006 metrics

Metric	Description
A	Number of arcs.
AND_j	Number of and joins.
AND_s	Number of and splits.
Cycle	Cycle. Describes if the model has cycles. Seems to be a binary metric, but the exact calculation is not described.
E_{end}	Number of end events. This has the same meaning as TNEE, which was defined by Rolón et al. [Rol+06b], see Table B.9.
E_{int}	Number of internal events.
E_{start}	Number of start events. This has the same meaning as NTSE, which was defined by Rolón et al. [Rol+06b], see Table B.9.
F	Number of functions.
JC	Join complexity. It seems to be calculated as the summation of the number of incoming arcs into joins, but the calculation not clearly stated.

Continued on next page

[2]The author did not provided a symbol or abbreviation for this metric

Table B.12 – *Continued from previous page*

Metric	Description
JSR	Join-split-ratio. $$\texttt{JSR} = \frac{\texttt{JC}}{\texttt{SC}}$$
$\texttt{OR}_j$	Number of or joins.
$\texttt{OR}_s$	Number of or splits.
SC	Split complexity. This has the same meaning as CFC, which was defined by Cardoso [Car05b], see Table B.8. Same as the control-flow complexity as defined by Cardoso [Car05b] CFC.
$\texttt{XOR}_j$	Number of xor joins.
$\texttt{XOR}_s$	Number of xor splits.

Mendling [Men06] (2006) recognized that most metrics that are defined are unbound, because they consist of just counting elements in the notation, which makes it impossible to compare two processes. He considered social network analysis and identified density as an appropriate metric, and proposed it as a complexity metric that produces an output between zero and one, where zero is not complex and one represents maximum complexity. This made it possible to compare the complexity of different process models. He provided four versions of the metric described in Table B.13.

Table B.13: Mendling's 2006 second set of metrics

Metric	Description
$\texttt{d}_w$	Density of a workflow-net. [van97]. For $W = (P, \texttt{T}, \texttt{A})$ with P is the set of places, T the set of transitions, and $\texttt{A} \subseteq (P \times \texttt{T}) \cup (\texttt{T} \times P)$ $$\texttt{d}_w = \frac{\lvert\texttt{A}\rvert}{\lvert P\rvert * \lvert\texttt{T}\rvert + \lvert\texttt{T}\rvert * \lvert P\rvert}$$
$\texttt{d}_y$	Density of a ywal model. [vT05]. For $Y = (C, \texttt{T}, \texttt{A})$ with C is the set of conditions, T the set of tasks, and A the set of arcs $$\texttt{d}_y = \frac{\lvert\texttt{A}\rvert}{\lvert C\rvert * \lvert\texttt{T}\rvert + \lvert\texttt{T}\rvert * \lvert C\rvert + \lvert\texttt{T}\rvert * \lvert\texttt{T}\rvert}$$
$\texttt{d}_e$	Density of an epc. [van99]. For $EPC = (E, \texttt{F}, C, l, \texttt{A})$, where E is the set of events, F the set of functions, l a mapping from connector onto the connector label, and A the set of arcs. The small letters n, e, f, c, and a represents the number of nodes, events, functions, connectors, and arcs respectively.

Continued on next page

Table B.13 – *Continued from previous page*

Metric	Description
	Note that $a_{min} = n - 1$. Depending on the number of connectors, we can calculate d_{even} for even number of connectors, or d_{odd} for an odd number of connectors as $$\mathtt{d}_e = \begin{cases} d_{even} = \frac{a - a_{min}}{c_{maxeven} + 2*(e+f) - a_{min}}, & \text{if even connectors} \\ d_{odd} = \frac{a - a_{min}}{c_{maxodd} + 2*(e+f) - a_{min}}, & \text{if odd connectors} \end{cases}$$ Where, $c_{maxeven} = \left(\frac{c}{2} + 1\right)^2$ $c_{maxodd} = \left(\frac{c-1}{2} + 1\right)^2 + \frac{c-1}{2} + 1$ $c_{c \leq 1} = 1$ $d_{c \leq 1} = 0$

Gruhn and Laue [GL06a] (2006) defined complexity metrics as measurements that can tell whether a model is easy or difficult to comprehend. They formalized a cognitive weight metric for YAWL models. The cognitive weight metric that was proposed in Gruhn and Laue [GL06b] is defined in detail in this paper, by providing a formal definition and a table of cognitive weights for YAWL modeling elements. They recognized that their cognitive metric for process models does not account for layout and textual complexity of the model. Table B.14 describes the metric.

Han and Zhang [HZ09] (2009) provided theoretical validation for the cognitive complexity for YAWL $\mathtt{CC}_{\mathtt{YAWL}}$ [GL06a].

Gruhn and Laue [GL07] (2007) identified five factors that influence the control-flow complexity of a Business Process Management (BPM) model, these factors are described in Table B.15. In addition, they defined a few metrics described in Table B.11.

Table B.14: Gruhn and Laue's 2006 second set of metrics

Metric	Description
$\mathtt{CC}_{\mathtt{YAWL}}$[3]	Cognitive complexity for yawl. This has the same meaning as `CW`, which was defined by Gruhn and Laue [GL06b], see Table B.11. is based on [SW03] software cognitive complexity metric and [GL06b][4] $$\mathtt{CC}_{\mathtt{YAWL}} = \sum W_{\text{YAWL}}$$ Where,

Continued on next page

[3]The author did not provided a symbol or abbreviation for this metric
[4]Equation was described by Gruhn and Laue [GL06a] but not formally presented

Table B.14 – *Continued from previous page*

Metric	Description
	cognitive weight, W_{YAWL} is given by the following table

Software control structure	**BPM Control structure**	$W_{\texttt{YAWL}}$
Sequence	Consecutive steps in a workflow	1
Branching with if-then	XOR-split with corresponding XOR-join (one of two branches is selected)	2
Branching with case (arbitrary number of selectable cases)	XOR-split with corresponding XOR-join (one of ≥ 3 branches is selected)	3
Execution of control-flow in parallel	AND-split with corresponding AND-join	4
Branching with case, followed by parallel execution	OR-split with corresponding OR-join	7
Call of a user defined function	Composite task (subtask)	2
Branching, followed by parallel execution	Multiple instance activity	6
Cancel activity	Cancellation (by activating an activity one deactivates another one)	1
cancel case (comparable to a function call)	Cancellation (by activating one deactivates all elements with in another part of the model)	2
	Cancellation (by activating one deactivates all elements with in another part of the model including nested elements)	3

Mendling's [Men07] (2007) PhD dissertation focused on metrics that helped predict formal errors in EPC process models. He considered comprehensibility to be the main determinant for probability of errors in an EPC model. He used a sample of 2003 EPC models developed by practitioners. Using tools to find errors in the models, he identified 215 models with at least one error. Most of the metrics, used as a group, were able to predict which models had errors, with the exception of the density metric (Δ), and the average connector degree ($\acute{d_C}$).

Table B.15: Gruhn and Laue's 2007 factors that influence the control-flow complexity

Factor	**Metric**	**Description**
Size of the model	lines of code	Suggest the use of the size metrics described by Cardoso et al. [Car+06]
Control-flow complexity of the model	McCabe-Metric	Suggest the use of the `CFC` as described by Cardoso [Car05d]
Structure of the model	nesting depth	No formal metric is proposed
Structure of the model	jumps out of a control structure	Suggest the use of the `JSR` metric as described by Mendling et al. [Men+06]
Comprehensiveness of the model	cognitive complexity metrics	Suggest the use of the cognitive weight metric as described by Gruhn and Laue [GL06a]
(Anti)Patterns for BPM	(Anti)Patterns for BPM	No formal metric is proposed, but mention the BPM implicit termination anti-pattern described in [van+03]
Modularization of the model	Fan-in / Fan-out	Adapts the metric defined by Henry and Kafura [HK81] and arrives to the same metric as Gruhn and Laue [GL06b] $fan-in/fan-out = ((fan- \in)*(fan-out))^2$

The metrics described by Mendling [Men07] were based on a graph that represented the EPC. The graph was defined as $G = (\mathtt{N}, \mathtt{A})$, where `N` represented the nodes. The nodes could be tasks (`T`), splits (`S`), or joins (J), and so $\mathtt{N} = \mathtt{T} \cup \mathtt{S} \cup J$. `A` represents the arcs, and so $\mathtt{A} \subseteq \mathtt{N} \times \mathtt{N}$. In addition, he used the term *connector* to represent splits and joins, and so $C = \mathtt{S} \cup J$. Based on his previous work [Men+06], he concluded that complexity has an impact on error probability. The metrics are defined in Table B.16.

Mendling and Neumann [MN07] (2007) concluded that error probability increases with size, and decreases with higher separability or structuredness. Therefore, the set, size, separability, and structuredness can be used to predict errors in EPC. The metrics and the results from the studies are also discussed in Mendling and Neumann [MN07] and Mendling et al. [Men+07a; Men+07b]. Mendling [Men08] (2008) described the same work as Mendling [Men07].

Table B.16: Mendling's 2007 metrics

Type	**Metric**	**Description**
Size	`diam`	Diameter. It is defined as the length of the longest path from a start node to an end node
	$\texttt{S}_A$	Number of arcs. This has the same meaning as `A`, which was defined by Mendling et al. [Men+06], see Table B.12.
	$\texttt{S}_C$	Number of connectors. This has the same meaning as `NDWP`, which was defined by García et al. [Gar+04a], see Table B.6.
	$\texttt{S}_{C_{and}}$	Number of and connectors.
	$\texttt{S}_{C_{or}}$	Number of or connectors.
	$\texttt{S}_{C_{xor}}$	Number of xor connectors.
	$\texttt{S}_E$	Number of events. This has the same meaning as `TNE`, which was defined by Rolón et al. [Rol+06b], see Table B.9.
	$\texttt{S}_{E_E}$	Number of end events. This has the same meaning as `TNEE`, which was defined by Rolón et al. [Rol+06b], see Table B.9.
	$\texttt{S}_{E_S}$	Number of start events. This has the same meaning as `NTSE`, which was defined by Rolón et al. [Rol+06b], see Table B.9.
	$\texttt{S}_F$	Number of functions. This has the same meaning as `F`, which was defined by Mendling et al. [Men+06], see Table B.12.
	$\texttt{S}_{j_{and}}$	Number of and joins. This has the same meaning as $\texttt{AND}_j$, which was defined by Mendling et al. [Men+06], see Table B.12.
	$\texttt{S}_{j_{or}}$	Number of or joins. This has the same meaning as $\texttt{OR}_j$, which was defined by Mendling et al. [Men+06], see Table B.12.
	$\texttt{S}_{j_{xor}}$	Number of xor joins. This has the same meaning as $\texttt{XOR}_j$, which was defined by Mendling et al. [Men+06], see Table B.12.
	$\texttt{S}_N$	Number of nodes. This has the same meaning as `NA`, which was defined by García et al. [Gar+03], see Table B.5.
	$\texttt{S}_{S_{and}}$	Number of and splits. This has the same meaning as $\texttt{AND}_s$, which was defined by Mendling et al. [Men+06], see Table B.12.
	$\texttt{S}_{S_{or}}$	Number of or splits. This has the same meaning as $\texttt{OR}_s$, which was defined by Mendling et al. [Men+06], see Table B.12.
	$\texttt{S}_{S_{xor}}$	Number of xor splits. This has the same meaning as $\texttt{XOR}_s$, which was defined by Mendling et al. [Men+06], see Table B.12.

Continued on next page

Table B.16 – *Continued from previous page*

Type	**Metric**	**Description**
Partitionability	$\prod$	Separability ratio. $$\Pi(G) = \frac{\lvert\{n \in \mathtt{N} \lvert n \text{ is cut-vertex}\}\rvert}{\lvert\mathtt{N}\rvert - 2}$$
	Λ	Maximum depth of all nodes. $$\Lambda(G) = \max\{\lambda(n) \lvert n \in \mathtt{N}\}$$ Where, $$\lambda(n) = \min(\lambda_{in}(n), \lambda_{out}(n))$$ $\lambda_{in}(n)$ is calculated using an algorithm that evaluates nesting on $\mathtt{S}$ (split nodes) starting from the start nodes. $\lambda_{out}(n)$ is calculated using an algorithm that evaluates reverse nesting on J (join nodes) starting on the termination nodes.
	Ξ	Sequentiality ratio. $$\Xi(G) = \frac{\lvert\mathtt{A} \cap (\mathtt{T} \times \mathtt{T})\rvert}{\lvert\mathtt{A}\rvert}$$
	Φ	Structuredness ratio. $$\Phi(G) = 1 - \frac{\mathtt{S_N}(G')}{\mathtt{S_N}(G)}$$ Where, G' is a reduced process graph calculated by an algorithm proposed by Mendling [Men07, p 118].
Density	CNC	Coefficient of network connectivity. This has the same meaning as $\mathtt{CNC}_P$, which was defined by Latva-Koivisto [Lat01], see Table B.4. gives the ratio of arcs to nodes. $$\mathtt{CNC}(G) = \frac{\lvert\mathtt{A}\rvert}{\lvert\mathtt{N}\rvert}$$
	$\widehat{\mathtt{d}_C}$	Maximun degree of a connector. $$\widehat{d_C}(G) = \max\{d(c) \mid c \in C\}$$ Where, $d(c)$ is the degree of connectors for node C which is the number of outgoing arcs from c.

Continued on next page

Table B.16 – *Continued from previous page*

Type	Metric	Description
	$\overline{\mathtt{d}_C}$	Average degree of connectors. $$\overline{\mathtt{d}_C}\,(G) = \frac{1}{\lvert C\rvert}\sum_{c\in C} d(c)$$
	Δ	Density of the process graph. refers to the number of arcs divided by the number of the maximum number of arcs for the same number of nodes. $$\Delta\,(G) = \frac{\lvert \mathtt{A}\rvert}{\lvert \mathtt{N}\rvert \cdot (\lvert \mathtt{N}\rvert - 1)}$$
Cyclicity.	CYC	Cyclicity. $$\mathtt{CYC}\,(G) = \frac{\lvert \mathtt{N}_C\rvert}{\lvert \mathtt{N}\rvert}$$ Where, $$\lvert \mathtt{N}_C\rvert = \sum_{n\in\{n\in\mathtt{N}\vee n\hookrightarrow n\}} 1$$
Connector Interplay	MM	Connector mismatch. $$\mathtt{MM}\,(G) = \mathtt{MM}_{or} + \mathtt{MM}_{xor} + \mathtt{MM}_{and}$$ Where, $$\mathtt{MM}_l = \left\lvert \sum_{c\in\mathtt{S}_l} d\,(c) - \sum_{c\in J_l} d\,(c)\right\rvert$$ $l\in\{and, xor, or\}$
	CFC	CFC as defined by Cardoso [Car05b]
	CH	Connector heterogeneity. $$\mathtt{CH}\,(G) = -\sum_{l\in\{and,or,xor\}} p(l)\cdot \log_3\,(p(l))$$ Where, $$p\,(l) = \frac{\lvert C_l\rvert}{\lvert C\rvert}$$
Concurrency	TS	Token split. $$\mathtt{TS}\,(G) = \sum_{n\in\mathtt{S}_{or}\cup\mathtt{S}_{and}} d\,(n) - 1$$

Cardoso [Car07b] (2007) applied his original control-flow complexity metric that was introduced in Cardoso [Car05b; Car05d] to BPEL processes. Although, he mentioned BPEL in

Cardoso [Car05a], his initial focus was on workflow processes. This paper refines the metric to different types of activities in a BPEL process, and calls it $\mathtt{CFC}^{\mathtt{BPEL}}_{\mathtt{Process}}$. Table B.17 describes the metric.

Table B.17: Cardoso's 2007 metrics

Metric	**Description**
$\mathtt{CFC}^{\mathtt{BPEL}}_{\mathtt{Process}}$	Control-flow complexity for bpel process. $$\mathtt{CFC}^{BPEL}_{Process}(P) = \sum_{a \in P} \mathtt{CFC}^{BPEL}_{Act}(a)$$ Where, a is an activity of process P. $\mathtt{CFC}^{\mathtt{BPEL}}_{\mathtt{Act}}(a)$is the Activity flow complexity for bpel processdefined as $$\mathtt{CFC}^{BPEL}_{Act}(a) = \begin{cases} a\text{= basic activity} & \Rightarrow 1 \\ a\text{= sequence}(\mathtt{S}) & \Rightarrow \sum_{a \in \mathtt{S}} \mathtt{CFC}^{BPEL}_{Act}(a) \\ a\text{= switch}(Sw) & \Rightarrow \lvert Sw \rvert \times \sum_{a \in Sw} \mathtt{CFC}^{BPEL}_{Act}(a) \\ a = \text{while}(W) & \Rightarrow \log_2 \left(\mathtt{CFC}^{BPEL}_{Act}(a) + 2 \right) \times \mathtt{CFC}^{BPEL}_{Act}(a) \\ a = \text{flow}(\mathtt{F}) & \Rightarrow (\lvert \mathtt{F} \rvert - l)! \times \sum_{a \in \mathtt{F}} \mathtt{CFC}^{BPEL}_{Act}(a) \\ a = \text{pick}(Pk) & \Rightarrow \left(2^{\lvert Pk \rvert} - 1 \right) \times \sum_{a \in Pk} \mathtt{CFC}^{BPEL}_{Act}(a) \end{cases}$$ l is the number of cross boundary links

Parizi and Ghani's [PG08] paper (2008) presented a subset of Cardoso's [Car07b] paper describing the same metrics and arriving at the same conclusions.

Cardoso [Car07a] (2007) introduced a design time BPMN complexity metric based on the number of distinct traces that a process can generate. Processes with a single activity will always generate the same trace. The worst case scenario is a process with n activities that can be executed in any order, and so can generate $n!$ traces. The calculation of the metric used the workflow patterns defined by van der Aalst et al. [van+00; van+03]. Table B.18 describe this metric.

Vanderfeesten et al. [Van+07a] (2007) defined a coupling metric for EPC process models. Although they did not categorize the metric as a complexity metric, but rather as a coupling metric, they indicated that it could be used to identify problems in the understandability and maintainability of process models [Van+07a]. The metric is defined with no validation as described in Table B.19.

Table B.18: Cardoso's 2007 second set of metrics

Metric	Description
$\mathtt{LBC}_T$	Log-based complexity. The metric is not fully defined, but it is based on the 20 patterns presented in van der Aalst et al. [van+00; van+03]. The applications of the patters follows a described but not formalized algorithm. The calculation for the five patterns described in the paper is as $$LBC_T(wf) = \sum_{i=1}^{m} LBC_{xi}(wf)$$ $$LBC_{P1}(wf) = \prod_{i=1}^{n} LBC_{xi}(wf_i)$$ $$LBC_{P4}(wf) = LBC_{P16}(wf) = \sum_{i=1}^{n} p_i \times LBC_{xi}(wf_i)$$ $$LBC_{P10}(wf) = \left(\sum_{j=0}^{L-1} p^j(1-p) \times j \times LBC_x(wf_2)\right) + (p^L(1-p) + p^{L+1}) \times L \times LBC_x(wf2)$$ $$LBC_{P17}(wf) = n! \times \prod_{i=1}^{n} LBC_{xi}(wf_i)$$ Where, m is the number of patterns presents in the BPMN model. The algorithm should reduce the process by applying the patterns from the inside of the process, and collapsing each pattern into a single node. xi represents a pattern. n is the number of paths. L is the maximum number of iterations.

Mendling and Strembeck [MS08] (2008) conducted an online survey as an empirical experiment on process understandability using EPC notation. They used a subset of the metrics defined by Mendling [Men07]. They focused the empirical experiment on how activity labels might influence understandability. Understandability was operationalized by using the sum of correct answers provided by the participants. They found that theoretical understanding of process modeling, separability ($\prod$), and the use of labels correlated with understandability.

Table B.19: Vanderfeesten et al.'s 2007 metrics

Metric	Description
CP	Coupling. $$\texttt{CP} = \frac{\sum_{t1,t2 \in \texttt{T}} connected(t1,t2)}{\lvert\texttt{T}\rvert * (\lvert\texttt{T}\rvert - 1)}$$ Where, $$connected\,(t1,t2) = \begin{cases} 1, & if(t1 \rightarrow t2) \wedge (t1 \neq t2) \\ 1, & if(t1 \rightarrow AND \rightarrow t2) \wedge (t1 \neq t2) \\ \frac{1}{(2^m-1)\cdot(2^n-1)} + \frac{(2^m-1)\cdot(2^n-1)-1}{(2^m-1)\cdot(2^n-1)} \cdot \frac{1}{m \cdot n}, & if(t1 \rightarrow OR \rightarrow t2) \wedge (t1 \neq t2) \\ \frac{1}{m \cdot n}, & if(t1 \rightarrow XOR \rightarrow t2) \bigwedge (t1 \neq t2) \\ 0, & if(t1 = t2) \end{cases}$$ $t1$ and $t2$ are activities m is the number of incoming arcs n is the number of outgoing arcs

Cardoso [Car08] (2008) introduced relative control-flow complexity $\texttt{CFC}_{rel}$, and renamed his original control-flow complexity CFC [Car05b] to absolute control-flow complexity $\texttt{CFC}_{abs}$. The paper presented the same theoretical validation of CFCthat was provided in Cardoso [Car05b] (2005), and exactly the same experiment that was done in [Car06b] (2006). The empirical validation [Car06b] was based on an experiment using 19 subjects and 22 processes. The results seem to supports CFC as a complexity metric. The metrics are defined in Table B.20.

Table B.20: Cardoso's 2008 metrics

Metric	Description
$\texttt{CFC}_{abs}$	The Absolute control-flow complexity. This has the same meaning as CFC, which was defined by Cardoso [Car05b], see Table B.8. $$\begin{aligned} \texttt{CFC}_{abs}(p) = & \sum_{i \in \{\text{XOR-splits of p}\}} \texttt{CFC}_{XOR-split}(i) \\ & + \sum_{j \in \{\text{OR-splits of p}\}} \texttt{CFC}_{OR-split}(j) \\ & + \sum_{k \in \{\text{AND-splits of p}\}} \texttt{CFC}_{AND-split}(k) \end{aligned}$$

Continued on next page

Table B.20 – *Continued from previous page*

Metric	Description
	Where, $\mathtt{CFC}_{XOR-split}(XOR\ \ activity) = n$ $\mathtt{CFC}_{OR-split}(OR\ \ activity) = 2^n - 1$ $\mathtt{CFC}_{AND-split}(AND\ \ activity) = 1$ n is the fan-out of the split.
$\mathtt{CFC}_{rel}$	Relative control-flow complexity. $$\mathtt{CFC}_{rel}(p) = \frac{\mathtt{CFC}_{abs}(p)}{\lvert\{\text{XOR-splits of p}\} \cup \{\text{OR-splits of p}\} \cup \{\text{AND-splits of p}\}\rvert}$$

Vanderfeesten et al. [Van+08b] (2008) introduced the cross-connectivity metric (CC) based on insights from cognitive research into visual programming languages. They used the same data set used by Mendling [Men07] to validate the metrics. With 99% significance they concluded that a decrease of CC implies an increase in error probability. They also evaluated understandability, for which they used 73 students, and 12 models (the same data used by Mendling et al. [Men+07c]). They compared the CC with some of the metrics defined by Mendling [Men07], and found that density (Δ), and the average connector degree ($\acute{d_C}$) were better predictors of understandability. They concluded that CC could be used to improve the explanatory power of the other metrics. The cross-connectivity metric is defined in Table B.21. They claimed that the metric was generic, because it captured the routing elements that could be expressed with standard process modeling languages such as EPCs, UML AD, Petri Nets, BPMN, or YAWL.

Table B.21: Vanderfeesten et al.'s 2008 second set of metrics

Metric	Description
CC	Cross-connectivity. $$\mathtt{CC} = \frac{\sum_{n1,n2 \in \mathtt{N}} \mathtt{V}(n1, n2)}{\lvert\mathtt{N}\rvert \cdot (\lvert\mathtt{N}\rvert - 1)}$$ Where, Value of a connection, $$\mathtt{V}(n1, n2) = \max_{p \in P_{n1,n2}} v(p)$$ Value of a path, $$v\,(p) = \prod_{a_i \in p} W(a_i)$$

Continued on next page

Table B.21 – *Continued from previous page*

Metric	Description
	Weight of an arc, $$W(a) = w(src(a)) \cdot w(dest(a))$$ Weight of a node, $$w(n) = \begin{cases} 1, & if\ n \in C \wedge n = AND \\ \frac{1}{d}, & if\ n \in C \wedge n = XOR \\ \frac{1}{2^d-1} + \frac{2^d-2}{2^d-1}, & if\ n \in C \wedge n = OR \\ 1, & if\ n \in \mathtt{T} \end{cases}$$ $\mathtt{N}$ set of nodes $\mathtt{N} = \{\mathtt{T} \cup C\}$ n is a node $\{n \in \mathtt{N}\}$ $\mathtt{T}$ is a set of tasks t is a task $\{t \in \mathtt{T}\}$ C is a set of connectors c is a connector $\{c \in C\}$ d is the degree of the node, which is the total number of incoming and outgoing arcs from the node $src(a)$ is a source node $dest(a)$ is a destination node

Kreimeyer et al. [Kre+08] (2008) presented three metrics, namely McCabe's Cyclomatic number [Car+06] adapted to EPC, Cardoso's control-flow complexity[Car05b], and Activity/Passivity. However, Activity/Passivity is not fully described or formalized in their work. Therefore for this review this paper is considered a duplicate of Cardoso et al. Cardoso; Cardoso et al.

Cheng [Che08] (2008) described four complexity metrics (Structural complexity, Interaction complexity, Usability complexity, and Total operational complexity) using a graph. The metrics were based on entropy. Some of the metrics required information not present in the process model, therefore they were not included in this review. For example, the interaction complexity was based on interaction between actors, including human-to-human, human-to-machine, and machine-to-machine; and was calculated based on an interaction diagram instead of the process model. The usability complexity required information on how a user executes a task that is not present in the process model (for example, which buttons to click on the screen). The total operational complexity was calculated based on the other three metrics. The metric used in this review is described in Table B.22.

Table B.22: Cheng's 2008 metrics

Metric	**Description**
$\mathtt{H}_{SC}$	Structural complexity. $$H_{SC} = aH_{seq} + bH_{split} + cH_{merge} + dH_{mix}$$ Where, $$H_i = -\sum_{j=1}^{m} p_{ij} \log_2 p_{ij}$$ i is a set of sequence, split, merge, or mix. p_{ij} is the normalized set of processing requirements, with the constraint that $\sum i = 1^r \sum_{j=1}^{m} p_{ij} = 1$. a is the number of structures with one incoming arc and one outgoing arc. b is the number of split structures with one incoming arc and more than one outgoing arcs. c is the number of merge structures with more than one incoming arcs and one outgoing arc. d is the number of sequence structures with more than one incoming arcs and more than one outgoing arcs.

Chen and Prabhu [CP08] (2008) proposed a complexity metric inspired by both software science [Hal87] and graph theory. The metrics combined ideas from Halstead [Hal87] and McCabe [McC76] into a single metric. The article includes a quick survey of complexity metrics in software engineering that could be adapted to process models. Table B.23 describes the proposed metric.

Table B.23: Chen and Prabhu's 2008 second set of metrics

Metric	**Description**
`Complexity`	Variety-based complexity. $$\texttt{Complexity} = \frac{\rho \times \eta_1 \times \eta_2}{N_1}$$ Where, ρ is the number of possible paths. η_1 is the number of unique pattern types. η_2 is the number of unique node types. N_1 is the number of dificult paths.

Vanderfeesten et al. [Van+08a] (2008) proposed three cohesion and coupling metrics to guide the evaluation of generic workflow models. Their concern was with the resources required to execute the operations required in an activity. They claimed that based on the similarities between workflow processes and software programs, loose coupling of activities should result in reduced probability of runtime errors, and highly cohesive activities are likely to be better understood and performed by people. They implemented the metrics in a tool and used a workflow process as a use case to evaluate, based on their intuition, the usability of the metrics. The defined metrics are provided in Table B.24.

Table B.24: Vanderfeesten et al.'s 2008 metrics

Metric	Description
`ch`	Process cohesion. This has the same meaning as `c`, which was defined by Reijers and Vanderfeesten [RV04], see Table B.7. $$ch = \frac{\sum_{t \in \bar{S}} c(t)}{\left\|\bar{S}\right\|}$$ Where, $\bar{S}$ is a set of operation ($\bar{S} = \{t \mid \exists S \text{ such that } (t, e) \in S\}$) S is a process t is an operation e is a resource able to execute the operation $c(t)$ is the activity cohesion, defined as $$c(T) = \lambda(T) \times \mu(T)$$ $T = (t, e)$ is an activity $\lambda(T)$ is the activity relation cohesion $\mu(T)$ is the activity information cohesion
`cp`	Process coupling. This has the same meaning as `k`, which was defined by Reijers and Vanderfeesten [RV04], see Table B.7. $$ch = \begin{cases} \frac{\left\|\{(T_1,T_2) \in S \times S \mid \bar{T}_1 \neq \bar{T}_a \vee (\hat{T}_1 \cap \hat{T}_2) \neq \varnothing\}\right\|}{\left\|\bar{S}\right\| \times (\left\|\bar{S}\right\| - 1)} & \text{for } \|S\| > 1 \\ 0 & \text{for } \|S\| \leq 1 \end{cases}$$ Where, $\bar{T}$ is the input-output relations in an activity $\hat{T}$ is the number of information elements processed in an activity
ρ	Process coupling/cohesion ratio. This has the same meaning as ρ, which was defined by Reijers and Vanderfeesten [RV04], see Table B.7. $$\rho = \frac{cp}{ch}$$

Lassen and van der Aalst [Lv09] (2009) described four complexity metrics that they implemented in a process analysis tool. They used WF-Net as the notation, and asserted that the results could be applicable to other process modeling notations, like EPC, BPMN, flow charts, UML AD, etc. In addition, they tested the metrics using 262 models collected from student projects. After comparing the metrics between the model using correlation they concluded, based on intuition, that the structuredness metric (`SM`) outperformed the other metrics. Their complexity metrics are described in Table B.25.

Table B.25: Lassen and van der Aalst's 2009 metrics

Metric	Description
`ECaM`	Extended cardoso. is a reformulation of `CFC` [Car05b] using WF-Net, as follows $$\texttt{ECaM}(PN) = \sum_{p\in P} ECFC_p(p)$$ Where, $PN = (P, \texttt{T}, \texttt{F})$ is a WF-Net, defined as follows P is a finite set of places used to route the flow $\texttt{T}$ is a finite set of transitions ($P\cap\texttt{T} = \varnothing$), corresponding to workflow tasks. Transitions representing forks and joins maybe needed to model parallel tasks. $\texttt{F}\subseteq(P \times \texttt{T})\cup(\texttt{T} \times P)$ is a set of arcs There is one source place, $i\in P$ such that $\circ i = \varnothing$ There is one sink place, $o\in P$ such that $o\circ = \varnothing$ Every node $x\in P\cup\texttt{T}$ is in a path from i to o $\circ x$ denotes the set of input places for $x\in P\cup\texttt{T}$ $x\circ$ denotes the set of output places for $x\in P\cup\texttt{T}$ $ECFC_p : P \to \texttt{N}$ an auxiliary function for any $p\in P$, defined as $$ECFC_p(p) = \|\{t\circ \mid t\in p\circ\}\|$$
`ECyM`	Extended cyclomatic. It is a reformulation of McCabe [McC76], as follows $$\texttt{ECyM}(PN) = \|E\| - \|\texttt{V}\| + p$$ Where, $E = \{(M_1, M_2)\in\texttt{V} \times \texttt{V} \mid M_1 \to M_2\}$ $\texttt{V} = \left\{M \mid i \xrightarrow{*} M\right\}$, where i is a source place in PN such that $i\in P$ such that $\circ i = \varnothing$ $M\in P \to \texttt{N}$ $M_1 \to M_2$ indicates that M_1 can transition into M_2 $M_1 \xrightarrow{*} M_2$ indicates that M_2 can be reached from M_1

Continued on next page

Table B.25 – *Continued from previous page*

Metric	**Description**
	Both, E and $\mathtt{V}$ form the reachability graph $G = (\mathtt{V}, E)$ for the WF-Net PN p is the number of strongly components in G

`SM`	Structuredness. It is calculated using an algorithm that reduces a WF-Net while calculating the metric, as follows:
	$X \leftarrow (PN, \tau)$ where $\tau(t) = 1, \forall t \in T$
	while $[X] \neq \varnothing$ **do** (i.e., $X = (PN, \tau)$) contains a non-trivial component)
	    pick C so that $\rho_X(C) = min\{\rho_X(C\prime) \| C\prime \in [X]\}$
	    $PN' \leftarrow fold(PN, C)$ where t_c is the added transition
	    $\tau'(t_c) = \omega_X(C)$ and $\tau\prime(t) = \tau(t)$ for all other t
	    $X \leftarrow (PN', \tau\ \prime)$
	end while
	Output $SM(PN) = \tau(t)\ T = \{t\}$ after the net is reduced
	Where,
	$PN = (P, T, F)$ is a WF-Net.
	$\rho_X : [X] \to \mathbb{N}$ is a component priority function that maps components to a value between 1 and 7.
	$\omega_X : [X] \to \mathbb{R}+$ is a component weight function.
	$fold(PN, C) = (P', T', F'$ is a function that replaces C in PN in a single transition.

δ_S	Aggregated depth fraction. $$\delta_S(\alpha_C) = \frac{\sum_{PN \in S} \left(\sum_{C \in \lvert PN \rvert : \rho_P N(C) = \alpha_C} \delta_{PN}(C) \right)}{\sum_{PN \in S} \lvert \{C \in [PN] \mid \rho_{PN}(C) = \alpha_C\} \rvert}$$

Khoshkbarforoushha et al. [Kho+09] (2009) presented two metrics to calculate context dependency. Context dependency is defined as the interaction and knowledge required by BPELprocesses of their context that includes the partners that are involved in the service orchestration that is the focus of BPELprocesses. It was claimed, in this paper, that context dependency should be minimized, due to its negative effects. It was also claimed that heavy coupling and context dependency leads to undesirable effects including poor understandability, inflexibility, inadaptability, and an increased number of defects. The main metric presented by Khoshkbarforoushha et al. [Kho+09] was tested using a controlled experiment using 20 processes (eight processes with multiple versions) and asking three experts to evaluate and select one of the versions of each process. The metrics are defined in Table B.26.

Table B.26: Khoshkbarforoushha et al.'s 2009 metrics

Metric	Description
P_{CM}	Process coupling metric. $$P_{CM} = \sum_{1}^{k} C_{ia} + \sum_{1}^{j} C_i$$ Where, j is the number of structured activities k is the number of interaction activities C_{ia} is the coupling value of interaction activities, defined as $C_{ia} = 1$. The set of interaction activities is: { `invoke`, `reply`, `receive`, `onAlarm`, `onMessage` } C_i is the coupling value imposed by i structured activity, defined as $$C_{sequence} = n$$ $$C_{flow} = n$$ $$C_{switch} = \sum_{i=1}^{n} \left(\frac{1}{n} \times m \right)_i$$ $$C_{pick} = \sum_{i=1}^{n} \left(\frac{1}{n_p} \times m_p \right)$$ $$C_{while} = N_i \times n$$ n is the number of interactions activities with in a sequence, a flow, a switch, a pick or a while statement m is the number interaction activity within a condition i m_p is the number of `onAlarm` and `onMessage` statements n_p is the number of interaction activities within `onAlarm` and `onMessage` statements N_i is the number of loop iterations
P_{CIM}	Process context independency metric. $$P_{CIM} = \frac{1}{P_{CM}}$$

Borgert et al. [Bor+09a] (2009) applied information theory metrics to a generic process graph. They focused on metrics that could be used for the structural complexity of Business Process Models. They adapted four metrics to the generic process graph and used correlations to compare the metrics between the graphs, based on six process models. Table B.27 describes the proposed metrics. Borgert et al. [Bor+09b] (2009) used 125 similar graphs, and tried to identify which of the four metrics could discriminate between them. This paper used the same methodology as that used by Borgert et al. [Bor+09a].

Table B.27: Borgert et al.'s 2009 metrics

Metric	Description
`W`	Wiener's index. $$W(G) = \frac{1}{2}\sum_{i=1}^{n} d(v_i)$$ Where, $$d(v_i) = \sum_{j=1}^{\lvert V\rvert} d(v_i, v_j)$$ G is a finite, undirected, and connected graph $G = (V, E)$ $d(u, v)$ is the shorterst distance between u and v $(u, v \in V)$
`R`	Randi'c's connectivity index. $$R(G) = \sum_{(v_i,v_j)\in E} [\delta(v_i)\delta(v_j)]^{-\frac{1}{2}}$$ Where, $\delta(v)$ is the degree of vertex $v \in V$
`C`	Mccabe's cyclomatic number. This has the same meaning as `ECyM`, which was defined by Lassen and van der Aalst [Lv09], see Table B.25. $$C(G) = \lvert E\rvert - \lvert V\rvert + p$$ Where, p is the number of components
$\mathtt{I}_{f^V}$	Dehmer's graph entropy. $$I_{f^V}(G) = -\sum_{i=1}^{\lvert V\rvert} \rho^V(v_i) \log\left(\rho^V(v_i)\right)$$ Where, $f^V(v) = c_1 \lvert S_1(v, G)\rvert + c_2 \lvert S_2(v, G)\rvert + \cdots + c_{\rho(G)} \left\lvert S_{\rho(G)}(v, G)\right\rvert$ $c_1 = \rho(G), c_2 = \rho(G) - 1, \ldots, c_{\rho(G)} = 1$ $S_j(v_i, G) = \{v \in V \mid d(v_i, v_j) = j, j \geq 1\}$ $\rho^V(v_i) = \frac{f^V(V_i)}{\sum_{j=1}^{\lvert V\rvert} f^V(v_j)}$

Held and Blochinger [HB09] (2009) described a collaborative workflow system that included workflow model analysis. They used workflow process model metrics for the analysis. They reused several metrics from the literature, and introduced new metrics. In addition to the metrics that were well defined in the paper, they mentioned other potential metrics. For the purpose of this review, only the metrics that were included in their two tables with metrics were used in the review, and presented in Table B.28.

Table B.28: Held and Blochinger's 2009 metrics

Metric	Description
`NOA`	Number of activities. This has the same meaning as `NA`, which was defined by García et al. [Gar+03], see Table B.5.
`NOAC`	Number of activities and control. This has the same meaning as `NOAC`, which was defined by Cardoso et al. [Car+06], see Table B.10.
`NOACC`	Number of activities, control structures, and copy.
`MCC`	Mccabe's cyclomatic number. This has the same meaning as `S`, which was defined by Latva-Koivisto [Lat01], see Table B.4. $$MCC = d + 1$$ Where, d is a count of binary decision points, corresponding to count the • number of `Receive` with `createInstance = true` • number of conditional branches, loops, and events • number of logical conjuctions and disjunctions in XPath expressions
`CFC`	Cardoso's control flow complexity. This has the same meaning as `CFC`, which was defined by Cardoso [Car05b], see Table B.8. Adapted to BPEL with `Scope` activities. Define CFC for `Scope` activities as the CFC value of its child activity. Define the value of `Flow` activity as $$CFC(F) = \sum_{a \in F} CFC(a) \text{ iff } l \geq n$$
`DOP`	Degree of parallelism. $$DOP(P) = \sum_{a \in P} DOP(a)$$ Where, $DOP(a)$ is defined for each type of activity as follows, $$\begin{cases} DOP(a) = 1 & \text{basic activities} \\ DOP(F) = \max_{(A \subseteq F) \wedge (\lvert A \rvert = wf)} \sum_{a \in A} DOP(a) & \text{Flow} \\ DOP(A) = \max_{a \in A} DOP(a) & \text{sequence, if, pick} \\ DOP(L) = DOP(a) & \text{Loop, scope} \end{cases}$$

Continued on next page

Table B.28 – *Continued from previous page*

Metric	Description
DFI	Data flow intensity. $$DFI = \frac{\lvert\text{definitions}\rvert + \lvert\text{references}\rvert}{\lvert\text{decisions}\rvert + 1}$$ Where, definitions is the set of assignments references is the set of variables references both in computations and predicates decisions is the set of decision points
SA[5]	Structured activities.
DC[5]	Number of decisions. This has the same meaning as TNG, which was defined by Rolón et al. [Rol+06b], see Table B.9.
AIR[5]	Assign/invoke ratio.
CAR[5]	Copy/assign ratio.

Rolón et al. [Rol+09a] (2009) reused the data that was collected in a series of experiments described in Rolón et al. [Rol+08] (2008), to validate the CFC metric described by Cardoso [Car05b]. They concluded that CFC does correlate with the concepts of understandability and modifiability of BPMNprocess models. Rolón et al. [Rol+09b] (2009) presented the second group of experiments validating the 29 metrics found in Rolón et al. [Rol+05; Rol+06b] (2005), the first group of experiments were presented in Rolón et al. [Rol+08] (2008).

Fu et al. [Fu+10] (2010) proposed a control-flow complexity metric for web service composition processes. They introduced the concept of a structure tree to represent the process and theoretically validated the proposed metric using Weyuker's [Wey88] properties. The resulting metric is similar to control-flow complexity CFC [Car05b; Car05d], with the addition of a new term for loops. Unfortunately, the authors did not realize that in most cases the number of iterations in a loop is unknown at modeling time. Therefore, strictly speaking this metric cannot be used for static analysis, however it was included in this SLR under the assumption. that a maximum number of looping may be acceptable in order to calculate the metric. C is described in Table B.29.

Table B.29: Fu et al.'s 2010 metrics

Metric	Description
C	Sequence control-flow complexity. $$\texttt{C}(P) = \sum_{P_i \in P.sub} C(P_i)$$ Where,

Continued on next page

[5]The author did not provided a symbol or abbreviation for this metric

Table B.29 – *Continued from previous page*

Metric	Description
	$C(P_i)$ can be one of the following $$C_{and}(P) = \sum_{P_i \in P.sub} C(P_i) + 1$$ $$C_{or}(P) = \sum_{P_i \in P.sub} C(P_i) + 2^n$$ $$C_{xor}(P) = \sum_{P_i \in P.sub} C(P_i) + n$$ $$C_{loop}(P) = \sum_{P_i \in P.sub} C(P_i) + m$$ n is the number of substructures or paths in which an 'or' or 'xor' splits m is the number of iterations of the loop[6]

Muketha et al. [Muk+10b] (2010) introduced seven metrics for BPEL processes, provided three examples, and validated them using Briand et al.'s [Bri+95] framework and Weyuker's [Wey88] properties. The metrics are described in Table B.30.

Table B.30: Muketha et al.'s 2010 metrics

Metric	Description
`NOBA`	Number of basic activities. This has the same meaning as `NT`, which was defined by Rolón et al. [Rol+06b], see Table B.9. This is a counter of activities that exclude AND, OR, or XOR activities.
`NOSA`	Number of structured activities. This has the same meaning as `TNG`, which was defined by Rolón et al. [Rol+06b], see Table B.9. This is a counter of the number of AND, OR, and XOR activities.
`IF4BP`	Information flow complexity. Corresponds to $fan-in/fan-out$ in Gruhn and Laue [GL06b]. $$\texttt{IF4BP} = \sum_{m=1}^{n} IF4BP_m$$ Where, $$\texttt{IF4BP}_m = (NOIA * NOOA)^2$$ m is the number of BPEL modules $NOIA$ is the number of input activities $NOOA$ is the number of output activities

Continued on next page

[6]This cannot be known at design time

Table B.30 – *Continued from previous page*

<table>
<tr><th>Metric</th><th>Description</th></tr>
<tr><td>SCBP</td><td>Structural complexity.

$$\texttt{SCBP} = l\,(P) * asc(P)$$

Where,
$l(P)$ is the length of the process, given by the number of activities in the process
$asc(P)$ is the average structural complexity,

$$asc\,(P) = \frac{\sum_{i=1}^{n} l(P_i) * asc(P_i)}{\sum_{i=1}^{n} l(P_i)}$$

N is the number of units in the process (sequences, branches, loops, and parallel blocks)
$asc\,(P_i)$ is given by the following table:
<table>
<tr><th>category</th><th>activity</th><th>asc value</th></tr>
<tr><td>sequence</td><td>sequence</td><td>1.1</td></tr>
<tr><td>branch</td><td>if, pick</td><td>1.3</td></tr>
<tr><td>loop</td><td>while, forEach, repeatUntil</td><td>1.5</td></tr>
<tr><td>parallel</td><td>flow</td><td>1.7</td></tr>
</table>
</td></tr>
<tr><td>CCBP</td><td>Cognitive complexity. It is based on [SW03] software cognitive complexity metric and [GL06b]

$$\texttt{CCBP} = (NOIA + NOOA) * \sum_{c \in \texttt{NOSA}} W_c$$

Where,
Cognitive weight., W_c is given by the following table
<table>
<tr><th>Category</th><th>Activity</th><th>W_c</th></tr>
<tr><td>sequence</td><td>sequence</td><td>1</td></tr>
<tr><td>branch</td><td>if, pick</td><td>2</td></tr>
<tr><td>loop</td><td>while, forEach, repeatUntil</td><td>3</td></tr>
<tr><td>parallel</td><td>flow</td><td>4</td></tr>
</table>
</td></tr>
<tr><td>ALSA</td><td>Average length of structured activity. It is a ratio between basic activities and structured activities

$$\texttt{ALSA} = \texttt{NOBA}/\texttt{NOSA}$$
</td></tr>
<tr><td>ACCSA</td><td>Average cognitive complexity of structured activity. It is a ration between cognitive complexity and structured activities

$$ACSA = \texttt{CCBP}/\texttt{NOSA}$$
</td></tr>
</table>

Mao [Mao10a] (2010) looked at processes in the context of web service technology described using Petri Nets. He proposed a set of metrics to help software maintainers analyze and understand these web service processes. The metrics are defined in Table B.31.

Table B.31: Mao's 2010 metrics

Metric	Description
$\texttt{N}_P$	Number of places. Number of places in the Petri Net
$\texttt{N}_T$	Number of transitions. Number of transitions in the Petri Net
$\texttt{N}_S$	Number of services. Number of Web servixes being invoked by the process
ADP	Average degree of place. $$ADP = \frac{\sum_i deg(p_i)}{\lvert P \rvert}$$ Where, P is the set of places in the Petri Net $p_i \in P$ is the i^{textth} place in the Petri Net $deg(p_i)$ is the degre of nodes (number of adjacent and connected nodes)
ADT	Average degree of transition. $$ADT = \frac{\sum_i deg(t_i)}{\lvert T \rvert}$$ Where, T is the set of transitions in the Petri Net $t_i \in T$ is the i^{textth} transition in the Petri Net $deg(t_i)$ is the degre of nodes (number of adjacent and connected nodes)
TNS	Transfer number per service. $$TNS = \frac{\lvert F \rvert}{N_s}$$ Where, F is the set of directed arcs in the Petri Net
CC	Mccabe's cyclomatic number. This has the same meaning as S, which was defined by Latva-Koivisto [Lat01], see Table B.4. $$CC = \lvert F \rvert - \lvert P \rvert - \lvert T \rvert + 2$$
AEPC	Average execution path complexity. $$AEPC = \sum_{i=1}^{k} (prob(Pt_i) \times C(Pt_i))$$ Where, $C(Pt_i)$ is the execution path complexity for path i, defined as the number of places plus the number of transitions on path i $prob(Pt_i) = 1/k$ where k is the number of execution paths.

Mao [Mao10b] (2010) used the metrics in Mao [Mao10a], and described a new metric based on cognitive informatics [Wan02]. He used the metrics in two processes, and based on intuition concludes that the metrics are effective. His new metric is described in Table B.32.

Table B.32: Mao's 2010 second set of metrics

<table>
<tr><th>Metric</th><th>Description</th></tr>
<tr><td>AEPC_{CI}</td><td>Average execution path complexity based on cognitive informatics.

$$\text{AEPC}_{CI} = \sum_{i=1}^{k} (prob(Pt_i) \times C\prime(Pt_i))$$

Where,
$C\prime(Pt_i)$ is the execution path complexity for path i, defined as the number of places plus the number of transitions on path i multiplied by the cognitive weights
$prob(Pt_i) = 1/k$ where k is the number of execution paths.
<table>
<tr><th>Type</th><th>Structure</th><th>Weight</th></tr>
<tr><td>Branch</td><td>OR split (two-way)</td><td>2</td></tr>
<tr><td></td><td>OR split (many-ways)</td><td>3</td></tr>
<tr><td></td><td>OR join (two-way)</td><td>2</td></tr>
<tr><td></td><td>OR join (many-ways)</td><td>3</td></tr>
<tr><td>Iteration</td><td>while</td><td>3</td></tr>
<tr><td></td><td>repeatUntil</td><td>3</td></tr>
<tr><td></td><td>forEach</td><td>3</td></tr>
<tr><td>Concurrency</td><td>flow</td><td>4</td></tr>
<tr><td></td><td>join node</td><td>4</td></tr>
<tr><td>Invocation</td><td>service invocation</td><td>2</td></tr>
<tr><td>Interrupt</td><td>exception handler</td><td>3</td></tr>
<tr><td></td><td>event handler</td><td></td></tr>
</table>
</td></tr>
</table>

Debnath et al. [Deb+10] (2010) described six new metrics at the swimming lane level (not at the process level), and reused some of the metrics that were described in Rolón et al. [Rol+05]. A use case where the metrics were used was presented in the paper.

Kreimeyer [Kre10] (2010) described a large set of metrics (52) that dealt with complex processes. Some of the metrics cannot be computed, and some are not defined at the process level. Only metrics that could be computed and were defined at the process level were included in this SLR. He used process metrics (at multiple levels – for example activities) to find interesting outliers in a large and complex process. He used visualization to help identify the outliers. The paper did not validates the metrics. Table B.33 describes the proposed metrics.

Kreimeyer et al. [Kre+10] (2010) presented another use case of how to use the metrics defined by Kreimeyer [Kre10] to find outliners that require further attention.

Table B.33: Kreimeyer's 2010 metrics

Type	Metric	Description
Size and density	`NDos`[7]	Number of domains. Number of different domains within the network (i.e., classes of entities). Sub-graphs in the network, similar to separating the graph into pools in BPMN, where each pool is a domain.
	`NN`[7]	Number of nodes. This has the same meaning as `NA`, which was defined by García et al. [Gar+03], see Table B.5.
	`NEN`[7]	Number of edges per node.
	`NCl`[7]	Number of classes. Number of unique nodes (number of nodes that do not bear the same name, as opposed to total number of nodes)
	`NE`[7]	Number of edges. This has the same meaning as `A`, which was defined by Mendling et al. [Men+06], see Table B.12.
	`RD`[7]	Relational density. Quotient of the number of edges in a domain and the number of possible edges
	`NUN`[7]	Number of unconnected nodes. Number of nodes which are not connected to the graph
Hierarchies	`HH`[7]	Height of hierarchy.
	`WH`[7]	Width of hierarchy. Number of all end nodes (per level) of a tree
Adjacency	`NIS`[7]	Number of independent sets.
Cycles	`NF`[7]	Number of feedbacks.
Paths	`APL`[7]	Average path length. Average path length for all paths across the overall network

Abreu et al. [Abr+10] (2010) defined and applied metrics to two Information Technology (IT) service management processes. They reused metrics from several sources, in particular from Porciúncula [Por10]. Note that Porciúncula [Por10] was rejected for this review, because the document was not published in English. However, Table B.34 describes most of the non-duplicated metrics in Porciúncula [Por10], and was published in the same year.

[7] The author did not provided a symbol or abbreviation for this metric

Table B.34: Abreu et al.'s 2010 metrics

Metric	Description
CFC	Cfc. This has the same meaning as CFC, which was defined by Cardoso [Car05b], see Table B.8.
CNC	Cnc. This has the same meaning as $\texttt{CNC}_P$, which was defined by Latva-Koivisto [Lat01], see Table B.4.
HKM	Henry and kafura metric. $$\texttt{HKM} = \texttt{activities in a pool} \times (\text{start events in the pool} \times \text{end events in the pool})^2$$
NA[8]	Number of activities. This has the same meaning as NA, which was defined by García et al. [Gar+03], see Table B.5.
NAf[8]	Number of artifacts.
NAS[8]	Number of associations.
NC[8]	Number of connectors. This has the same meaning as NDWP, which was defined by García et al. [Gar+04a], see Table B.6.
NCG[8]	Number complex gateways. This has the same meaning as NCD, which was defined by Rolón et al. [Rol+06b], see Table B.9.
ND[8]	Nesting depth. This has the same meaning as MaxND, which was defined by Gruhn and Laue [GL06b], see Table B.11.
NDO[8]	Number of data objects. This has the same meaning as NWP, which was defined by García et al. [Gar+03], see Table B.5.
NDOp[8]	Number of different objects in all possible paths.
NEE[8]	Number of end events. This has the same meaning as TNEE, which was defined by Rolón et al. [Rol+06b], see Table B.9.
NFO[8]	Number of flow objects.
NFOSP[8]	Number of flow objects in smallest path.
NG[8]	Number of groups.
NGa[8]	Number of gates (fan-in + fan-out).
NGDE[8]	Number gateway data based exclusive.
NGDI[8]	Number gateway data based inclusive.
NGEE[8]	Number gateway event based exclusive.
NIE[8]	Number of intermediate events. This has the same meaning as NTIE, which was defined by Rolón et al. [Rol+06b], see Table B.9.
NIG[8]	Number of input gates (fan-in).

Continued on next page

[8]The author did not provided a symbol or abbreviation for this metric

Table B.34 – *Continued from previous page*

Metric	Description
NL[9]	Number of lanes. This has the same meaning as NPR, which was defined by García et al. [Gar+03], see Table B.5.
NMF[9]	Number of message flows. This has the same meaning as NMF, which was defined by Rolón et al. [Rol+06b], see Table B.9.
NOBP[9]	Number of objects in biggest path.
NOG[9]	Number of output gates(fan-out).
NP[9]	Number of pools. This has the same meaning as NP, which was defined by Rolón et al. [Rol+06b], see Table B.9.
NPG[9]	Number parallel gateways. This has the same meaning as NPF, which was defined by Rolón et al. [Rol+06b], see Table B.9.
NPP[9]	Number of possible paths.
NSE[9]	Number of start events. This has the same meaning as NTSE, which was defined by Rolón et al. [Rol+06b], see Table B.9.
NSF[9]	Number of sequence flows. This has the same meaning as NDWP, which was defined by García et al. [Gar+04a], see Table B.6.
NSL[9]	Number of swimlanes. This has the same meaning as NP, which was defined by Rolón et al. [Rol+06b], see Table B.9.
NSP[9]	Number of sub-processes.
NTA[9]	Number of text annotations.
PC[9]	Process complexity. $$\mathtt{PC} = \mathtt{ND} + \mathtt{CFC}$$
TNE	Number of events. This has the same meaning as TNE, which was defined by Rolón et al. [Rol+06b], see Table B.9.
TNG	Number of gateways. This has the same meaning as TNG, which was defined by Rolón et al. [Rol+06b], see Table B.9.
TNT	Number of tasks. This has the same meaning as TNT, which was defined by García et al. [Gar+03], see Table B.5.

Sánchez-González et al. [S+́10b] (2010) used data from a set of six experiments to validate metrics from Mendling [Men07]. Sánchez-González et al. [S+́11] (2011) conducted an experiment to find thresholds for the CFC metric [Car06b]. The paper does not introduce or validate new metrics. Sánchez-GonzáLez et al. [S+́12] (2012) described a similar experiment used to find thresholds for some metrics. Sánchez-González et al. [Sán+15] (2015) described

[9]The author did not provided a symbol or abbreviation for this metric

a case study that evaluated the thresholds that had been identified in their previous work [S+́11; S+́12].

Mäesalu [Mäe11] (2011) used metrics from Cardoso [Car05b], Mendling [Men06; Men07], Rolón et al. [Rol+05], and Vanderfeesten et al. [Van+08b] for a controlled experiment between structured process models and unstructured process models. The aim of his research was to confirm that structured process models are less complex that unstructured process models, but he was unable to validate this hypothesis.

La Rosa et al. [La +11b] (2011) described a set of metrics used to evaluate abstract syntax modification patterns for model complexity. Twelve patterns were presented. The patterns were presented in BPMN, but described in a language independent way and compared to implementations of UML AD, EPC, BPMN, YAWL, and others. In this work, they evaluated the usability of the patterns, but they did not validate the metrics. This paper was a continuation of their paper on concrete syntax patterns to visualize a process model, which did not require the use of metrics [La +11c]. Their metrics are presented in Table B.35.

Table B.35: La Rosa et al.'s 2011 metrics

Metric	Description
`MS`[10]	Model size. This has the same meaning as `F`, which was defined by Mendling et al. [Men+06], see Table B.12. The number of nodes in a process, however based on the description, it seems like subprocesses are not counted.
`Depth`[10]	Depth. [Web+11]. Number of modular levels
`diam`[10]	Diameter. This has the same meaning as `diam`, which was defined by Mendling [Men07], see Table B.16. The number of nodes in the longest path from a start element to an end element in a process model
`AGD`[10]	Average gateway degree. This has the same meaning as $\overline{\mathtt{d}_C}$, which was defined by Mendling [Men07], see Table B.16. The number of nodes a gateway is in average connected to.
`S`[10]	Structuredness. [LM10]. Restructuring ratio of an unstructured model to a block-structured variant
`MO`[10]	Modules overhead. $$\mathtt{MO} - \frac{\mathtt{NM}}{\mathtt{MS}}$$ Where, `NM` is the number of modules in the process (number of sub-processes plus number of lanes – for process notations that support swimming lanes)

Continued on next page

[10]The author did not provided a symbol or abbreviation for this metric

Table B.35 – *Continued from previous page*

Metric	Description
DMC[11]	Different modeling concepts. The number of different model concepts used in a process model.

Antonini et al. [Ant+11] (2011) introduced seven metrics of size, structural complexity, and coupling. They did not consider size and coupling to be complexity metrics, however in this review size is considered a complexity metric. Therefore, this review only included the two structural complexity metrics and four size metrics, as shown in Table B.36. They divided software attributes into internal (size, structural complexity, cohesion, coupling, and length [Mor99; Mor08]), and external (maintainability) attributes. Their approach was based on calculating internal attributes, in particular size, structural complexity, and coupling. Their theoretical validation was based on the work of Morasca [Mor99; Mor08].

Table B.36: Antonini et al.'s 2011 metrics

Metric	Description
$\texttt{Size}_A$	Activity size of process. $$\texttt{Size}_A(p) = \sum_{i=1}^{n_T} \texttt{Size}_A(\texttt{task}_i) + \sum_{i=1}^{n_{ST}} \texttt{Size}_A(\texttt{supertask}_i)$$ $$\texttt{Size}_A(\texttt{task}) = 1$$ $$\texttt{Size}_A(\texttt{supertask}) = \sum_{i=1}^{n_T} \texttt{Size}_A(\texttt{task}_i)$$ Where, p is a process defined in BPMN n_T number of simple tasks inside the process or a super task[12] **task** a simple task **supertask** a compound task (subflow or subprocess) n_{ST} number of super tasks
$\texttt{Complexity}_{CF}$	Control flow complexity. This has the same meaning as CFC, which was defined by Cardoso et al. [Car+06], see Table B.10. $$\texttt{Complexity}_{CF}(p) = \sum_{rt \in \texttt{AND-split}} \texttt{CFC}_{\texttt{AND-split}}(rt) + \sum_{rt \in \texttt{OR-split}} \texttt{CFC}_{\texttt{OR-split}}(rt) + \sum_{rt \in \texttt{XOR-split}} \texttt{CFC}_{\texttt{XOR-split}}(rt)$$

Continued on next page

[11] The author did not provided a symbol or abbreviation for this metric
[12] Note that this definition seems to ignore the fact that subprocesses can have subprocesses inside.

Table B.36 – *Continued from previous page*

Metric	Description
	Where, rt is a set of routing tasks $$\texttt{CFC}_{\texttt{AND-split}} = 1$$ $$\texttt{CFC}_{\texttt{OR-split}} = 2^{\text{fan-out}} - 1$$ $$\texttt{CFC}_{\texttt{XOR-split}} = \text{fan-out}$$
$\texttt{Size}_{CF}$	Size of control flow graph. This has the same meaning as `NOAC`, which was defined by Cardoso et al. [Car+06], see Table B.10. $$\texttt{Size}_{CF}(p) = \text{NOAC}$$
$\texttt{Size}_{DF}$	Size of data flow graph. $$\texttt{Size}_{DF}(p) = \sum_{j=1}^{n_T} V_{i,o}^{j}$$ Where, n_T is the number of activities in the process $V_{i,o}^{j}$ is the number of input data received by node j plus the number of output data produced by the same node
$\texttt{Complexity}_{DF}$	Data flow complexity. $$\texttt{Complexity}_{DF}(p) = \sum_{j=1}^{n_{RT}} \texttt{Complexity}_{DF}(rt_j) + \sum_{j=1}^{n_T} \texttt{Complexity}_{DF}(t_j)$$ Where, n_{RT} is the number of routing tasks n_T is the number of non-routing tasks $$\texttt{Complexity}_{DF}(\texttt{routing task}) = 1$$ $$\texttt{Complexity}_{DF}(\texttt{non-routing task}) = V_{i,o}$$
$\texttt{Size}_{R}$	Resource size. $$\texttt{Size}_{R}(p) = \texttt{R}$$ Where, `R` is the number of resources available for executing process `p` (normally the number of lanes in the BPMN diagram)

Kluza and Nalepa [KN12] (2012) provided a short survey of existing process model complexity metrics and introduced two new complexity metrics for BPMN. The metrics are described in Table B.37.

Kluza et al. [Klu+14] (2014) presented an attempt to validate the two metrics in Kluza and Nalepa [KN12], comparing them to other metrics.

Table B.37: Kluza and Nalepa's 2012 metrics

Metric	Description
DSM	Durfee square metric. based on [Hir05], and is described by the authors as DMS "equals d if there are d types of elements which occurs at least d times (each)" [KN12]
PSM	Perfect square metric. based on Egghe [Egg06], and it is described by the authors as "given a set of element types ranked in decreasing order of the number of their instances, the PSM is the (unique) largest number such that the top p types occurs (together) at least p^2 times." [KN12]

Solichah et al. [Sol+13a] (2013) tried to validate three metrics proposed by Antonini et al. [Ant+11] and Cardoso et al. [Car+06]. They used four processes from two open source systems (modeled in BPMN) to calculate MCC, CFC, D and used metrics from the source code implementing these four processes (number of form fields, number of source files, and McCabe's cyclomatic number of the source code). They assume that higher numbers were more complex, and concluded that D was a good metric for measuring complexity.

Setiawan and Sadiq [SS13] (2013) proposed a framework that evaluates the complexity of implemented business process models using three perspectives: structural, variance and performance. The structural perspective is the control-flow structural complexity measured using CFC [Car05b]. The variance perspective corresponds to subsets of the process model extracted from the execution process logs. The performance perspective corresponds with Shannon's entropy [Sha48] calculated using information from the execution process log. The three perspectives can be presented in a 3D plot. For this review, only the CFC metric was taken into account because the other two metrics require runtime information.

Sun and Hou [SH14] (2014) described an information flow complexity metric for YAWLprocesses. They claimed that the metric could be described for BPEL, UML AD, or EPML. The metric was based on the idea of fan-in/fan-out [GL06a]. A theoretical validation of the metric was presented. The metric was calculated for two processes and compared against three other metrics (NOA, CFC, and ND). The metric is presented in Table B.38.

Table B.38: Sun and Hou's 2014 metrics

Metric	Description
`IF`	Information flow complexity. This has the same meaning as `IF4BP`, which was defined by Muketha et al. [Muk+10b], see Table B.30. $$\mathtt{IF} = \sum_{i}^{n} \mathtt{IF}_i$$ Where, $\mathtt{IF}_i$ is the complexity of each task, calculated as follows $$\mathtt{IF}_i = (\mathtt{NOIT} \times \mathtt{NOOT})^2$$ `NOIT` is the number of incoming arcs (fan in) `NOOT` is the number of outgoing arcs (fan out)

Çoşkun [Çoş14] defined a cognitive activity depth arc control flow (`CADAC`) metric for a subset of BPMN. The metric is constrained to BPMN 2.0 models with only exclusive, parallel and inclusive gateways [Çoş14]. The metric was theoretically validated, and an attempt was made to empirically validate the metric by comparing it with other known metrics. However, there was no clear indication of which metric really measured the complexity of the 12 models. The evaluation was done subjectively without any statistical analysis. Table B.39 describes the `CADAC` metric.

Table B.39: Çoşkun's 2014 metrics

Metric	Description
`CADAC`	Cognitive activity depth arc control flow. $$\begin{aligned}\mathtt{CADAC} = \mathtt{NOA} &+ (\mathtt{Maximun\ Nesting\ Depth} \times 14) \\ &+ (\mathtt{Number\ of\ XORs} \times 2) + (\mathtt{Number\ of\ ANDs} \times 4) \\ &+ (\mathtt{Numbers\ of\ ORs} \times 7) + (\mathtt{Number\ of\ Arcs} \times 1) \\ &+ \mathtt{Maximun}\ (\text{fan-in} \times \mathtt{fan\text{-}out})^2 \times 4)\end{aligned}$$

Martinho et al. [Mar+15d] (2015) applied four metrics from Cardoso [Car08] to three processes that were implemented in a health care organization. The paper compared the original three processes to the three new versions of the processes, and concluded that they had practically remained the same with respect to complexity metrics.

Lübke [LÏ5] (2015) implemented BPEL metrics in order to gain a better understanding of how the processes develop over time. Their tool provided a time-line of how process size metrics evolve over time (versions of a process). Their metrics are described in Table B.40.

Table B.40: Lübke's 2015 metrics

Metric	Description
ATC[13]	Activity type count. Number of activities by type (Receive, Invoke, exit, etc.).
TBAC[13]	Total basic activity count. This has the same meaning as NT, which was defined by Rolón et al. [Rol+06b], see Table B.9. Number of all basic activities
TSAC[13]	Total structured activity count. This has the same meaning as SA, which was defined by Held and Blochinger [HB09], see Table B.28. Number of all structured activities
EAC[13]	Extension activity count. Number of all vendor extension activities
BAD[13]	Basic activity distribution. $$\mathtt{BAD} = \frac{\mathtt{TBAC}}{\mathtt{TBAC} + \mathtt{TSAC} + \mathtt{EAC}} \times 100$$
EDB[13]	Extension activity distribution. $$\mathtt{EAD} = \frac{\mathtt{EAC}}{\mathtt{TBAC} + \mathtt{TSAC} + \mathtt{EAC}} \times 100$$

Kluza [Klu15] (2015) proposed a complexity metric for processes that use business rules. The paper provides a good survey of complexity metrics for both process models and rules. The metric is described in Table B.41.

Table B.41: Kluza's 2015 metrics

Metric	Description
Complexity	Complexity of process integrated with rules. $$\mathtt{Complexity} = \frac{\mathtt{NOR}}{\mathtt{NOD}} \times \frac{\mathtt{NoA}}{\mathtt{NOAC} + \mathtt{NOF}} \times \mathtt{Concurrency}$$ Where, NOR is the number of rules [Sue+90] NOD is the number of decision components [Sue+90] NoA is the number of activities in a process [Car+06] NOAC is the number of activities and control-flow elements in a process [Car+06] NOF is the number of control-flow connections [Lat01] Concurrency is the maximum number of paths in a process that may be concurrently active due to splits [S+10b]

[13]The author did not provided a symbol or abbreviation for this metric

Anugrah et al. [Anu+15] (2015) used process model complexity metrics from Cardoso [Car08] and Mao [Mao10a] in the context of process mining. The paper described an approach to decompose a business process into several variants [Van+08c]. It then uses control-flow complexity metrics to select the best variant. It also provided an example, but this paper did not describe new metrics and did not validate any of the metrics.

B.2 Identified Metrics

Table B.42 describes all of the metrics identified during the SLR.

Table B.42: Complexity summary of BPM metrics identified by the literature review

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	Empirical	Primary	Secondary
CA = activity coupling [Gar+03] Table B.5	Ratio	SPEM			[Gar+04b], [Gar+04c], [Gar+04a], [Gar+04d]	[Gar+03]	[Gar+04b], [Gar+04c], [Gar+04a], [Gar+04d]
$\mathtt{Size}_A$ = activity size of process [Ant+11] Table B.36	Counter	BPMN		[Ant+11]		[Ant+11]	
ATC = activity type count [LÏ5] Table B.40	Counter	BPEL				[LÏ5]	
δ_S = aggregated depth fraction [Lv09] Table B.25	Calculated	Workflow-Nets			[Lv09]	[Lv09]	
AIR = assign/invoke ratio [HB09] Table B.28	Ratio	BPEL				[HB09]	
$\bar{\mathtt{C}}_A$ = average activity complexity [Tja99] Table B.3	Average	Graph				[Tja99]	
ACCSA = average cognitive complexity of structured activity [Muk+10b] Table B.30	Ratio	BPEL		[Muk+10b]	[Muk+10b]	[Muk+10b]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	**Empirical**	**Primary**	**Secondary**
`ACC` = average connector cohesion [Dan+96] Table B.1	Average	EPC			[Dan+96]	[Dan+96]	
$\overline{\texttt{d}_C}$ = average degree of connectors [Men07] Table B.16	Average	EPC	`AGD` [La +11b] Table B.35		[S+́10b], [Mäe11], [Men+07c], [Men07]	[Men07]	[S+́10b], [Mäe11], [Men+07c]
`ADP` = average degree of place [Mao10a] Table B.31	Average	Petri-Net			[Mao10b]	[Mao10a]	[Mao10b]
`ADT` = average degree of transition [Mao10a] Table B.31	Average	Petri-Net			[Mao10b]	[Mao10a]	[Mao10b]
`AEC` = average event cohesion [Dan+96] Table B.1	Average	EPC			[Dan+96]	[Dan+96]	
`AEPC` = average execution path complexity [Mao10a] Table B.31	Average	Petri-Net			[Mao10b]	[Mao10a]	[Mao10b]
AEPC_{CI} = average execution path complexity based on cognitive informatics [Mao10b] Table B.32	Weighted	Petri-Net			[Mao10b]	[Mao10b]	
`AFC` = average function cohesion [Dan+96] Table B.1	Average	EPC			[Dan+96]	[Dan+96]	
`ALSA` = average length of structured activity [Muk+10b] Table B.30	Ratio	BPEL		[Muk+10b]	[Muk+10b]	[Muk+10b]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	**Empirical**	**Primary**	**Secondary**
APL = average path length [Kre10] Table B.33	Average	Graph				[Kre10]	
BAD = basic activity distribution [Lï5] Table B.40	Percentage	BPEL				[Lï5]	
CNC_K = coefficient of network complexity (Kaimann) [Lat01] Table B.4	Calculated	Graph			[Lat01]	[Lat01]	
CNC_P = coefficient of network complexity (Pascoe) [Lat01] Table B.4	Ratio	Graph	CNC [Car+06] Table B.10, CNC [Abr+10] Table B.34,CNC [Men07] Table B.16		[Sol+13a], [Lat01], [Men+07c], [Ś+10b], [Men07]	[Lat01]	[Sol+13a], [Men+07c], [Ś+10b], [Men07]
CADAC = cognitive activity depth arc control flow [Çoş14] Table B.39	Weighted	BPMN		[Çoş14]	[Çoş14]	[Çoş14]	
CCBP = cognitive complexity [Muk+10b] Table B.30	Weighted	BPEL		[Muk+10b]	[Muk+10b]	[Muk+10b]	
CW = cognitive weight [GL06b] Table B.11	Weighted	YAWL	CC_{YAWL} [GL06a] Table B.14	[HZ09]		[GL06b]	[HZ09],
CI = complexity index [Lat01] Table B.4	Algorithm	Graph			[Lat01]	[Lat01]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	**Type**	**Notation**	**Same metric**	**Validation**		**Sources**	
				Theoretical	**Empirical**	**Primary**	**Secondary**
`Complexity` = complexity of process integrated with rules [Klu15] Table B.41	Calculated	BPMN			[Klu15]	[Klu15]	
`CLA` = connectivity level between activities [Rol+06b] Table B.9	Ratio	BPMN		[Rol09]	[Rol+09b], [Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+09b], [Rol+08], [Rol+07a]
`CLP` = connectivity level between pools [Rol+06b] Table B.9	Ratio	BPMN		[Rol09]	[Rol+09b], [Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+09b], [Rol+08], [Rol+07a]
`CH` = connector heterogeneity [Men07] Table B.16	Calculated	EPC			[MS08], [S+́10b], [Men+07c], [Men07]	[Men07]	[MS08], [S+́10b], [Men+07c]
`MM` = connector mismatch [Men07] Table B.16	Calculated	EPC			[S+́10b], [Men+07c], [Men07]	[Men07]	[S+́10b], [Men+07c]

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	**Empirical**	**Primary**	**Secondary**
`CFC` = control-flow complexity [Car05b] Table B.8	Calculated	Graph	$\texttt{CFC}_{abs}$ [Car08] Table B.20, `CFC` [HB09] Table B.28, `CFC` [Abr+10] Table B.34, $\texttt{Complexity}_{CF}$ [Ant+11] Table B.36, `SC` [Men+06] Table B.12	[Car08], [Car05b], [HB09], [Ant+11]	[Rol+09a], [Sol+13a], [Car06b], [Mäe11], [Men+06]	[Car05b]	[Car08], [HB09], [Ant+11], [Rol+09a], [Sol+13a], [Car06b], [Mäe11], [Men+06]
$\texttt{CFC}^{\texttt{BPEL}}_{\texttt{Process}}$ = control-flow complexity for BPEL process [Car07b] Table B.17	Calculated	BPEL				[Car07b]	
`CAR` = copy/assign ratio [HB09] Table B.28	Ratio	BPEL				[HB09]	
`CP` = coupling [Van+07a] Table B.19	Calculated	Graph				[Van+07a]	
`CC` = cross-connectivity [Van+08b] Table B.21	Calculated	EPC			[Van+08b], [Mäe11]	[Van+08b]	[Mäe11]
`Cycle` = cycle [Men+06] Table B.12	Counter	YAWL			[Men+06]	[Men+06]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	**Empirical**	**Primary**	**Secondary**
CYC = cyclicity [Men07] Table B.16	Ratio	EPC			[Men+07c], [MS08], [S+́10b], [MN07], [Men07]	[Men07]	[Men+07c], [MS08], [S+́10b], [MN07]
S = cyclomatic number [Lat01] Table B.4	Calculated	Graph	MCC [Car+06] Table B.10, MCC [HB09] Table B.28, CC [Mao10a] Table B.31	[HB09]	[Lat01], [Mao10b]	[Lat01]	[HB09], [Mao10b]
$\texttt{Complexity}_{DF}$ = data flow complexity [Ant+11] Table B.36	Calculated	BPMN		[Ant+11]		[Ant+11]	
DFI = data flow intensity [HB09] Table B.28	Calculated	BPEL		[HB09]		[HB09]	
DOP = degree of parallelism [HB09] Table B.28	Calculated	BPEL		[HB09]		[HB09]	
$\texttt{I}_{f^V}$ = Dehmer's graph entropy [Bor+09a] Table B.27	Calculated	Graph			[Bor+09a]	[Bor+09a]	
$\texttt{d}_e$ = density of an EPC [Men06] Table B.13	Calculated	EPC			[Men06]	[Men06]	
$\texttt{d}_w$ = density of a workflow-net [Men06] Table B.13	Calculated	Workflow-Nets				[Men06]	
$\texttt{d}_y$ = density of a YWAL model [Men06] Table B.13	Calculated	YAWL				[Men06]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	Empirical	Primary	Secondary
Δ = density of the process graph [Men07] Table B.16	Calculated	EPC			[Ś+10b], [Mäe11], [Men+07c], [Men07]	[Men07]	[Ś+10b], [Mäe11], [Men+07c]
`Depth` = depth [La +11b] Table B.35	Counter	BPMN				[La +11b]	
`diam` = diameter [Men07] Table B.16	Counter	EPC	`diam` [La +11b] Table B.35		[Ś+10b], [MS08], [Men+07c], [Men07]	[Men07]	[Ś+10b], [MS08], [Men+07c]
`DMC` = different modeling concepts [La +11b] Table B.35	Counter	BPMN				[La +11b]	
`DSM` = Durfee square metric [KN12] Table B.37	Calculated	BPMN			[Klu+14]	[KN12]	[Klu+14]
`ECaM` = extended Cardoso [Lv09] Table B.25	Calculated	Workflow-Nets			[Lv09]	[Lv09]	
`ECyM` = extended cyclomatic [Lv09] Table B.25	Calculated	Workflow-Nets	`C` [Bor+09a] Table B.27		[Lv09], [Bor+09a]	[Lv09]	[Bor+09a]
`EM` = extended measure [Sob99] Table B.2	Average	VPML				[Sob99]	
`EAC` = extension activity count [LÏ5] Table B.40	Percentage	BPEL				[LÏ5]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoreti-cal	Empirical	Primary	Secondary
`EDB` = extension activity distribution [Lï5] Table B.40	Counter	BPEL				[Lï5]	
$\mathtt{F}_i\mathtt{F}_o$ = fan-in / fan-out [GL06b] Table B.11	Calculated	Graph				[GL06b]	
`Flexibility` = flexibility [Tja99] Table B.3	Percentage	Graph				[Tja99]	
`GREM` = global ripple effect measure [Sob99] Table B.2	Ratio	VPML				[Sob99]	
`HH` = height of hierarchy [Kre10] Table B.33	Counter	Graph				[Kre10]	
`HKM` = Henry and Kafura metric [Abr+10] Table B.34	Calculated	BPMN				[Abr+10]	
`IF4BP` = information flow complexity [Muk+10b] Table B.30	Calculated	BPEL	`IF` [SH14] Table B.38	[Muk+10b], [SH14]	[Muk+10b], [SH14]	[Muk+10b]	[SH14]
`Integration` = integration [Tja99] Table B.3	Percentage	Graph				[Tja99]	
`JC` = join complexity [Men+06] Table B.12	Counter	YAWL			[Men+06]	[Men+06]	
`JSR` = join-split-ratio [Men+06] Table B.12	Ratio	YAWL			[Men+06]	[Men+06]	
$\mathtt{LBC}_T$ = log-based complexity [Car07a] Table B.18	Algorithm	BPMN				[Car07a]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	Empirical	Primary	Secondary
Λ = maximum depth of all nodes [Men07] Table B.16	Algorithm	EPC			[S+́10b], [Men07], [Men+07c]	[Men07]	[S+́10b], [Men+07c]
`MaxND` = maximum nesting depth [GL06b] Table B.11	Counter	EPC	`ND` [Abr+10] Table B.34			[GL06b]	
$\widehat{\mathtt{d}_C}$ = maximun degree of a connector [Men07] Table B.16	Calculated	EPC			[S+́10b], [Men+07c], [Men07]	[Men07]	[S+́10b], [Men+07c]
`MeanND` = mean nesting depth [GL06b] Table B.11	Average	EPC				[GL06b]	
`MM` = modularization measure [Sob99] Table B.2	Ratio	VPML				[Sob99]	
`MO` = modules overhead [La +11b] Table B.35	Ratio	BPMN				[La +11b]	
`NGDE` = number gateway data based exclusive [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	
`NGDI` = number gateway data based inclusive [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	
`NGEE` = number gateway event based exclusive [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	**Empirical**	**Primary**	**Secondary**
NA = number of activities [Gar+03] Table B.5	Counter	Graph	AC [Car05b] Table B.8,NA [Abr+10] Table B.34, NOA [Car+06] Table B.10, NA [GL06b] Table B.11, NOA [HB09] Table B.28, TNA [Rol+06b] Table B.9, S_N [Men07] Table B.16, NN [Kre10] Table B.33, TNT [Abr+10] Table B.34, TNT [Rol+06b] Table B.9	[HB09], [Rol09]	[Gar+04d], [Gar+04c], [Gar+04b], [Gar+04a], [Rol+08], [Rol+09b], [Rol+07a], [MS08], [Mäe11], [Ś+10b], [Men+07c], [MN07], [Men07]	[Gar+03]	[HB09], [Rol09], [Gar+04d], [Gar+04c], [Gar+04b], [Gar+04a], [Rol+08], [Rol+09b], [Rol+07a], [MS08], [Mäe11], [Ś+10b], [Men+07c], [MN07], [Men07]
NOAC = number of activities and control flow elements [Car+06] Table B.10	Counter	Graph	NOAC [HB09] Table B.28, $\texttt{Size}_{CF}$ [Ant+11] Table B.36	[HB09], [Ant+11]		[Car+06]	[HB09], , [Ant+11]
NOACC = number of activities, control structures, and copy [HB09] Table B.28	Counter	BPEL		[HB09]		[HB09]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	**Empirical**	**Primary**	**Secondary**
NOAJS = number of activities joins and splits [Car+06] Table B.10	Counter	Graph				[Car+06]	
$S_{C_{and}}$ = number of AND connectors [Men07] Table B.16	Counter	EPC			[Men07], [Men+07c]	[Men07]	[Men+07c]
AND_j = number of AND joins [Men+06] Table B.12	Counter	YAWL	$S_{j_{and}}$ [Men07] Table B.16		[Men+06], [Men07], [Men+07c]	[Men+06]	[Men07], [Men+07c]
AND_s = number of AND splits [Men+06] Table B.12	Counter	YAWL	$S_{S_{and}}$ [Men07] Table B.16		[Men+06], [Men+07c], [Men07]	[Men+06]	[Men+07c], [Men07]
A = number of arcs [Men+06] Table B.12	Counter	YAWL	S_A [Men07] Table B.16, NE [Kre10] Table B.33		[Men+06], [Men07], [Men+07c]	[Men+06]	[Men07], [Men+07c]
NAf = number of artifacts [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	
NAS = number of associations [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	
NCl = number of classes [Kre10] Table B.33	Counter	Graph				[Kre10]	
NCSA = number of collapsed ad-hoc sub-process [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	**Empirical**	**Primary**	**Secondary**
NCSC = number of collapsed compensation sub-process [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+07a], [Rol+08]	[Rol+06b]	[Rol09], [Rol+07a], [Rol+08]
NCSL = number of collapsed looping sub-process [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
NCSMI = number of collapsed multiple instance sub-process [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
NCS = number of collapsed sub-process [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
TNCS = total number of collapsed sub-processes [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a], [Rol+09b]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a], [Rol+09b]
NTC = number of compensation tasks [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
NCD = number of complex decision/merge [Rol+06b] Table B.9	Counter	BPMN	NCG [Abr+10] Table B.34	[Rol09]	[Rol+07a], [Rol+09b], [Rol+08]	[Rol+06b]	[Rol09], [Rol+07a], [Rol+09b], [Rol+08]

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	**Empirical**	**Primary**	**Secondary**
`NDWP` = number of dependences [Gar+04a] Table B.6	Counter	EPC	S_C [Men07] Table B.16, `NC` [Abr+10] Table B.34,`NSF` [Rol+06b] Table B.9, `NSF` [Abr+10] Table B.34	[Rol09]	[Men07], [Men+07c], [Gar+04d], [Gar+04a], [Gar+04b], [Gar+04c], [Rol+09b], [Rol+08], [Rol+07a]	[Gar+04a]	[Rol09], [Men07], [Men+07c], [Gar+04d], [Gar+04b], [Gar+04c], [Rol+09b], [Rol+08], [Rol+07a]
`NDOp` = number of different objects in all possible paths [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	
`NDos` = number of domains [Kre10] Table B.33	Counter	Graph				[Kre10]	
`NEN` = number of edges per node [Kre10] Table B.33	Counter	Graph				[Kre10]	
`NECaE` = number of end cancel event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
`NECoE` = number of end compensation event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]

Continued on next page

Table B.42 – *Continued from previous page*

Metric	**Type**	**Notation**	**Same metric**	**Validation**		**Sources**	
				Theoretical	**Empirical**	**Primary**	**Secondary**
`NEEE` = number of end error event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
`TNEE` = total number of end events [Rol+06b] Table B.9	Counter	BPMN	`NEE` [Abr+10] Table B.34, $\texttt{E}_{end}$ [Men+06] Table B.12, $\texttt{S}_{E_E}$ [Men07] Table B.16	[Rol09]	[Rol+09b], [Rol+08], [Rol+07a], [Men+06], [Men+07c], [Men07]	[Rol+06b]	[Rol09], , [Rol+09b], [Rol+08], [Rol+07a], [Men+06], [Men+07c], [Men07]
`NELE` = number of end link event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
`NEMsE` = number of end message event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+07a], [Rol+08]	[Rol+06b]	[Rol09], [Rol+07a], [Rol+08]
`NEMuE` = number of end multiple event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
`NENE` = number of end none event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	Empirical	Primary	Secondary
NETE = number of end terminate event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
TNE = total number of events [Rol+06b] Table B.9	Counter	BPMN	TNE [Abr+10] Table B.34, S_E [Men07] Table B.16	[Rol09]	[Rol+07a], [Rol+08], [Rol+09b], [Men+07c], [Men07]	[Rol+06b]	[Rol09], , [Rol+07a], [Rol+08], [Rol+09b], [Men+07c], [Men07]
NEDDB = number of exclusive decision/merge data-based [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+09b], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+09b], [Rol+07a]
NEDEB = number of exclusive decision/merge event-based [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+09b], [Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+09b], [Rol+08], [Rol+07a]
NF = number of feedbacks [Kre10] Table B.33	Counter	Graph				[Kre10]	
NFO = number of flow objects [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	Empirical	Primary	Secondary
NFOSP = number of flow objects in smallest path [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	
F = number of functions [Men+06] Table B.12	Counter	BPMN	MS [La +11b] Table B.35, S_F [Men07] Table B.16		[Men+06], [Men+07c], [Men07]	[Men+06]	[Men+07c], [Men07]
NGa = number of gates (fan-in + fan-out) [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	
TNG = total number of gateways [Rol+06b] Table B.9	Counter	BPEL	DC [HB09] Table B.28, TNG [Abr+10] Table B.34, NOSA [Muk+10b] Table B.30	[Rol09], [Muk+10b]	[Rol+09b], [Rol+08], [Rol+07a], [Muk+10b]	[Rol+06b]	[Rol09], [Muk+10b], [Rol+09b], [Rol+08], [Rol+07a]
NG = number of groups [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	
NH = number of handles [GL06b] Table B.11	Counter	Workflow-Nets				[GL06b]	
NID = number of incusive decision/merge [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+09b], [Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+09b], [Rol+08], [Rol+07a]
NIS = number of independent sets [Kre10] Table B.33	Counter	Graph				[Kre10]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	Empirical	Primary	Secondary
NDOIn = number of input data objects [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+09b], [Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+09b], [Rol+08], [Rol+07a]
NDWPIn = number of input dependences [Gar+04a] Table B.6	Counter	SPEM			[Gar+04a], [Gar+04b], [Gar+04d], [Gar+04c]	[Gar+04a]	[Gar+04b], [Gar+04d], [Gar+04c]
NIG = number of input gates (fan-in) [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	
NICaE = number of intermediate cancel event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
NICoE = number of intermediate compensation event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
NIEE = number of intermediate error event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+07a], [Rol+08]	[Rol+06b]	[Rol09], [Rol+07a], [Rol+08]
NTIE = total number of intermediate events [Rol+06b] Table B.9	Counter	BPMN	NIE [Abr+10] Table B.34	[Rol09]	[Rol+09b], [Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], , [Rol+09b], [Rol+08], [Rol+07a]

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	**Empirical**	**Primary**	**Secondary**
`NILE` = number of intermediate link event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
`NIMsE` = number of intermediate message event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
`NIMuE` = number of intermediate multiple event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
`NINE` = number of intermediate none event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
`NIRE` = number of intermediate rule event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+07a], [Rol+08]	[Rol+06b]	[Rol09], [Rol+07a], [Rol+08]
`NITE` = number of intermediate timer event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
E_{int} = number of internal events [Men+06] Table B.12	Counter	YAWL			[Men+06]	[Men+06]	
`NTL` = number of looping tasks [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	Empirical	Primary	Secondary
NMF = number of message flows [Rol+06b] Table B.9	Counter	BPMN	NMF [Abr+10] Table B.34	[Rol09]	[Rol+09b], [Rol+07a], [Rol+08]	[Rol+06b]	[Rol09], , [Rol+09b], [Rol+07a], [Rol+08]
NTMI = number of multiple instances tasks [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
NOBP = number of objects in biggest path [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	
$\mathtt{S}_{C_{or}}$ = number of OR connectors [Men07] Table B.16	Counter	EPC			[Men07], [Men+07c]	[Men07]	[Men+07c]
$\mathtt{OR}_j$ = number of OR joins [Men+06] Table B.12	Counter	YAWL	$\mathtt{S}_{j_{or}}$ [Men07] Table B.16		[Men+06], [Men07], [Men+07c]	[Men+06]	[Men07], [Men+07c]
$\mathtt{OR}_s$ = number of OR splits [Men+06] Table B.12	Counter	YAWL	$\mathtt{S}_{S_{or}}$ [Men07] Table B.16		[Men+06], [Men+07c], [Men07]	[Men+06]	[Men+07c], [Men07]
NDOOut = number of output data objects [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+09b], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+09b], [Rol+07a]

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	Empirical	Primary	Secondary
NDWPOut = number of output dependences [Gar+04a] Table B.6	Counter	SPEM			[Gar+04d], [Gar+04c], [Gar+04a], [Gar+04b]	[Gar+04a]	[Gar+04d], [Gar+04c], [Gar+04b]
NOG = number of output gates(fan-out) [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	
NPF = number of parallel fork/join [Rol+06b] Table B.9	Counter	BPMN	NPG [Abr+10] Table B.34	[Rol09]	[Rol+09b], [Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], , [Rol+09b], [Rol+08], [Rol+07a]
N_P = number of places [Mao10a] Table B.31	Counter	Petri-Net			[Mao10b]	[Mao10a]	[Mao10b]
NP = number of pools [Rol+06b] Table B.9	Counter	BPMN	NP [Abr+10] Table B.34, NSL [Abr+10] Table B.34	[Rol09]	[Rol+08], [Rol+09b], [Rol+07a]	[Rol+06b]	[Rol09], , [Rol+08], [Rol+09b], [Rol+07a]
NPP = number of possible paths [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	
NDRA = number of precedence dependences between activities [Gar+03] Table B.5	Counter	SPEM			[Gar+04d], [Gar+04c], [Gar+04b], [Gar+04a]	[Gar+03]	[Gar+04d], [Gar+04c], [Gar+04b], [Gar+04a]

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	**Empirical**	**Primary**	**Secondary**
NPR = number of roles [Gar+03] Table B.5	Counter	BPMN	NL [Rol+06b] Table B.9, NL [Abr+10] Table B.34	[Rol09]	[Rol+08], [Rol+09b], [Rol+07a], [Gar+04c], [Gar+04a], [Gar+04d], [Gar+04b]	[Gar+03]	[Rol09], , [Rol+08], [Rol+09b], [Rol+07a], [Gar+04c], [Gar+04a], [Gar+04d], [Gar+04b]
NSFG = number of sequence flows from gateways [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+09b], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+09b], [Rol+07a]
NSFA = number of sequence flows between activities [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+07a], [Rol+08], [Rol+09b]	[Rol+06b]	[Rol09], [Rol+07a], [Rol+08], [Rol+09b]
NSFE = number of sequence flows from events [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+09b], [Rol+07a], [Rol+08]	[Rol+06b]	[Rol09], [Rol+09b], [Rol+07a], [Rol+08]
N_S = number of services [Mao10a] Table B.31	Counter	Petri-Net			[Mao10b]	[Mao10a]	[Mao10b]

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	**Empirical**	**Primary**	**Secondary**
NTSE = total number of start events [Rol+06b] Table B.9	Counter	BPMN	NSE [Abr+10] Table B.34, $\mathtt{E}_{start}$ [Men+06] Table B.12, $\mathtt{S}_{E_S}$ [Men07] Table B.16	[Rol09]	[Rol+08], [Rol+07a], [Rol+09b], [Men+06], [Men07], [Men+07c]	[Rol+06b]	[Rol09], , [Rol+08], [Rol+07a], [Rol+09b], [Men+06], [Men07], [Men+07c]
NSLE = number of start link event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
NSMsE = number of start message event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
NSMuE = number of start multiple event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
NSNE = number of start none event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
NSRE = number of start rule event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	**Empirical**	**Primary**	**Secondary**
NSTE = number of start timer event [Rol+06b] Table B.9	Counter	BPMN		[Rol09]	[Rol+08], [Rol+07a]	[Rol+06b]	[Rol09], [Rol+08], [Rol+07a]
NSTP = number of steps (tasks) [Gar+03] Table B.5	Counter	SPEM				[Gar+03]	
NSP = number of sub-processes [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	
NT = number of simple tasks [Rol+06b] Table B.9	Counter	BPEL	NOBA [Muk+10b] Table B.30, TBAC [LÏ5] Table B.40	[Muk+10b], [Rol09]	[Muk+10b], [Rol+08], [Rol+07a]	[Rol+06b]	[Muk+10b], , [Rol09], [Rol+08], [Rol+07a]
NTA = number of text annotations [Abr+10] Table B.34	Counter	BPMN				[Abr+10]	
N_T = number of transitions [Mao10a] Table B.31	Counter	Petri-Net			[Mao10b]	[Mao10a]	[Mao10b]
T = number of trees in a graph [Lat01] Table B.4	Counter	Graph			[Lat01]	[Lat01]	
NUN = number of unconnected nodes [Kre10] Table B.33	Counter	Graph				[Kre10]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation: Theoretical	Validation: Empirical	Sources: Primary	Sources: Secondary
NWP = number of work products [Gar+03] Table B.5	Counter	BPMN	TNDO [Rol+06b] Table B.9, NDO [Abr+10] Table B.34	[Rol09]	[Rol+09b], [Rol+08], [Rol+07a], [Gar+04a], [Gar+04d], [Gar+04c], [Gar+04b]	[Gar+03]	[Rol09], , [Rol+09b], [Rol+08], [Rol+07a], [Gar+04a], [Gar+04d], [Gar+04c], [Gar+04b]
$S_{C_{xor}}$ = number of XOR connectors [Men07] Table B.16	Counter	EPC			[Men+07c], [Men07]	[Men07]	[Men+07c]
XOR_j = number of XOR joins [Men+06] Table B.12	Counter	YAWL	$S_{j_{xor}}$ [Men07] Table B.16		[Men+06], [Men+07c], [Men07]	[Men+06]	[Men+07c], [Men07]
XOR_s = number of XOR splits [Men+06] Table B.12	Counter	YAWL	$S_{S_{xor}}$ [Men07] Table B.16		[Men+06], [Men07], [Men+07c]	[Men+06]	[Men07], [Men+07c]
PSM = perfect square metric [KN12] Table B.37	Calculated	BPMN			[Klu+14]	[KN12]	[Klu+14]
c = process cohesion [RV04] Table B.7	Calculated	Graph	ch [Van+08a] Table B.24		[RV04], [Van+08a]	[RV04]	[Van+08a]
PC = process complexity [Abr+10] Table B.34	Calculated	BPMN				[Abr+10]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	**Type**	**Notation**	**Same metric**	**Validation**		**Sources**	
				Theoretical	**Empirical**	**Primary**	**Secondary**
$\mathtt{P}_{CIM}$ = process context independency metric [Kho+09] Table B.26	Calculated	BPEL			[Kho+09]	[Kho+09]	
`k` = process coupling [RV04] Table B.7	Calculated	Graph	`cp` [Van+08a] Table B.24		[RV04], [Van+08a]	[RV04]	[Van+08a]
ρ = process coupling/cohesion ratio [RV04] Table B.7	Ratio	Graph	ρ [Van+08a] Table B.24		[RV04], [Van+08a]	[RV04]	[Van+08a]
$\mathtt{P}_{CM}$ = process coupling metric [Kho+09] Table B.26	Calculated	BPEL				[Kho+09]	
`D` = process difficulty [Car+06] Table B.10	Calculated	Graph			[Sol+13a]	[Car+06]	[Sol+13a]
`N` = process length [Car+06] Table B.10	Calculated	Graph				[Car+06]	
`V` = process volume [Car+06] Table B.10	Calculated	Graph				[Car+06]	
`R` = Randi'c's connectivity index [Bor+09a] Table B.27	Calculated	Graph			[Bor+09a]	[Bor+09a]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	Empirical	Primary	Secondary
RDWPIn = ratio between input dependencies and number of dependencies [Gar+04a] Table B.6	Ratio	BPMN	PDOPIn [Rol+06b] Table B.9	[Rol09]	[Rol+09b], [Rol+08], [Rol+07a], [Gar+04c], [Gar+04d], [Gar+04a], [Gar+04b]	[Gar+04a]	[Rol09], , [Rol+09b], [Rol+08], [Rol+07a], [Gar+04c], [Gar+04d], [Gar+04b]
RDWPOut = ratio between output dependencies and the number of dependencies [Gar+04a] Table B.6	Ratio	BPMN	PDOPOut [Rol+06b] Table B.9	[Rol09]	[Rol+09b], [Rol+08], [Rol+07a], [Gar+04c], [Gar+04d], [Gar+04a], [Gar+04b]	[Gar+04a]	[Rol09], , [Rol+09b], [Rol+08], [Rol+07a], [Gar+04c], [Gar+04d], [Gar+04b]
RSTPA = ratio of steps and activities [Gar+03] Table B.5	Ratio	SPEM				[Gar+03]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	**Empirical**	**Primary**	**Secondary**
RPRA = ratio of process roles and activities [Gar+03] Table B.5	Ratio	BPMN	PLT [Rol+06b] Table B.9	[Rol09]	[Rol+09b], [Rol+08], [Rol+07a], [Gar+04c], [Gar+04d], [Gar+04a], [Gar+04b]	[Gar+03]	[Rol09], , [Rol+09b], [Rol+08], [Rol+07a], [Gar+04c], [Gar+04d], [Gar+04a], [Gar+04b]
RWPA = ratio of work products and activities [Gar+03] Table B.5	Ratio	BPMN	PDOTOut [Rol+06b] Table B.9	[Rol09]	[Rol+08], [Rol+07a], [Gar+04a], [Gar+04b], [Gar+04d], [Gar+04c]	[Gar+03]	[Rol09], , [Rol+08], [Rol+07a], [Gar+04a], [Gar+04b], [Gar+04d], [Gar+04c]
RD = relational density [Kre10] Table B.33	Calculated	Graph				[Kre10]	
CFC_{rel} = relative control-flow complexity [Car08] Table B.20	Calculated	Graph				[Car08]	
$Size_R$ = resource size [Ant+11] Table B.36	Counter	BPMN		[Ant+11]		[Ant+11]	
RT = restrictiveness estimator [Lat01] Table B.4	Calculated	Graph	RT [Car+06] Table B.10		[Lat01]	[Lat01]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	**Validation**		**Sources**	
				Theoretical	**Empirical**	**Primary**	**Secondary**
$\prod$ = separability ratio [Men07] Table B.16	Calculated	EPC			[Men+07c], [MN07], [S+10b], [Men07], [MS08]	[Men07]	[Men+07c], [MN07], [S+10b], [MS08]
`C` = sequence control-flow complexity [Fu+10] Table B.29	Calculated	BPEL		[Fu+10]		[Fu+10]	
Ξ = sequentiality ratio [Men07] Table B.16	Ratio	EPC			[MS08], [Men07], [Men+07c], [S+10b], [MN07]	[Men07]	[MS08], [Men+07c], [S+10b], [MN07]
`Simplicity` = simplicity [Tja99] Table B.3	Percentage	Graph				[Tja99]	
$\texttt{Size}_{DF}$ = size of data flow graph [Ant+11] Table B.36	Counter	BPMN		[Ant+11]		[Ant+11]	
$\texttt{H}_{SC}$ = structural complexity [Che08] Table B.22	Calculated	Graph				[Che08]	
`SCBP` = structural complexity [Muk+10b] Table B.30	Calculated	BPEL		[Muk+10b]	[Muk+10b]	[Muk+10b]	
`SA` = structured activities [HB09] Table B.28	Counter	BPEL	`TSAC` [LÏ5] Table B.40			[HB09]	

Continued on next page

Table B.42 – *Continued from previous page*

Metric	Type	Notation	Same metric	Validation		Sources	
				Theoretical	Empirical	Primary	Secondary
SM = structuredness [Lv09] Table B.25	Algorithm	Workflow-Nets			[Lv09]	[Lv09]	
S = structuredness [La +11b] Table B.35	Calculated	BPMN				[La +11b]	
Φ = structuredness ratio [Men07] Table B.16	Ratio	EPC			[MN07], [Men07], [MS08], [Men+07c]	[Men07]	[MN07], [MS08], [Men+07c]
TS = token split [Men07] Table B.16	Ratio	EPC			[S´+10b], [MS08], [MN07], [Men+07c], [Men07]	[Men07]	[S´+10b], [MS08], [MN07], [Men+07c]
TNS = transfer number per service [Mao10a] Table B.31	Calculated	Petri-Net			[Mao10b]	[Mao10a]	[Mao10b]
Complexity = variety-based complexity [CP08] Table B.23	Calculated	BPMN				[CP08]	
WH = width of hierarchy [Kre10] Table B.33	Counter	Graph				[Kre10]	
W = Wiener's index [Bor+09a] Table B.27	Calculated	Graph			[Bor+09a]	[Bor+09a]	

Appendix C

Ethical Clearance Approval Letter

This appendix provides the approval letter from the College of Science, Engineering and Technology's (CSET) Research and Ethics Committee in response to the application made by the researcher in Appendix D file 15 (2016-05-23 MAMarin_Student_Ethical_Clearance-v5.pdf) to conduct the experiment described in Chapter 8.

UNISA | college of science, engineering and technology

Dear Mr Mike Andy Marin (49097040)

Date: 2016-05-24

Application number:

031/MAM/2016/CSET_SOC

REQUEST FOR ETHICAL CLEARANCE: Exploring complexity metrics and model comprehension for artifacts-based process models

The College of Science, Engineering and Technology's (CSET) Research and Ethics Committee has considered the relevant parts of the studies relating to the abovementioned research project and research methodology and is pleased to inform you that ethical clearance is granted for your research study as set out in your proposal and application for ethical clearance.

Therefore, involved parties may also consider ethics approval as granted. However, the permission granted must not be misconstrued as constituting an instruction from the CSET Executive or the CSET CRIC that sampled interviewees (if applicable) are compelled to take part in the research project. All interviewees retain their individual right to decide whether to participate or not.

We trust that the research will be undertaken in a manner that is respectful of the rights and integrity of those who volunteer to participate, as stipulated in the UNISA Research Ethics policy. The policy can be found at the following URL:
http://cm.unisa.ac.za/contents/departments/res_policies/docs/ResearchEthicsPolicy_apprvCounc_21Sept07.pdf

Please note that the ethical clearance is granted for the duration of this project and if you subsequently do a follow-up study that requires the use of a different research instrument, you will have to submit an addendum to this application, explaining the purpose of the follow-up study and attach the new instrument along with a comprehensive information document and consent form.

Yours sincerely

RECEIVED
2016 -05- 26
OFFICE OF THE EXECUTIVE DEAN
College of Science, Engineering and Technology

Adele da Veiga

Dr. A Da Veiga
Chair: Ethics Sub-Committee School of Computing, CSET

Prof I. Osunmakinde
Director: School of Computing, CSET

Prof I. Alderton
Executive Dean (Acting): College of Science, Engineering and Technology (CSET)

University of South Africa
College of Science, Engineering and Technology
The Science Campus
C/o Christiaan de Wet Road and Pioneer Avenue,
Florida Park, Roodepoort
Private Bag X6, Florida, 1710
www.unisa.ac.za/cset

UNISA | college of science, engineering and technology

Appendix D

Supplementary Material

There is a companion CD to this thesis containing the supplementary material described in this appendix.

D.1 Data Sets

The following files correspond to the data collected during the empirical validation described in Chapter 8. These can be found in the supplementary material media under the *data* folder.

File 1: *dataset-all(description).pdf*. This document describes all of the variables present in the data-set – variables in files 2 (dataset-all.csv) and file 3 (dataset-clean.csv).

File 2: *dataset-all.csv*. This comma separated value file contains all of the data collected from the survey, including data from incomplete surveys. This file was generated using the raw data file 6 (in-survey-data-file.csv) and file 53 (CMMN-Convert-File.Rmd).

File 3: *dataset-clean.csv*. This comma separated values file contains the data set containing only complete and usable surveys. It is a subset of file 2 (dataset-all.csv).

File 4: *in-independent-variables-map.csv*. This comma separated values file contains a mapping of the independent variables with each of the 30 sets. Each set corresponds to two models (model A and model B) extracted from the six models that were tested (see 27 (The6Models.pdf)).

File 5: *in-independent-variables.csv*. This comma separated values file contains the calculated value of the independent variables.

File 6: *in-survey-data-file.csv*. This comma separated values file contains the raw data from LimeSurvey after being converted to a file format suitable for use with R. This file was generated from raw data file 12 (results-survey338792.csv) using file 57 (copy-and-fix-file.r).

File 7: *in-survey-var-names.csv*. This comma separated values file contains the survey variable names.

File 8: *in-weights-scaled-ordinal-rounded.csv*. This comma separated values file contains the independent variable weight scaled to the ordinal values.

File 9: *in-weights.csv*. This comma separated values file contains the independent variables for the weights used to calculate CC.

File 10: *out-comments.txt*. This file contains the comments left by the survey participants. The data was extracted from the last optional question in the survey ("Any final comments that you may want to share with the research team?").

File 11: *results-survey338792 (description).pdf*. This document is the LimeSurvey logic file describing all of the questions included in the survey. Each subject was exposed to a subset of the questions described in this document.

File 12: *results-survey338792.csv*. This comma separated values file contains the raw data exported from LimeSurvey.

File 13: *survey_archive_338792.lsa*. This file is a LimeSurvey archive containing the survey and the responses from all of the subjects.

D.2 Documents

The following documents are related to this thesis. These can be found in the supplementary material media under the *docs* folder.

File 14: *2012-10-31 MMarin DPSET02 Proposal.pdf*. This document contains the research proposal presented to University of South Africa (UNISA) for this research.

File 15: *2016-05-23 MAMarin_Student_Ethical_Clearance-v5.pdf*. This document contains the research ethical clearance application form presented to the College of Science, Engineering and Technology's (CSET) Research and Ethics Committee.

File 16: *2016-06-08 Pilot-full-answers.pdf*. This is the output of LimeSurvey containing all of the responses from the pilot survey that included 12 subjects who were used to test the survey instrument.

File 17: *2016-06-10 Survey and Tutorial Pilot.pdf*. This is a short report describing the pilot that was conducted for the survey and the tutorial.

File 18: *2016-06-15 Invitation.pdf*. This document contains the invitation that was posted in the Case Management Model and Notation (CMMN) [OMG14a]'s Linkedin group soliciting participants for the survey and tutorial. Similar invitations were posted in other Linkedin groups or emailed to potential participants.

File 19: *2016-06-15 Survey-Example.pdf*. This a sample survey generated by LimeSurvey. This sample survey uses models one and two, of the six possible models. Each survey presents the subject with only two models, and the possible responses to the questions are ordered in a random fashion, so that almost every subject gets a slightly different version of the survey.

File 20: *2016-06-15 Survey-Tutorial.pdf*. This is a textual version of the CMMN tutorial that was used for the survey. Each page on this document corresponds to a webpage presented to the subjects.

File 21: *2016-07-05-ART-Case-Management-Modeling-MMarin.pdf*. This document contains a short article published in BPTrends [Mar16b] soliciting participants for the survey and tutorial.

File 22: *2016-07-06 Column 2.pdf*. This document contains a blog post by Kemsley [Kem16] soliciting participants for the survey and tutorial.

File 23: *2016-08-16 Charity-donations.pdf*. This file contains the receipts or emails confirming payment to the charities that subjects participating in the survey selected. In total $510.00 was paid to the different charities.

File 24: *2016-08-16 CMMN-basic-stats(raw-dataset).pdf*. This document was generated using R and contains basic statistics that were calculated using the raw data generated from the survey. This file was emailed to the subjects who had requested to be informed of the outcome of the survey.

File 25: *Basic-stats.pdf*. This document was generated using R and contains the statistical analysis of the survey using the completed and valid survey data (a subset of the raw data).

File 26: *Cherries.pdf*. This document contains the completed Checklist for Reporting Results of Internet E-Surveys (CHERRIES) [Eys12].

File 27: *The6Models.pdf*. This document describes how the researcher arrived at the six models that were used for the empirical validation conducted in Chapter 8.

File 28: *SLR-analysis.pdf*. This document was generated using R and contains the full statistical analysis for the Systematic Literature Review (SLR).

File 29: *SLR-Response from University Putra Malaysia.pdf*. This file contains the email response from the University Putra Malaysia to the inter-library loan requested by a UNISA librarian.

File 30: *FSM-2-GSM.pdf*. This document is an extract from Marin et al. [Mar+16] describing the transformation of Deterministic Finite State Machine (DFSM) into Guard-Stage-Milestone (GSM) [Hul+11b] types.

D.3 Systematic Literature Review of Metrics

State of the Art through Systematic Reviews (StArt) [Fab+16] was used for the SLR presented in Chapter 6.

File 31: *report(full).xls*. This spreadsheet was generated using StArt, and contains all of the informations from the study selection activity performed in the SLR.

File 32: *Analysis.xlsx*. This spreadsheet contains all of the information extracted from the papers during the data extraction activity performed in the SLR.

File 33: *StArt/SLR-BPM_Metrics.start*. This file is in the StArt file format and contains all of the information from the study selection activity. This file can be read using StArt version 2.3.4.

File 34: *bibtex/2016-07-01_ACM.bib*. This file contains the results from the ACM digital library query.

File 35: *bibtex/2016-07-02_Scopus.bib*. This file contains the results from the Elsevier's Scopus query.

File 36: *bibtex/2016-07-02_Web-of-science.bib*. This file contains the results from the Thomson Reuters' Web of Science query.

File 37: *bibtex/2016-07-03_Google-academic.bib*. This file contains the results from the Google Scholar query.

File 38: *bibtex/2016-07-03_IEEE-1.bib*. This file contains the results from the first IEEE Xplore digital library query.

File 39: *bibtex/2016-07-03_IEEE-2.bib*. This file contains the results from the second IEEE Xplore digital library query.

File 40: *bibtex/2016-07-03_Science-direct.bib*. This file contains the results from the Elsevier's Science Direct query.

File 41: *bibtex/2016-07-03_Springer.bib*. This file contains the results from the Springer's SpringerLink query.

File 42: *data/in.slr.raw.report.csv*. This file contains the list of all of the papers that were reviewed during the SLR. This file is exported from file 31 (report(full).xls) in a comma separated values file format that can be read by an R program.

File 43: *data/in.slr.raw.dup-metrics.csv*. This file contains the duplicated metrics that were identified during the review. This file was exported from file 32 (Analysis.xlsx) in a comma separated values file format that can be read by an R program.

File 44: *data/in.slr.raw.metrics.csv*. This file contains all of the metrics identified by the review. This file was exported from file 32 (Analysis.xlsx) in a comma separated values file format that can be read by an R program.

File 45: *data/in.slr.raw.papers.csv*. This file contains all of the papers that were accepted during the selection activity. This file was exported from file 32 (Analysis.xlsx) in a comma separated values file format that can be read by an R program.

File 46: *data/in.slr.raw.theor.vali.csv*. This file contains the theoretical validation information extracted from the papers in this review. This file was exported from file 32 (Analysis.xlsx) in a comma separated values file format that can be read by an R program.

File 47: *data/in.slr.raw.validated-metrics.csv*. This file contains a record for each metric and for each empirical validation performed in that metric. This file was exported from file 32 (Analysis.xlsx) in a comma separated values file format that can be read by an R program.

File 48: *data/in.slr.raw.validation.csv*. This file contains information for each of the validation studies identified during the SLR. This file was exported from file 32 (Analysis.xlsx) in a comma separated values file format that can be read by an R program.

File 49: *data/SLR-data(variable-description).pdf*. This file describes the variables used in all of the comma separated values files. The variable names correspond to the names used in the R programs.

D.4 Sources

During the production of this thesis several sources were created. These can be found in the supplementary material media under the *src* folder.

D.4.1 R

R [R C16] was used to perform most of the statistical calculations. With the exception of power calculations that were done using G*Power 3 [Fau+07]. Some of the files in the R folder include:

File 50: *Instructions(read-me-first).pdf*. Basic instructions on how to perform and build the reports for the empirical validation's statistical analysis.

File 51: *CMMN-basic-stats.Rmd*. Generate basic demographic statistics.

File 52: *Basic-stats.Rnw*. This file generates the basic statistical analysis for the survey used in the empirical validation.

File 53: *CMMN-Convert-File.Rmd*. Script that generates the dataset-all.csv, and dataset-clean.csv files.

File 54: *CMMN-Sample.Rmd*. Compares the data set against the expected sample size for each experiment.

File 55: *CMMN-Weights.Rmd*. Recalculates *CC* (`iv.A.CC`, `iv.B.CC`, and `iv.C.CC`) and generates the dataset-clean-post.csv.

File 56: *Results.Rnw*. Contains the main statistical analysis for the survey used for the empirical validation. Part of the output of this file is automatically included in Section 8.3.

File 57: *copy-and-fix-file.r*. Script used to copy and fix the LimeSurvey exported file.

File 58: *daily.r*. Main R script that calls all *.Rmd scripts.

File 59: *share-my-functions.r*. A set of statistical analysis functions that were developed for the empirical validation.

File 60: *share-read-dataset.r*. A set of functions used to implement a common way to read the data sets.

File 61: *SLR-analysis.Rmd*. This file contains the processing and analysis of the SLR data sets. Part of the output of this file is automatically included in Section 6.4.

D.4.2 eXeLearning

EXeLearning [eXe15] was used to create the online tutorial.

File 62: *README.txt*. This file contains instructions on how to update the CMMN tutorial using the files in this directory.

File 63: *Tutorial.pdf*. This is a pdf version of the tutorial.

File 64: *Tutorial.zip*. This compressed file contains the tutorial as a deployable web application.

File 65: *pics.zip*. This compressed file contains all of the figures used in the tutorial.

File 66: *script.vim*. This script improves the page navigation of the tutorial.

File 67: *tutorial.elp*. eXe Learning version 2.0.4 source of the tutorial.

D.4.3 LimeSurvey

LimeSurvey [Lim16] was used to develop and to run the online survey.

File 68: *CMMN Complexity metrics project.pdf*. LimeSurvey logic file for the CMMNcomplexity metrics survey.

File 69: *README.txt*. This file describes how to modify and use the LimeSurvey files in this directory.

File 70: *limesurvey_survey_338792.lss*. LimeSurvey version 2.06lts containing the source of the CMMNcomplexity metrics survey.

File 71: *resources-survey-338792.zip*. This compressed file contains all of the figures used in the CMMNcomplexity metrics survey.

D.4.4 MiniZinc

MiniZinc [Net+07] was used to model and solve the constraints to identify the six models used in the online survey. The MiniZinc folder contains all of the source and data files that were used to solve the constraints required to create the six models with $CTS = 90$, and the other metrics with very different values.