Table of Contents

List of Figures

List of Tables

List of Definitions

List of Code Listings

1 Introduction

Currently, knowledge and knowledge work are the key factors of successful companies – there is a clear shift from routine work to knowledge work during the current era (El-Farr, 2009). The term knowledge work "[…] indicate[s] the knowledge intensiveness of the current working tasks and the required abilities, skills, qualifications and working conditions for employees to accomplish their work" (El-Farr, 2009, p.3). Individuals who perform knowledge work act in different roles and drive the innovation in companies. Davenport (2010, p.18) has stated that "without knowledge workers, there would be no new products and services, and no growth". Further, he indicated that individuals who perform knowledge work have "[…] high degrees of expertise, education or experience and the primary purpose of their jobs involves the creation, distribution, or application of knowledge" (Davenport, 2005b, p.10).

The following sections describe the background of the study and introduce the underlying research questions.

1.1 Background

Conventional business process management has been exceedingly successful for routine work but has deficiencies in dealing with flexibility, which is needed to perform knowledge work. Process models may not be in accordance with the real work that must be done. Classical workflow[1] management systems are well suited to supporting the execution of rigidly structured business processes but fail to allow changes in unexpected situations (Adams, Edmond and ter Hofstede, 2003). Van der Aalst et al. (2005, p.131) have stated that workflow management systems suffer from a lack of flexibility, based on "[…] the fact that routing is the only mechanism driving the case". Process flexibility can be required at both build-time and run-time (Dadam, Reichert and Rinderle-Ma, 2010). During build-time, it is desirable to be able to change a workflow definition without major effort and during run-time it might be necessary to adapt and determine the control flow as a reaction to unforeseeable events. For knowledge workers, it is recommended to provide the flexibility to determine their processes continuously – thus "making the knowledge worker's work more productive and focused, in addition to minimizing their stress and increasing interaction"(El-Farr, 2009, p.8).

Knowledge work cannot be represented sufficiently in traditional business process management, where the work can be structured and described in advance. It is especially difficult to predict upcoming tasks because knowledge work deals with many different requirements simultaneously. Type and scope of tasks are hard to determine in advance. The sequence of tasks may vary due to already achieved results and unforeseeable events. Knowledge work is not routine work; "[…] the sequence of actions depends so much upon the specifics of the situation […] necessitating that part of doing the work is to make the plan itself" (Swenson, Palmer and Silver, 2011, p.8). It is not always possible to define the entire structure including all elements of a knowledge-intensive process at build-time or just before instantiation. This lack of definition becomes evident when dealing with business process standardisation. Schäfermeyer, Rosenkranz and Holten (2012, p.268) have revealed that "[…] higher standardization effort

[1] Workflow: whole or partial automation of a business process (Swenson, 2010).

cannot compensate for higher business process complexity to ensure business process standardization".

Conversely, routine work is well-defined and repeatable. It can be described in traditional business processes execution means. However, knowledge work can benefit from business process management – even in unstructured knowledge work, it might be possible that a number of structured elements or process fragments can be identified. Although it is not adequate to describe the entire knowledge work using business process methodology, it certainly makes sense to place knowledge work within a process-oriented direction. This process orientation implies that a number of structured elements, which can exist in knowledge work, can be captured as process fragments. This would make structured elements and process fragments more efficient without losing the necessary flexibility for non-routine and knowledge intensive activities. The process orientation "[…] can make knowledge work more productive" (Davenport, 2005a, p.4). Moreover, knowledge workers "[…] would benefit from the discipline and structure that a process brings, while remaining free to be creative and improvisational when necessary and desirable" (Davenport, 2010, p.19).

This study created an approach that improves knowledge work by placing it in a process orientation regarding combining business process management and knowledge work.

1.2 Problem Statement and Purpose of the Study

Traditional workflow approaches do not provide the flexibility required in executing knowledge work. There are several approaches for dealing with flexibility. Examples of such flexible process execution[2] approaches, as

[2] Process execution: manual or automatic execution of a business process

introduced by Endl et al. (1998) or more recently by Feldkamp et al. (2007), attempt to overcome a lack of flexibility by using business rules. These process execution approaches are a suitable way to provide more flexibility to certain knowledge-intensive tasks where the actual sequence of work is unforeseeable. However, the rules must be defined and approved, which still does not provide the needed flexibility at run-time for new and unforeseen situations. They typically do not place decision power into the hands of the knowledge worker. Hence, flexibility in business processes using business rules is addressed at build-time by modifying the business rules and process model by a domain expert.

In a combined knowledge-intensive approach, routine and knowledge work can exist within the same process model (see Figure 1.1). An example of a modelling approach that named the knowledge-intensive portion of the process explicitly was introduced by Hinkelmann et al. (2002). They defined these parts as knowledge-intensive tasks based on the work of Abecker et al. (1998). The ad-hoc element of BPMN 2.0 (OMG, 2011) offers the possibility to specify tasks and data of knowledge work without determining the flow of work (see Figure 1.1). The remaining question regarding the knowledge-intensive processes is then, to which level of detail the process structure can and should be determined and at which time, i.e. at build-time or when instantiating the process.

In order to execute knowledge-intensive processes and tasks this execution requires knowledge, which is needed for process and task execution. This so-called functional knowledge includes skills and experience. It is common in knowledge-intensive environments that individuals' functional knowledge is either found within the mind or written down in forms and documents (Nägele and Schreiner, 2002). In addition to functional knowledge, the knowledge concerning the process itself (knowledge about the flow of the process, activities and their relationships) is also necessary knowledge-intensive processes. This knowledge is called process knowledge and it is primarily represented in business process models (Nägele and Schreiner, 2002), i.e. sequences of tasks and control constructs such as gateways and roles.

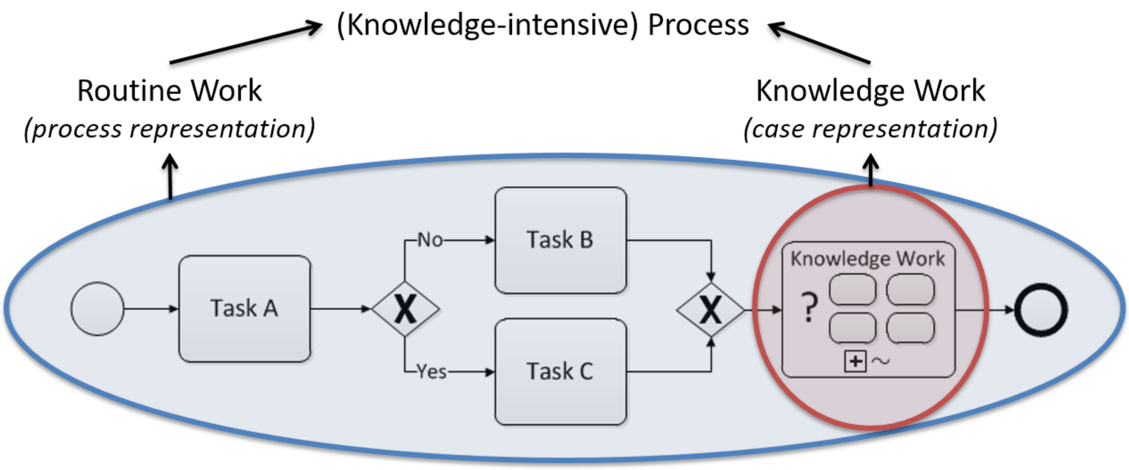

Figure 1.1: Knowledge-intensive Process containing Routine and Knowledge Work

In a combined approach, functional and process knowledge needs to be captured and provided for executing knowledge-intensive processes. Recent work (see Section 2.3) indicates that it is possible and worthwhile to capture functional and process knowledge in a number of cases. These cases, describing functional and procedural knowledge of process instances or instances of process fragments, make it possible to solve a new problem based on similar "old" cases. Case-based reasoning (CBR) uses the knowledge of previously experienced cases to propose a solution to a problem. In order to retrieve, reuse, revise, retain and store functional and process knowledge, case description is a critical issue. The knowledge must be made explicit and represented in a way that allows for machine processing as well as remain understandable (cognitively adequate) to humans. The implication, then, is that the case description language should be as natural as possible in order to gain wide acceptance from the end-user. On the contrary, the case description should be able to be processed by a computer using similarity and adaptability mechanisms during process execution.

When executing knowledge-intensive processes, the context of the process execution can be considered when assigning tasks to individuals, deciding on the next tasks or executing the task itself. The context contains knowledge regarding the process instance itself including information concerning used resources (such as documents or system data), actions (activities or tasks), actors (people or systems who execute the process or certain activities) and further workflow data. When describing functional and process knowledge as cases and turning these cases into execution, the process context can contribute to improving the retrieval of these cases. Therefore, the case description should contain attributes which are linked to the context. When adapting an "old" case to the new problem, a case adaptation mechanism is needed. Such an adaptation mechanism must deal with a different granularity of process fragments (the solution parts) or sub-processes that refer to other cases and once more trigger a case retrieval.

There is a gap between the literature and environmental requirements, such as business needs, with regard to a comprehensive CBR and process execution approach for knowledge-intensive work that incorporates knowledge from enterprise models and architectures. Therefore, CBR and process execution requires more attention concerning the integration of both parts. Thus, the following requirements can overcome these deficits when utilised within an approach which:

1. allows for maintaining case descriptions that contain functional and process knowledge, are cognitively adequate to humans and are computer processable;
2. provides CBR services that use the process execution context and enterprise knowledge for similarity, adaptation and learning;
3. provides an integration mechanism for existing enterprise knowledge from enterprise models, architectures and repositories.

1.3 The Objectives and Goals of the Study

The main purpose of this work is to investigate a new approach for supporting the modelling and execution of knowledge-intensive processes in order to improve knowledge work. It is the aim of the approach to support execution of knowledge work and provide the needed structure and flexibility to knowledge workers.

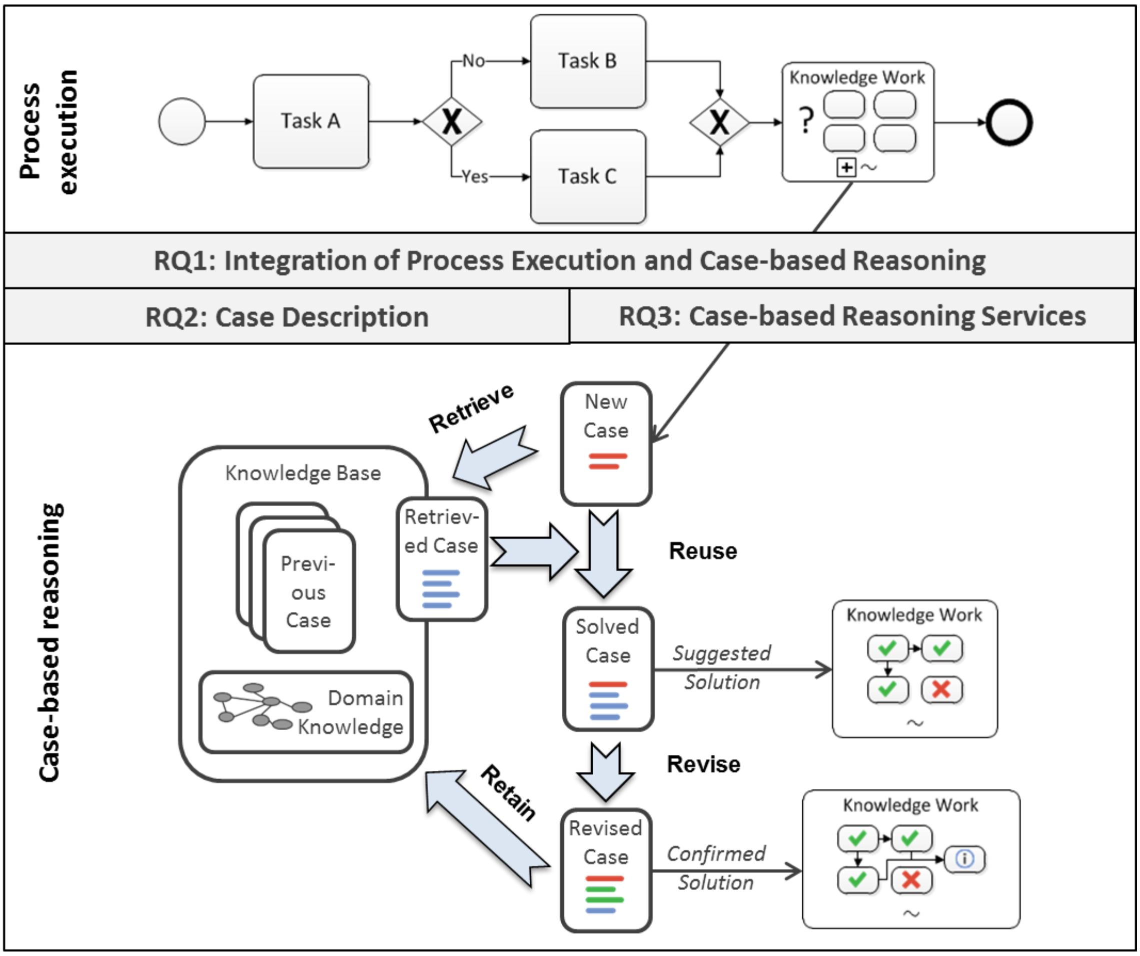

Figure 1.2: Research Goals (and Research Questions) represented as a Sketch

As mentioned in Section 1.2, it is possible and worthwhile to capture functional and process knowledge in such cases. In order to retrieve and adapt cases within a process execution environment in an automatic or semi-automatic way, a process execution environment requires the inclusion of a CBRsystem. As such, a CBR system provides the possibility to retrieve, reuse (and adapt), revise and retain cases containing the functional and process knowledge of knowledge intensive processes. However, CBR is not enough. It needs to be combined with the facility of the process execution in order to attain the benefits of both approaches.

This study investigates the adaptation of CBR in this means. Figure 1.2 shows a sketch of the proposed approach including the research questions (see 1.4.2 Research Questions). The figure displays both the process execution portion and the CBR portion (represented as adapted CBR-cycle, see Section 2.2). The depicted process consists of structured and knowledge-intensive elements. The knowledge-intensive portion is complemented by case recommendations - users can choose to follow such recommendations.

The case description is an important aspect of the approach. It should consider the process execution context (as described within the previous section) and contain functional and process related knowledge. Furthermore, the approach considers the need for a user-friendly case description. However, the case description should be processable by the CBR mechanisms.

Based upon the research problem (see Section 1.2), the main goal of this study is to introduce a new approach, called ICEBERG-PE[3], that combines and integrates CBR and process execution in order to retrieve, reuse, revise, retain and "execute" functional and process knowledge to improve knowledge-intensive work.

A prototype system is developed as a proof-of-concept for evaluating and demonstrating the approach including the sub-goals. In order to reach the main goal, the following two sub-goals are addressed:

1. The elaboration of a case description for knowledge-intensive processes that contains functional and process knowledge, can be integrated into a

[3] ICEBERG-PE: **I**nterlinked **C**as**e**-**b**as**e**d **R**easonin**g** for **P**rocess **E**xecution

process execution system, considers the process execution context, and is machine processable and still cognitively adequate.

2. The investigation and development of case-based reasoning services, which can be used for the retrieval, adaptation and learning of functional and procedural knowledge.

1.4 Thesis Statement and Research Questions

Based on the previous statements concerning research problem and goals, research questions and a thesis statement are defined. Based on Creswell (2008) qualitative researchers usually write at least one main research question and sub-questions.

1.4.1 Thesis Statement

The following thesis statement guides the research project:

"It is possible to improve knowledge work by using process execution context to retrieve, reuse, revise, retain and store cases that contain functional and process knowledge."

1.4.2 Research Questions

To combine case-based reasoning and process execution it is necessary to focus on the main- and the three sub- research areas. An approach will be developed to show that the mentioned combination is possible. The approach consists of (1) an overall integration approach for case-based reasoning and process execution, a (2) case description for knowledge-intensive work including a way to integrate and access enterprise knowledge and (3) case-based reasoning services (see Figure 1.2). Therefore, the research is divided into three areas, and the following three research questions (RQ) are defined.

1.4.2.1 Integration of process execution and case-based reasoning

The main goal of this study is to introduce an approach to (RQ 1) integrate process execution and case-based reasoning. Therefore, the following research question is defined as:

RQ 1: How can case-based reasoning be integrated with process execution?

This research question was used to guide the suggestion phase during the construction of the approach and conceptual model. Based on this conceptual groundwork the following research question could be addressed.

1.4.2.2 Case Description for Knowledge-intensive Work

The integration of process execution and case-based reasoning supports knowledge work by describing functional and process knowledge as cases (RQ 2). The case description assists knowledge workers when they perform knowledge intensive work. Therefore, it is needed to consider the needs and requirements of the knowledge worker as well. The following research question was used to address this requirement:

RQ 2: What should the case description for knowledge-intensive work consist of?

The case description might contain some elements from structured processes and unstructured knowledge work and remains understandable (cognitively adequate) to humans. Therefore, the following sub- question was defined:

RQ 2.1: How can functional and process knowledge be included in a case description that is cognitively adequate to humans?

It might be required to capture and structure certain elements of the process execution context for in the case description in an explicit way to allow machine support. Therefore, the following research question was defined:

RQ 2.2: How can the process execution context be integrated into the case description?

1.4.2.3 Support of Process Execution by Case-based Reasoning Services

To retrieve, adapt and learn functional and process knowledge that has been made available in cases, cease based reasoning services (RQ 3) are needed. Therefore, the following research question was defined:

RQ 3: How can case-based reasoning services support process execution?

The case retrieval (RQ 3.1) is the first step to provide knowledge or guidelines to a specific problem. A similarity mechanism is needed to determine the similarity between cases. Therefore, the following research question was defined:

RQ 3.1: How can the similarity between cases for knowledge work be calculated?

Case-based reasoning uses domain knowledge for performing its tasks and business process execution provides contextual information. Both portions' domain knowledge and contextual information can be useful for the retrieval of cases as well as the suggestion or adaptation of case file items. Potential case file items, which can be considered, are information resources (such as documents or emails), how-to knowledge objects (such as instructions or guidelines) or plan items up to ad-hoc tasks. Therefore, the following research question is defined:

RQ 3.2: How can domain knowledge and contextual information be used for retrieval of cases and suggestion or adaptation of case items?

1.5 Research Strategy

This research is based on the design science paradigm. This paradigm is adequate for addressing the research problem by building artefacts and investigating them based on relevant use cases or application scenarios.

Such a design science based research approach based on Vaishnavi and Kuechler (2004) consists of five elementary and iterative phases: (1) awareness, (2) suggestion, (3) development, (4) evaluation and finally (5) conclusion.

1.6 Scope, Research Subjects and Limitations

The following three subsections create the fixed framework of this study.

1.6.1 Scope of the Study

This study introduces an approach that deals with *knowledge-intensive work* in a process-oriented environment. This approach contains a case description for capturing functional and process related knowledge, a similarity mechanism to retrieve the provided knowledge and an adaptation mechanism to propose a solution to the problem.

This study falls within the field of knowledge and process management and is based on an application scenario that was derived from a real-world scenario in the public sector.

1.6.2 Research Subjects

The research subjects of this research were two organisations from the public sector and software industry, which were acting as application partners. These two organisations were involved either directly by serving the application scenario and primary data or indirectly by providing first-hand secondary data. Please refer to subsection 3.3.2 for a description of potential research methods for the data collection and analysis, and to subsection 3.4.2 describing which research subjects were involved to collect the data. The application data, which has been acquired from the application partner, has been triangulated for the application scenario (see section 4.1) and for the evaluation (see section 9.1) as well.

- The main research subject, which was coming from the public section, was the admission process of Master of Science (MSc) programmes at the school of business/FHNW University of Applied Sciences and Arts Northwestern Switzerland. Two primary data collection techniques, (1) interviews and (2) document and artefact study, were used to gain the application scenario and evaluation data. The stakeholders of the admission process were interviewed and selected because they work within the context of the process. Seven potential participants were working within the context of the application scenario. Moreover, the minimal interviewee sample size was three.

- The secondary research subject, the ELO Digital Office CH AG, was an application partner of the research project [sic!], in which a CBR system for the offer process and project management of a software company was developed. Through this research subject inclusion, it was possible to derive further requirements and to have access to first-hand secondary data and primary data specifically for this thesis. The elicited offering and project management scenario is used in the confirmatory evaluation as described in subsection 9.4.1.

1.6.3 Limitations

This research was conducted with two organisations and application partners from the public sector and software industry. They were engaged as domain specialists within the first three phases of the design research process (see Section 3.4) and were involved significantly within the evaluation phase. In general, there was a close relationship between the researcher and the domain specialist for gaining relevant research results. Further limitations are as follows:

- Limitation on case characterisation retrieval: This study focuses on the case characterisation rather than on the content in the similarity perspective. Nevertheless, the demonstrator shows the potential of providing knowledge items using a CBR approach. The retrieval is based on the case characterisation exclusively and not on the case content. Therefore, no process model graph similarity analysis is applied.

- Limitation on flexibility: This study is limited to the flexibility dimensions of processes and workflows as defined by Sadiq et al. (2001).
- Workflow built-time adaptation: It is not intended to provide workflow adaptation or any other changes to the workflow model or meta-model during build-time. Much research has already been done (see Section 2.3).
- Workflow technology: It is also not intended to investigate or modify the used workflow technology.

1.7 Rationale of this Study

The results of this study provide a contribution to the process and knowledge management community within the business or academic fields.

Knowledge-intensive processes are critical for the success of a company and occur, for example, in strategic management, product innovation, planning and whenever complex decisions need to be made. While the management of structured business processes is routine for a majority of companies, it is still challenging to deal with knowledge-intensive processes.

Increased flexibility and agility, co-operation, continuous improvement and the progress towards learning organisations are critical challenges for enterprises. The results of this study assist in meeting these challenges. The main artefact, the ICEBERG-PE prototype, is envisioned as a showcase for prospective and interested companies regardless of their domain. The outcomes of this research can be generalised and adapted to other specific working environments. In principle, the significance cannot be fixed to a specific sector of business, but is relevant for a company that runs knowledge-intensive work.

1.8 Contribution

The main outcome (artefact) of this research work is an approach for knowledge-intensive work which combines CBR and process execution approach. This approach is then implemented in a prototype system. In addition, the approach is divided into sub-artefacts; These are an ontology, a procure model, a case model, CBR services and a prototype.

The proposed scientific contribution is a new approach to supporting the execution of knowledge-intensive business processes by adopting CBR. It consists of a description of the approach, a real-world application scenario including test data and a reusable prototype for running further experiments.

From its inception, this research work has considered the real-world context. This consideration ensures that the results contribute to the business practice. It is expected that the results can be transferred to similar application areas.

This section lists the publications where results of this thesis have been published.

- Martin, Emmenegger and Wilke (2013): Martin, A., Emmenegger, S. and Wilke, G., 2013. Integrating an enterprise architecture ontology in a case-based reasoning approach for project knowledge. In: *Proceedings of the First International Conference on Enterprise Systems: ES 2013*. IEEE, pp.1–12.
- Witschel, Martin, Emmenegger and Lutz (2015): Witschel, H.F., Martin, A., Emmenegger, S. and Lutz, J., 2015. A new Retrieval Function for Ontology-Based Complex Case Descriptions. In: *International Workshop Case-Based Reasoning CBR-MD 2015*. Hamburg: ibai-publishing.
- Cognini, Hinkelmann and Martin (2016): Cognini, R., Hinkelmann, K. and Martin, A., 2016. A Case Modelling Language for Process Variant Management in Case-Based Reasoning. In: M. Reichert and A.H. Reijers, eds., *Business Process Management Workshops: BPM 2015, 13th International Workshops, AdaptiveCM 2015: 4th International Workshop on Adaptive Case*

Management and other non-workflow approaches to BPM, Innsbruck, Austria, August 31 -- September 3, 2015, Revised Pa. Cham: Springer International Publishing, pp.30–42.

- Martin, Emmenegger, Hinkelmann and Thönssen (2016): Martin, A., Emmenegger, S., Hinkelmann, K., and Thönssen, B., 2016. A Viewpoint-Based Case-Based Reasoning Approach Utilising an Enterprise Architecture Ontology for Experience Management. *Enterprise Information Systems*.
- Emmenegger et al. (forthcoming 2017): Emmenegger, S., Hinkelmann, K., Laurenzi, E., Martin, A., Thönssen, B., Witschel, H.-F. and Zhang, C., 2017. An Ontology-based and Case-based Reasoning supported Workplace Learning Approach. In: *Model-Driven Engineering and Software Development, Communications in Computer and Information Science (CCIS), In Press*. Springer Berlin / Heidelberg, p.23.

1.9 Structure of the Thesis

This section briefly describes the chapters and content of this thesis. The interrelationships of the chapters can be seen in Figure 1.3 – which is a thesis chapter map. Before introducing the chapter map, it is necessary to clarify the terms thesis and dissertation. The work described within this proposal is a requirement for obtaining the academic degree of doctor of philosophy (PhD) in information systems (IS) from the University of South Africa (UNISA). The work is a requirement for reaching the NQF[4] Level 10, and it is called a thesis. A dissertation is generally recognised as a requirement for obtaining a Master's degree in South Africa and other countries. There is no joint definition of the

[4] NQF: The National Qualifications Framework (NQF) overseen by the South African Qualifications Authority (SAQA)

terms **thesis and dissertation** – in Switzerland and a number of other countries, these words mean the opposite.

This thesis is divided into ten chapters, three introduction chapters and seven chapters of the main body.

1. The *Introduction* chapter, as the name suggests, introduces the thesis including the research objective and goals, research questions and thesis statement as well as a brief introduction to the topic in general.

2. The *Theoretical Framework* contains all relevant aspects of the literature review. It is the theoretical basis used to answer the research questions.

3. The *Research Methodology and Design* chapter contain several subsections, which can be used as guideline and process for conducting this research work. These subsections are research philosophy, research approach, research strategy, research design, data collection, analysis and triangulation and ethical considerations.

4. The *Problem Relevance and Application Scenario* chapter summarises the needs of practitioners and the scientific community concerning the topic of this research. The application scenario is used as demonstration material and source of requirements, which is addressed and described within the following implementation chapters.

5. The *Ontology-based CBR and Process Execution Approach* chapter contains a view on the proposed approach as a suggestion, based on the literature study and the application scenario.

6. The *Case Model* chapter contains the results of the creation and development of the case model including content and its characterisation.

7. The *Case-based Reasoning Services* chapter contains a description of the implemented CBR services (retrieval, adaptation and learning) that can be used during process execution and case management.

8. The chapter *Ontology-based CBR and Process Execution Prototype* reflects the results of the former chapters and describes a prototypical instantiation of the approach including its system architecture description.
9. The prototype is evaluated during the evaluation phase and the results and findings are presented within the *Evaluation* chapter.
10. Finally, this thesis ends with a *Discussion, Conclusion and Future Work*.

The described structure is visualised in Figure 1.3. Apart from the described chapters, this thesis contains a bibliography chapter and appendices.

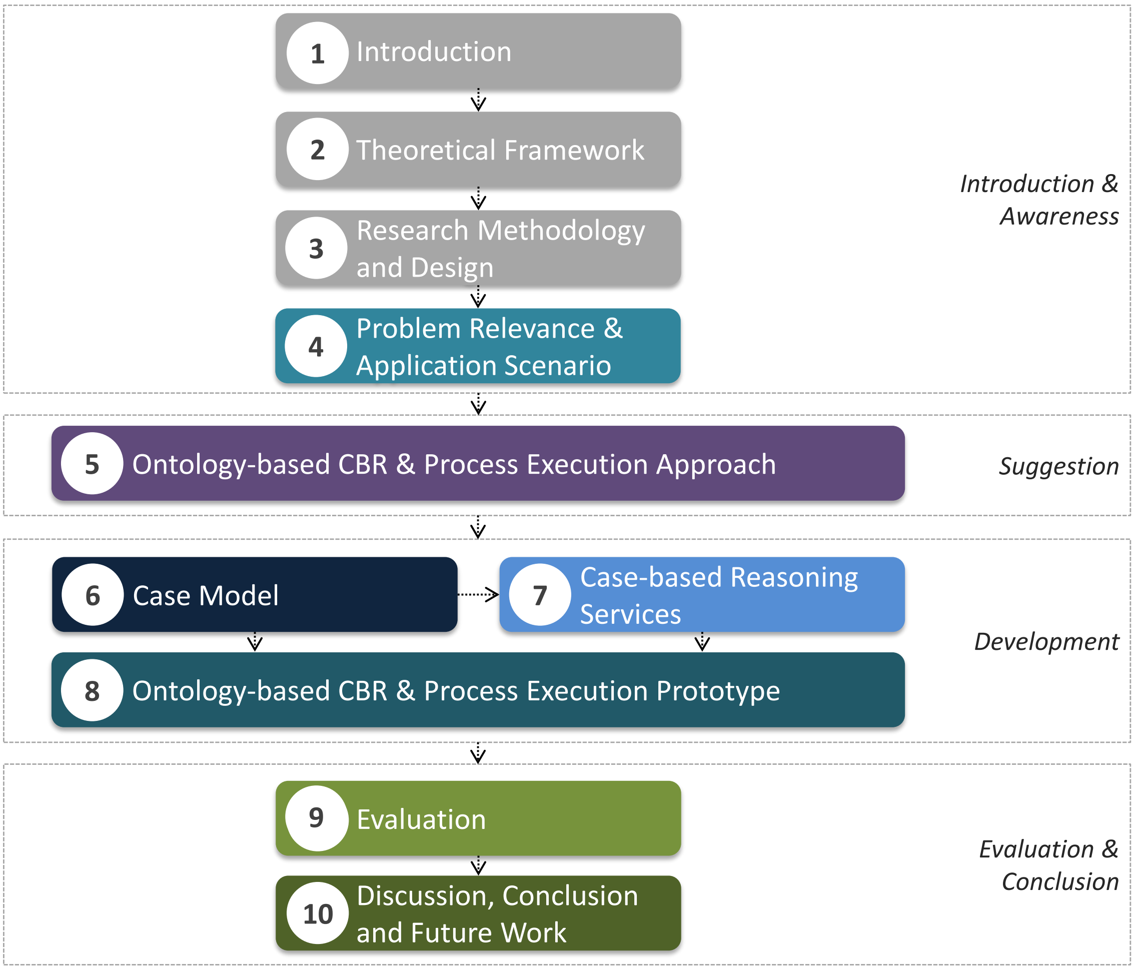

Figure 1.3: Thesis Map

2 Theoretical Framework

This chapter gives an outline of the theoretical framework of the thesis. It introduces the three relevant aspects, which supporting this research work: process execution and flexibility, case-based reasoning (CBR), and the proposed combination of CBR and process execution to support flexibility.

2.1 Process Execution and Flexibility

Process execution refers to the accomplishment of roughly, partially or entirely predefined processes by humans and/or information systems. According to the Workflow Management Coalition (WfMC, 1995, p.7) a process definition or model is a "[...] computerised representation of a process that includes the manual definition and workflow definition". This definition is rather narrow since it defines that a process model has to be computerised at all. In the following, this definition is extended:

> **Definition 2.1: Process Model/Definition**
> A process model or definition is a representation of a business process that includes a manual work definition and/or computerised workflow definition (adapted from WfMC (1995, p.7)).

In addition to this, a "[…] process model captures the activities to be executed, their control and data flow, the organizational entities performing the activities, and the data objects and documents accessed by them" (Reichert and Weber, 2012, p.15). The process models can serve as the basis for the execution of the containing process logic. Since most process-oriented information systems "[…] describe process logic explicitly regarding a process model providing the schema for process execution" (Reichert and Weber, 2012, p.4). The term process execution is related to and used interchangeably with the term workflow, which can be defined as follows:

Definition 2.2: Process Execution/Workflow
Process execution or workflow is the "[…] computerised facilitation or automation of a business process, in whole or part" (WfMC, 1995, p.6).

The computerised facilitation or automation of a business process, the process execution, is usually done with the support of a workflow management system.

Definition 2.3: Workflow Management System
A workflow management system "[…] completely defines, manages and executes "workflows" through the execution of software whose order of execution is driven by a computer representation of the [process] logic" (WfMC, 1995, p.6).

The specific execution component, the core element, of a workflow management system is called workflow engine or process execution engine.

Definition 2.4: Workflow Engine/Process Execution Engine
A workflow engine or process execution engine is a software component that "[…] allows creating, executing, and managing process instances related to the same or to different process models" (Reichert and Weber, 2012, p.33).

The introduced concept of processes execution and workflows exist since quite a while. Moreover, traditionally those workflow management systems are focussing on rigid and stable business processes. Nevertheless, the workflow concept has its legitimation but needs to be extended to support knowledge worker in a more flexible manner.

2.1.1 Flexibility of Business Processes and Workflows

Traditionally, workflow management systems are focussing "[…] predictable and repetitive business processes, which can be fully described prior to their execution in terms of formal process models" (Reichert and Weber, 2012, p.43).

However, "[…] flexibility is required to accommodate the need for evolving business processes" (Reichert and Weber, 2012, p.43). This flexibility especially required for business processes, which support knowledge workers perform their knowledge-intensive tasks. Reichert and Weber (2012) identified the following four major needs of business process flexibility: variability, looseness, adaptation, and evolution.

1. Process variability: Process variability occurs when processes need to be handled in different process variants based on the current context during process execution. "Process variants typically share the same core process whereas the concrete course of action fluctuates from variant to variant" (Reichert and Weber, 2012, p.45).

2. Process looseness: Process looseness refers to knowledge-intensive processes, which are regarded as unpredictable and emergent. "For processes of this category, only their goal is known a priori" (Reichert and Weber, 2012, p.46). An example of process looseness is patient treatment cases, where "[…] the parameters determining the exact course of action are typically not known a priori and might change during process execution" (Reichert and Weber, 2012, p.46). Therefore, a predefined and detailed process description is rather difficult and even impossible. "Instead, processes of this category require a loose specification" (Reichert and Weber, 2012, p.46).

3. Process adaptation: Process adaptation exists when entire processes or their structure need to be adapted to "special situations" or when certain "exceptions" occur (Reichert and Weber, 2012).

4. Process evolution: Process "[…] evolution represents the ability of the process implemented [in a workflow management system] to change when the corresponding business process evolves" (Reichert and Weber, 2012, p.47).

As mentioned the processes and workflows of knowledge workers tend to be unpredictable and can be assigned to the "process looseness" category. Reichert and Weber (Reichert and Weber, 2012, p.46) argue that "[…] it is not possible to establish a set of process variants for these processes, since the parameters causing differences between process instances are not known a priori" (Reichert and Weber, 2012, p.46).

Therefore, there is a need for an approach and corresponding method to enhance a workflow management system to deal with knowledge work, by providing process variants based contextual information.

In the following related concepts and methods will be introduced as a potential basis for the approach that will be presented in the research work.

2.1.2 Flexibility of Case Management

Case management provides the ability to manage cases (as the word suggests), which contains e.g. knowledge of previously experienced situations. This case knowledge can be workflow related information. Van der Aalst, Weske and Grunbauer (2005) propose case management as a paradigm shift in workflow related environments - especially for knowledge-intensive processes. Case management gives the workers more freedom, flexibility and provides the awareness of the whole context of activities within a business process (van der Aalst, Weske and Grunbauer, 2005). McCauley (2010, p.265) defines case management as follows "Case management is the management of long-lived collaborative processes that require coordination of knowledge, content, correspondence and resources to achieve an objective or goal. […] Human judgement is required in determining how to proceed, and the state of the case can be affected by external events."

Case management focuses on the whole case. Whereas in workflow management, the focus is on the current work item or activity the execution. Workflow management makes only a small contribution towards accomplishing the goal of the entire case. Case handling is driven by the data-flow instead of the control-flow, and this is also true for knowledge-intensive processes where the process is based on a collection of data objects (van der Aalst, Weske and Grunbauer, 2005). This data-flow focus means that the data objects representing the whole context are the key part in knowledge-intensive processes and case management.

Swenson (2010) has taken up the topic of case management and introduced a "new process-management orientation […] as adaptive case management (ACM) [, where] the case itself is the focus" (Swenson, 2010, p.2). Swenson (2010) defines ACM as "[…] systems that are able to support decision making and data capture while providing the freedom for knowledge workers to apply their understanding and subject matter expertise to respond to unique or changing circumstances within the business environment" (Swenson, 2010, p.4). The Workflow Management Coalition (WfMC) has also taken up the ACM topic, in the meantime several books (Swenson, Palmer and Silver, 2011; Swenson, Palmer, Pucher and MD, 2012) were published in association with the WfMC concerning ACM. The adaptability feature of ACM has some similarities with the case-based reasoning methodology. ACM includes mechanisms to reuse "[…] templates for initiating new cases, including the use of completed cases as templates. […] So the case itself can be a template for a new case instance" (Palmer, 2011, p.85).

2.1.3 Flexibility based on Experiential Knowledge

In knowledge-intensive environments, people learn from their experience. However, the own field of activity is always limited. Therefore people try to learn from others as well. Especially the so called procedural (Schumacher, Minor, Walter and Bergmann, 2012) or "how-to" knowledge (Plaza, 2009) as part of the experiential knowledge of people is a central aspect in knowledge intensive processes.

It is common in knowledge intensive environments that the people's procedural or "how-to" knowledge is in their mind or written down in forms or documents (Schumacher et al., 2012; Plaza, 2009). Recent work of (Madhusudan, Zhao and Marshall (2004) and Schumacher et al. (2012) has shown that it is possible and worthwhile to capture procedural and process knowledge as cases. According to Bergmann (2002), experiential knowledge can be divided into the three different categories vocabulary, experience base, and reuse-related knowledge.

CBR definition according to Kolodner (1993, p.13) "a case is contextualised piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoner". Kolodner (1993) points out the context awareness of cases – the context of the cases is described either explicit or implicitly.

2.2 Case-Based Reasoning

Case-based reasoning (CBR) can be seen as Leake (1996, p.2) defines "reasoning by remembering" (Leake, 1996, p.2) or "reasoning from reminding" (Madhusudan, Zhao and Marshall, 2004) and as a technical independent methodology (Watson, 1999) to humans and information systems. *"Case-based reasoning is both […] the ways people use cases to solve problems and the ways we can make machines use them"* (Kolodner, 1993, p.27). CBR can be seen according to Aamodt and Plaza (1994, p.1) as "[…]a recent approach to problem-solving and learning". As Bergmann et al. (2009) defines case-based reasoning is a sub-field of artificial intelligence. The roots of CBR are cognitive science, machine learning and knowledge-based systems (Bergmann et al., 2009).

Cognitive science strongly influences Case-based reasoning – the original idea was derived from the results of several studies concerning the human brain (Lopez de Mantaras et al., 2006). Schank and Abelson (1977, p.36 ff.) laid a foundation for further case-based reasoning studies by their studies of how humans understand stories and how the memory affects the understanding of particular stories. Schank (1982, p.83) introduced the concept of memory organisation packets MOPs. These MOPs try to explain how humans organise individual scenes in living that are linked to other MOPs and can be linked to a specific context or major goal. Schank (1999, p.123) came up with a revised definition of MOPs as follows: *"A MOP consists of a set of scenes directed toward the achievement of a goal. A MOP always has one major scene whose goal is the essence or purpose of the events organized by the MOP".* To use these MOPs as a source for reminding and adapting the memories to a new situation, Schank (1999, p.137) argued that *"[…] there must be structures that capture similarities between situations that occur in different domains"*. These structures are introduced by Schank (1999, p.137) as thematic organisation packets TOPs and contain abstract and domain-independent information. MOPs are a basis for creating cases in a case-based reasoning approach (Lopez de Mantaras et al., 2006) – a person who creates cases in a case-based reasoning system tries to describe individual (set) scenes (MOPs). A person us often able to use personal reminding's (MOPs) in a different situation by abstracting existing information using TOPs – this as a relationship to case-based reasoning and a basic requirement to the CBR users and the approach as well.

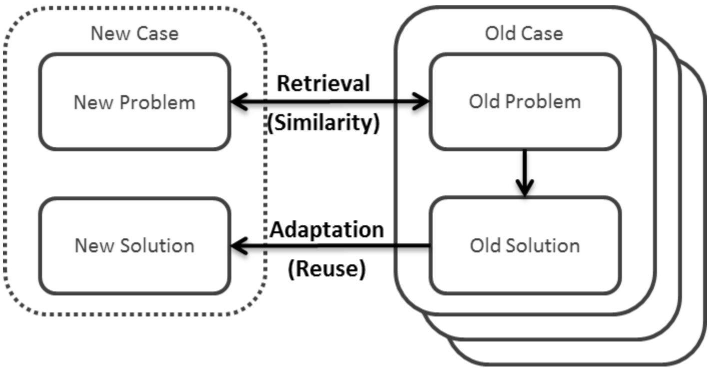

Figure 2.1: Case-based Reasoning Reuse Principle (adapted from Richter and Weber (2013))

Figure 2.1 depicts the basic reuse principle of case-based reasoning. Case-based reasoning (CBR) uses the specific knowledge of previously experienced situations (cases containing old problems and old solutions) to propose a solution (reuse) to a new situation (new problem). This suggestion is made by comparing the new problem with old problems based on the similarity (Aamodt and Plaza, 1994; Richter and Weber, 2013). This principle refers to the underlying assumption of CBR that, "[…] similar problems have similar solutions" (Watson, 2003, p.31). However, this also refers to one of the main challenges in CBR: "If similar problems have very different solutions, a case-based reasoner may give inaccurate advice" (Watson, 2003, p.31). Using the traditional CBR terminology a case consists of a *problem space* (problem items/descriptions) that is used for describing a certain *solution space* (solution items) (Bergmann, 2002). “A case-based reasoner solves new problems by adapting solutions that were used to solve old problems” (Riesbeck and Schank, 1989, p.25). Apart from the reuse of these historical cases, case-based reasoning also provides the “learning of new cases”. Based on Aamodt and Plaza (1994) and Madhusudan, Zhao and Marshall (2004) the generic CBR cycle (see Figure 2.2) consists of the following steps: *1.* ***Retrieve*** *the most similar cases from the knowledge base (case-base containing previous cases) based on the problem description of the new case (problem case) using a* *similarity* *mechanism. 2.* ***Reuse*** *the knowledge in the retrieved case(s) to solve the current problem – adapt the historical knowledge to the new problem (adaptation). 3.* ***Revise*** *and test the suggested solution e.g. by evaluating it under the real-world problem (evaluation). 4.* ***Retain*** *useful experience (past solutions and failures) for future reuse and store a new case in the knowledge base (case learning).*

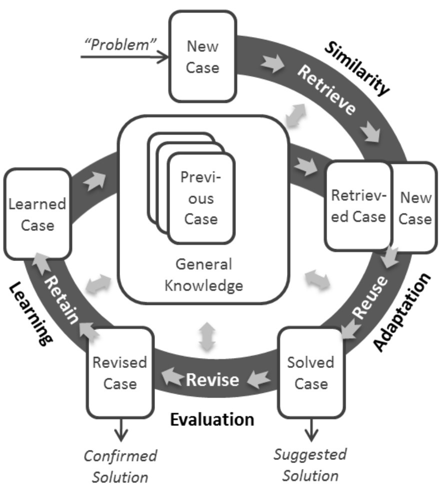

Figure 2.2: The CBR Cycle (adapted from Aamodt and Plaza (1994))

2.2.1 General Case Structure

The basic idea of the 'case' concept is to capture information for problem-solving as used in cognitive science (Bergmann, Kolodner and Plaza, 2005). Traditionally a CBR systems case description consists of a *problem* and a *solution* part. Bergmann (2002) extends the problem and solution view by the *characterisation part* and the *lesson part*. “The case characterization part describes all facts about the experience that are relevant for deciding whether the experience can be reused in a certain situation” (Bergmann, 2002, p.48). The characterisation part contains elements that can be seen as index or metadata to the case. In contrast to regular index or metadata, the characterisation part is usually more detailed as it must contain the whole context of a case. In CBR, the characterisation part can be seen as problem space. Bergmann (2002, p.50) extends this view to the characterisation part that can contain “[…] derived descriptions or properties that were not present in the problem-solving situations […]”. The lesson part contains elements that are needed to describe the case to enable the *user to take actions* based on the suggestion. Based on Secchi, Ciaschi and Spence (1999) several space agencies defines a lesson or lesson learned as “[…] knowledge or understanding gained by experience” (Weber, Aha and Becerra-Fernandez, 2000, p.63). In traditional CBR terminology, the lesson part can be seen as solution part. Bergmann (2002, p.50) extends this view by using the term lesson space that “[…] can contain information that is not the solution itself but useful to find a solution”.

2.2.1.1 A Case in CBR

Based on Bergmann, Kolodner and Plaza (2005, p.209), a case “[…] in case-based reasoning is a contextualized piece of experience”. They also suggest that the form of the case should not be specified in advance – therefore a general definition is presented here:

Definition 2.5: Case in Case-based Reasoning

A ***case*** in case-based reasoning consists of at least one ***experience/knowledge item*** that is contextualised (adapted from Bergmann, Kolodner and Plaza (2005)).

Definition 2.5 explains the meaning and the purpose of cases in Case-based Reasoning – a case tries to capture experience or fragments of it as the basis for future problems.

2.2.1.2 Elements of a Case in CBR

As mentioned at the beginning of this section, the traditional view of problem and solution should be extended. Based on Kolodner's (1993) work, Bergmann, Kolodner and Plaza (2005, p.209) describe the following possible five elements of a case structure: *"(i) a situation and its goal; (ii) the solution and, sometimes, means of deriving it; (iii) the result of carrying it out; (iv) explanations of results; and (v) lessons that can be learned from the experience."* The following definition lists possible case elements, which should be described in cases of a case-based reasoning system.

Definition 2.6: Case Elements

A ***case*** in case-based reasoning should contain the following ***elements***: (1) the ***situation*** including goals (*problem*); (2) the ***approach*** (*solution*) and ***derivations*** of the approach; (3) the ***result*** (or *outcome*) including ***explanations***; and (4) the ***lesson learned*** (adapted from Bergmann, Kolodner and Plaza (2005)).

2.2.1.3 CBR Knowledge Containers

The distribution of knowledge in a case-based reasoning system has been introduced by Richter (1995). In case-based reasoning, it is possible to identify four containers of CBR knowledge (Richter, 1998). These "knowledge container" are identified as shown in Figure 2.3: the vocabulary, similarity measure, solution transformation and case base (Richter, 1998).

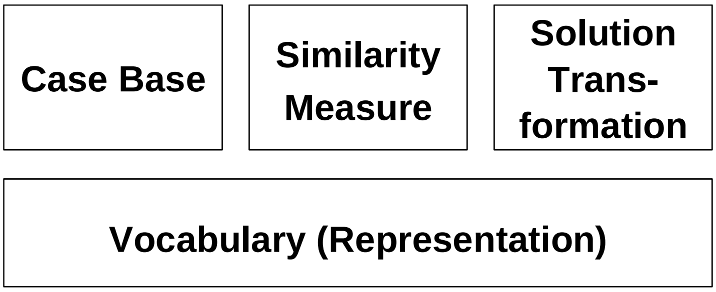

Figure 2.3: CBR Knowledge Containers (Bergmann and Schaaf, 2003)

The *vocabulary* container contains the *background knowledge*, which is "general and problem independent […]" and when it "[…] describes a specific part of the domain it is also called *contextual knowledge*" (Richter, 1998, p.4).

Definition 2.7: Case Vocabulary

The ***case vocabulary*** is used to contextualise a ***case*** in case-based reasoning (adapted from Richter (1998)).

The cases itself are captured in a *case base*, which is another knowledge container in case-based reasoning (Richter, 1998). A similarity measurement is needed to retrieve cases in a CBR system. "A case base CB is a set of cases which, for retrieval purposes, is usually equipped with additional structures; structured case bases also exist under the name of case memory" (Richter, 1998, p.6).

Definition 2.8: Case Base

A ***case base*** in case-based reasoning is a ***structured set of cases*** (adapted from Richter (1998)).

Bergmann and Schaaf (2003) described the knowledge containers of a structural case-based reasoning system (see Section 2.2.2.3 Structural CBR). In Figure 2.4 the elements are presented, which are relevant when describing (characterising) knowledge items (Bergmann and Schaaf, 2003).

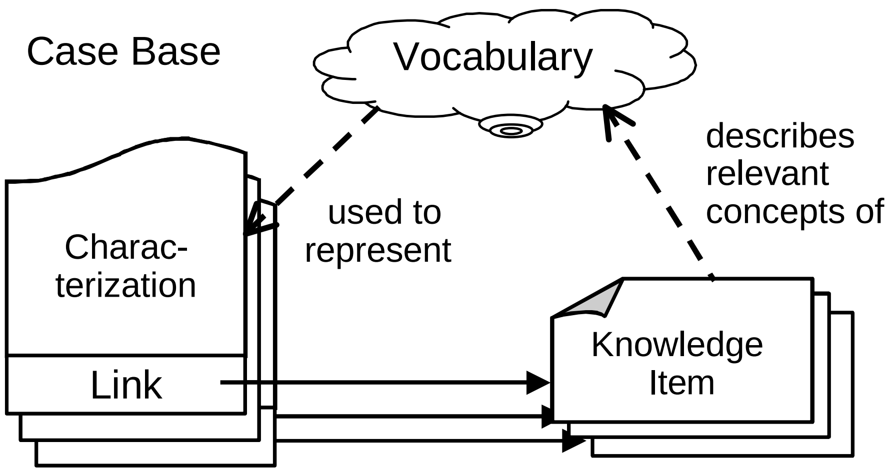

Figure 2.4: CBR Knowledge Containers (Bergmann and Schaaf, 2003)

In this traditional terminology, the term knowledge item is used, which can be seen as experience item as stipulated in Definition 2.5. The knowledge item is linked to a case characterisation, which describes to knowledge item using the vocabulary knowledge container. As mentioned before, this vocabulary consists of concepts for describing the knowledge items (Bergmann and Schaaf, 2003).

Definition 2.9: Case Characterisation

In extension to Definition 2.5, the ***experience/knowledge items*** that are linked to cases are abstracted using a ***characterisation*** (adapted from Bergmann and Schaaf (2003)). The case characterisation "[…] describes all facts about the experience that are relevant for deciding whether the experience can be reused in a certain situation" (Bergmann, 2002, p.50).

As mentioned a *similarity measure* is needed in CBR to retrieve similar cases. According to Richter (1998), this similarity measure is the another knowledge container in CBR. "A similarity measure is a container which can store more or less sophisticated knowledge about a problem class" (Richter, 1998, p.8).

The *solution transformation* knowledge container contains knowledge that is used during adaptation of a retrieved experience item to a new situation. This solution transformation is sometimes called adaptation knowledge (Richter, 1995; Wilke, Vollrath, Altho and Bergmann, 1997; Richter, 1998).

2.2.1.3.1 CBR Characterisation and Metadata

The term metadata is often used to describe the relationship of data that describes other data. Sometimes then, the question arises whether metadata and characterisation can be used interchangeably. "Metadata is often called data about data or information about information. […] Descriptive metadata describes a resource for purposes such as discovery and identification" (NISO, 2004, p.1). This definition of the National Information Standards Organization (NISO) is indeed a very general definition of metadata. Additionally, when using the definition of Greenberg (2005) it possible to come to a conclusion that both terms can be used interchangeably. "The term metadata […] addresses data attributes that describe, provide context, indicate the quality, or document another object (or data) characteristics" (Greenberg, 2005, p.20). Bergmann and Schaaf (2003) investigated the relation of structural CBR and Ontology-Based Knowledge Management and came to the conclusion that experience items (knowledge items) are both "[…] abstracted to a characterization by metadata descriptions, which are used for further processing" (Bergmann and Schaaf, 2003, p.609). Based on that investigation, it is possible to use metadata and characterisation interchangeably in structural CBR.

2.2.2 CBR Approaches

It is possible to distinguish three main approaches in case-based reasoning (Bergmann et al., 2003): 1. *Textual CBR* (see e.g. Shimazu (1998) and Weber, Ashley and Bruninghaus (2005)) where the cases are recorded as or derived from the free text. 2. *Conversational CBR* (see e.g. D. Aha and Breslow (1997), D. Aha, Maney and Breslow (1998) and Aha, Breslow, and Muñoz-Avila (2001)) where the case acquisition takes place in a conversational (dialogue) manner. 3. *Structural CBR* (see e.g. Yokoyama (1990) and Aamodt (1991)) where the cases are described by using a certain vocabulary or domain model (Bergmann, 2002).

2.2.2.1 Textual CBR

The textual case representation approach is free text based (technically free texts are usually strings). According to Bergmann (2002), textual case approaches are appropriate for huge collections of existing free text documents and the reader of the cases can reuse the information in the cases without any adaptation or further processing. Text based case approaches use free text, which is more cognitively adequate to humans it is possible to deduce, therefore, that more textual documents already exists and to is faster to acquire cases from existing documents Bergmann (2002). Because no or only a simple pre- processing steps are needed to acquire initial cases from existing documents compared to other approaches that will be explained in the following. To retrieve textual cases the *keyword matching* approach is often used (Bergmann, 2002). However, keyword matching approaches have a core issue – according to Bergmann (2002, p.54) textual retrieval methods "[…] are restricted to pure syntactic retrieval […]" and "[…] are mostly unable to capture the semantics of the text". Bergmann (2002) mentions frequently asked questions documents as a prominent example for textual cases where each case contains a problem represented as question and solution part, which is usually written text. It is necessary for textual approaches that the characterisation part is well described using distinguishable keywords. According to Bergmann's (2002) conclusion, textual case approaches are sufficient with only a few cases containing precisely and *discriminating* keywords.

2.2.2.2 Conversational CBR

The idea behind the conversation case representation is to capture knowledge that is gained during a conversation, similar to a person-system or person-person dialogue. In contrast to other approaches conversation based systems do not use a domain model or structure (Bergmann, 2002). An example of such a conversational-based approach is a call-centre situation, where an operator asks several questions to the client to localise the problem and provide an appropriate solution. The conversation contains several questions, which will be selected based on the course of the conversation. These questions can be answered by saying yes or now; or by selecting suggested options or by giving detailed information such as product types. The cases have a usually a "decision-tree-like structure" and the CBR system provides the user "predefined dialogues" (Bergmann, 2002). According to Bergmann (2002, p.55), the conversation approach "[…] is very useful for domains where a high volume of simple problems must be solved again and again".

2.2.2.3 Structural CBR

The structural case representation uses a *vocabulary* to restrict the possible case elements. The cases are usually represented technically as *attribute-value pairs* (in flat tables or object oriented classes), as *graphs* or it can be represented as *formulae* that contain variables as in predicate logic (Bergmann, 2002). The definition of the cases itself can be seen as domain model (Bergmann, 2002). According to Bergmann (2002, p.56) this "[…] domain model ensures that new cases are of high quality, and the maintenance effort is low". Compared to the textual and conversational case approaches the structural case approach produces the best results but on the other hand, this approach needs to biggest initial effort to create the domain model (Bergmann, 2002).

2.2.2.4 Comparison of CBR Approaches

Bergmann (2002) made an analysis to compare the difference approaches deeply. He also shares the experience of creating a case base in a company including the difficulties they might occur. Further, he points out three main requirements and efforts that should be taken into account when implementing a case-based reasoning system: Frist the *initial material* that is required to set up the case base. Second, the *effort to maintain* the case base. Moreover, third the *effort to control the accuracy* of the case-based reasoning system.

When implementing a case-based reasoning system, there usually exists no ideal situation in a company, where the needed material is available and is in that structure that is required. Most of the time a pre- processing step is needed – here the effort differs from one approach to another.

Based on the assessment of Bergmann (2002) it is possible to conclude that the structural approach is appropriate for *complex problem solving* in comparison to the textual and conversational approaches. The structural approach enables a more accurate retrieval as the textual approach, and the maintenance effort is lower as it would need the conversational approach when scaling the case base up (Bergmann, 2002).

2.2.3 Structural CBR Approaches

There exist several ways to represent cases in a structural manner. Bergmann (2002) listed four main representation approaches for cases. In the following, the four main approaches are introduced.

2.2.3.1 Attribute-value Representation

The attribute-value representation of cases is a relationship between an attribute and value that is restricted to a particular type, which defines the value range of the attribute. A case consists of at least one attribute-value pair. The assignment of the attribute-value pairs to the case can be fixed or differ from case to another (Bergmann, 2002).

2.2.3.2 Object-oriented Case Representation

According to Bergmann (2002), the object-oriented case representation can be seen as an extension to the attribute-value representation of cases. It extends the attribute-value pair approach with the object oriented "*is-a*" and "*part-of*" relations including the *inheritance* possibility. The cases are represented as objects that were instantiated from classes. These classes contain attributes, which also realise the "part-of" relation between objects. The "is-a" relation is realised by the inheritance principle of the object-oriented approach. Bergmann (2002) suggests the usage of an object-oriented case representation when cases with mixed structures need to be represented.

Plaza (1995) introduced a *feature-term* (sometimes called *feature structure*) representation that enables to represent cases in a structured (sometimes called *object-cantered)* way. This approach uses *anti-unification* and *subsumption* to evaluate the similarity of cases.

2.2.3.3 Graph Representation

The graph representation can be seen as (attributed) directed graph and also be seen as a tree structure (Bergmann, 2002). The attributed directed graph is represented as triple. The attributed graph is used as the case description, and the attribution of the graphs (node and edge descriptors) are restricted into a certain vocabulary based on the domain model.

2.2.3.4 Predicate Logic Representation

According to Bergmann (2002) predicate logic case representation is useful for planning experience and diagnosis related applications. For experience management, Bergmann (2002, p.69) argues that predicate logic would "[…] only play a minor role […] except for planning experience". In predicate logic case representation (vocabulary) consists of functions and predicate symbols. Instead of having several sets of ground formulas (function and predicate symbol), it is possible to use a structured subset of formulas. This structured subset gives the possibility to represent characterisation and lesson using one set. Apart from that with a predicate logic based representation, it is possible to represent arbitrary structures (Bergmann, 2002).

2.2.4 Case-Based Reasoning Processes

The methods for supporting the CBR process highly depend on the actual case representation approaches as presented in the sections before.

2.2.4.1 CBR Similarity (Retrieve)

Case-based reasoning uses similarity measures to retrieve similar cases. Most of the measures come from the research field of information retrieval. According to Lopez de Mantaras et al. (2006), it is possible to distinguish case-based reasoning applications using *surface similarity* and others using *structure similarity* also. Surface similarity algorithms use the surface features of a case. “The surface features of a case are those that are provided as part of its description and are typically represented using attribute-value pairs” (Lopez de Mantaras et al., 2006, p.219). According to Richter (1998, p.8): “Surface similarity considers only syntactic properties of the representation”. Cunningham (2009) did an extensive state of the art work concerning CBR measurements and proposed taxonomy. He stated feature-vector representation as the standard CBR methodology: “The standard methodology in CBR is to represent a case as a feature vector and then to assess similarity based on this feature vector representation” (Cunningham, 2009, p.1).

As mentioned before and in Mantaras et al. (2006), the most common approach in CBR is the feature vector based representation. In feature vector based representation, the surface similarity assessment is usually done using a local similarity and a global similarity method.

The *local similarity* measurement is defined for every feature as sketched in Figure 2.5. According to Richter (1998, p.8): “Local similarity deals with the values of a single attribute or feature”. For example, let us assume that there is needed to distinguish patient medical files. The task would be to retrieve a historical case of a specific disease using the International Classification of Diseases (ICD) identifier (represented as text). It is possible to use an equal function/measurement to assess the similarity between a case of given situation and the historical cases stored in the case base, which indicates if ICD[5] identifier is identical (equals) or not using the values 0 or 1. As a result, such a CBR system would list all historical cases where the specific disease identifier is assigned as value.

The *global similarity* (sometimes called object similarity) measurement is used to combine several local similarity values and calculate an overall similarity value. This global similarity calculation gives the opportunity to evaluate the similarity of cases with more than one feature. According to Richter (1998, p.8), global similarity "[…] represents a holistic view of the cases". Richter (1998, p.8) described that global similarity could be "derived" from the local similarity measurements. "The relative importance of attributes can be reflected by weights but Additionally, the relative position in a hierarchy, as well as general background knowledge, can be incorporated" (Richter, 1998, p.8). As it shown in Figure 2.5, the global similarity (object similarity) is calculated over different local similarity features. The weights are can pre-defined by domain experts or end-users, or learned by an adaptive learning algorithm (Lopez de Mantaras et al., 2006).

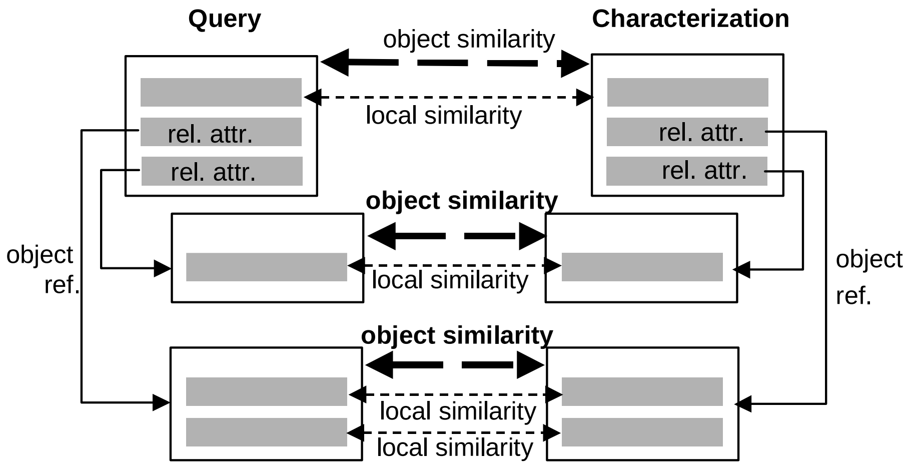

Figure 2.5: Sketch of the Similarity Computation (Bergmann and Schaaf, 2003)

[5] http://www.who.int/classifications/icd/en/

In Figure 2.5 it is possible to recognise certain (object) references to another global similarity (object similarity) calculation. This referencing to other global similarity elements represents a certain structure of the case characterisation, which is called *structural similarity*. According to Richter (1998, p.8), in CBR the term structural similarity has two distinct meanings: "On the one hand, it means to consider structural aspects of the problems or cases compared. On the contrary, it can also mean to consider the similarity of a whole set of cases".

The goal is the "conceptualization of noisy cases" and build a "decision surface" (Cunningham, 2009) when defining certain features in case-based reasoning. Vector space models and the application of "direct similarity" (Cunningham, 2009) methods is only one path to identify the similarities over several cases. There exists a vast number of approaches for calculating similarities in case-based reasoning (Lopez de Mantaras et al., 2006).

Apart from feature-value based, hierarchical structure and graph-based approaches, there exists other possibilities like transformation-based approaches for string matching, further information theoretic, ontology, taxonomy or machine learning approaches (Cunningham, 2009) – see Section 2.2.5 (Multi-Method CBR Approaches).

2.2.4.2 CBR Adaptation (Reuse)

The adaptation of historical knowledge to a new situation is an important task in case-based reasoning. Alternatively, in the words of Hanney and Keane (1997, p.359): "[…] *the success of a CBR system often critically depends on its ability to adapt the solution of a previous case to suit a new situation*". According to Lopez de Mantaras et al. (2006), it is possible to identify two different dimensions of adaptation: "what is changed in the retrieved solution and how the change is achieved". Additionally, Lopez de Mantaras et al. (2006) identified the following three different adaptation methods: *Substitution*, *transformation*, and *generative adaptation*.

When executing substitution adaptation, the experience items of a retrieved case will be manually or automatically re-instantiated to serve a new similar situation (Lopez de Mantaras et al., 2006).

Definition 2.10: Substitution adaptation
Substitution adaptation *re-instantiates* the selected experience items of the retrieved case (based on Kolodner (1993) and adapted from Lopez de Mantaras et al. (2006, p.227)).

In contrast to substitution adaptation, transformation adaptation changes the structure of a potential solution using a certain transformation method (Lopez de Mantaras et al., 2006).

Definition 2.11: Transformation adaptation
Transformation adaptation *restructures* the solution and experience items (based on Kolodner (1993) and adapted from Lopez de Mantaras et al. (2006, p.227)).

Generative adaptation does not change the retrieved case directly – it replays the procedure of a recent transformation. The actual modification of the potential solution is usually done using substitution or transformation adaptation (Lopez de Mantaras et al., 2006).

Definition 2.12: Generative adaptation
Generative adaptation *replays the procedure (method)* of adapting the retrieved case to the new situation (adapted from Lopez de Mantaras et al. (2006, pp.227 & 228)).

Such procedures in generative adaptation can be learned from the case base using already acquired knowledge and imported knowledge such as domain knowledge. One example of such a generative adaptation approach using additional knowledge is introduced in the following section.

2.2.4.2.1 Examples of CBR Adaptation

Hanney and Keane (1997) argue that many systems use hand-crafted rules for the adaptation task and identify adaptation algorithms based on *feature differences* within cases (situation vs. lessons) as popular (Hanney and Keane, 1997). The problem is that a knowledge engineer needs to understand the application domain and needs to able to create hand-crafted rules based on this feature-differences, apart from that the engineer has to predict the feature differences that might occur in future (Hanney and Keane, 1997). As Hanney and Keane (1997) propose, the learning of adaptation knowledge/rules would tackle this issue. They introduced a rule learning system based on feature differences within an existing case base. Wilke, Vollrath, Altho and Bergmann (1997) went into the same direction. They introduced a framework for learning adaptation knowledge in case-based reasoning. Wilke, Vollrath, Altho and Bergmann (1997, p.235) also argue that the "*modelling of appropriate adaptation knowledge*" is the main challenge in case-based reasoning – "*adaptation knowledge is in contrast to cases, not quickly available and hard to acquire*". The acquisition of the adaptation knowledge will be made using a "knowledge light" approach. The term "knowledge light" refers to the knowledge that is already captured inside a case-based reasoning system, which can be used to derive adaptation knowledge.

Definition 2.13: Knowledge Light Learning Algorithm
A **knowledge light learning algorithm** refers to algorithms, which "[…] *use already acquired knowledge inside the system to learn* […]" (Wilke et al., 1997, p.236) **rules** or **values of parameters** for a certain task.

The proposed framework executes two main steps to extract adaptation knowledge out of the case base.

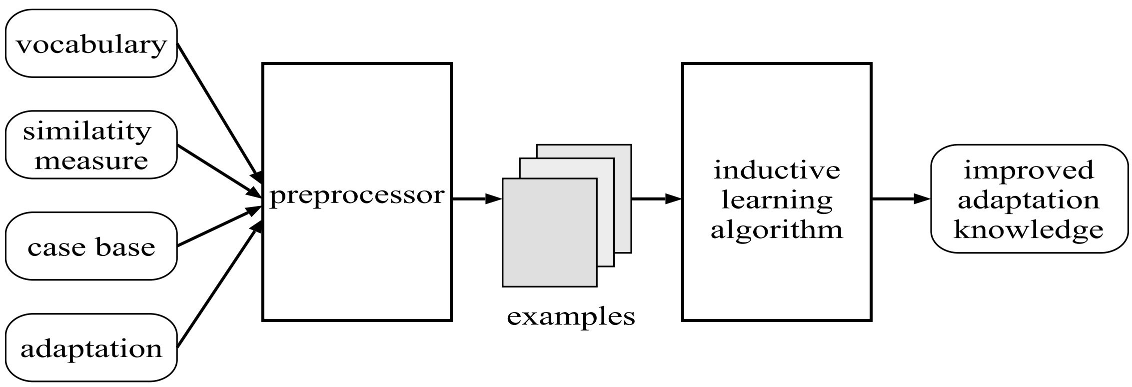

Figure 2.6: Elaboration adaptation knowledge (Wilke et al., 1997)

Frist one, is a *pre-processing step* where a set of example will be selected. As a second step, an *inductive learning algorithm* will be applied. The algorithm uses the example attributes, which were extracted in the pre-processing step. Wilke, Vollrath, Altho and Bergmann (1997) showed in their work how this approach could be successfully applied in rules-based, and optimisation-based adaptation knowledge approaches based on a case study. Wilke, Vollrath, Altho and Bergmann (1997, p.240) came up with general issues when implementing Knowledge Light Learning Algorithms. The first problem is the *adequacy* of the learning algorithm and regards whether an algorithm can solve a learning task – the availability of data and the learning task itself have a major impact on the adequacy. The second issue is the "*pre-processing of the knowledge*", which deals mainly with the selection, quality measurement and construction of the learning data. The third issue is the "*integration of the learned knowledge*" into an existing and already applied learning knowledge.

D'Aquin, Lieber and Napoli (2006) address in their work the issue of analogical reasoning in case-based reasoning by extending OWL to represent adaptation knowledge. Apart from that, d'Aquin, Lieber and Napoli (2006) presented a way to use OWL retrieval for retrieval and adaptation. One issue was the acquisition adaptation knowledge. As a consequence D'Aquin et al. (2007) introduced later a system for the elaboration of adaptation knowledge using case-based mining techniques - this system is used in a real world scenario to support physicians in breast cancer treatment. The system applies algorithms from the data-mining domain and is using a light extension of OWL DL for the case base and the domain knowledge.

2.2.4.3 CBR Evaluation (Revise)

The revision step of the CBR-cycle is often tightly connected to reuse step in case-based reasoning. Several research papers describe these two steps in an interactive manner. For example, Lopez de Mantaras et al. (2006, p.227) describe a possible and common interaction between adaptation and evaluation as follows: “Adaptation can also be used when feedback about a proposed solution indicates that a repair is needed; this is part of the revise stage in the CBR cycle”. The explanation of Althoff (1997, p.177) goes into the direction that it becomes clear that there is a tight information flow of the earlier steps: “The solutions found by, the earlier tasks and the knowledge structures may be controlled and confirmed or repaired if they do not work as expected”. It is possible to see the revision step as “[…] a preparing process for knowledge acquisition” (Althoff, 1997, p.177).

Aamodt and Plaza (1994) identified two main tasks in the revision step: evaluation and reparation. “The evaluation task takes the result from applying the solution in the real environment” (Aamodt and Plaza, 1994, p.17). The implication is that the evaluation task takes often place outside the case-based reasoning system. In the evaluation task, Aamodt and Plaza (1994) mentioned three techniques: (1) evaluate by a teacher, (2) evaluate in reality and (3) evaluate using a model. The evaluation by a teacher can also be seen as evaluation by an expert, and it is crucial that there be an expert available (Althoff, 1997). When executing an evaluation in a real world, the system can guide the process in a way that the system notifies that there are environmental changes (Althoff, 1997). A model based evaluation can happen in two ways. The first is an identification of “inconsistencies in symbolic models”, and the second option is an evaluation using “numerical simulation” (Althoff, 1997, p.187). The reparation task consists of two main techniques (Aamodt and Plaza, 1994): (1) self-reparation and (2) user-reparation. When execution the self-reparation technique, the system tries to repair the solution by itself. This self-reparation can be done using domain knowledge, or it is based on simulation results, or it is integrated into the adaptation step (Althoff, 1997). Finally, the user-reparation is done, as the name suggests, by the user himself and this is maybe also part of the adaptation step as well (Althoff, 1997).

2.2.4.4 CBR Learning (Retain)

According to Richter (1998), the retaining step is where the "lesson learning" is performed, and it is the final step of the original step in the CBR cycle of Aamodt and Plaza (1994). "The task of learning (we use machine learning as a synonym) is to improve a certain performance using some experience or instructions" (Richter, 1998, p.9). Learning in CBR can be performed in different ways. Starting with a simple storage of all fields of a revised case to sophisticated machine learning algorithms and case-based reasoning maintenance approaches.

Most systems simply store the solution of a case in their case base with the assumption that the solution was helpful, others add further information concerning the problem and other fine-grained information about the solution finding process (Lopez de Mantaras et al., 2006).

2.2.4.4.1 The Utility Problem

One prominent issue in case-based reasoning and its retention phase is the utility problem. "The utility problem in learning systems occurred when knowledge learned in an attempt to improve a system's performance degrades it instead" (Houeland and Aamodt, 2010, p.142). This utility problem often counts for so-called speed-up learners, "[…] where the system's knowledge is used to reduce the amount of reasoning required to solve a problem" (Houeland and Aamodt, 2010, p.142). Some case-based reasoning systems can be seen as speed-up learners. According to Houeland and Aamodt (2010, p.142) "[…] cases in a CBR system may be viewed as a form of speed-up knowledge, where storing, retrieving, and adapting cases provides for more efficient problem solving than first-principles or model-based methods". The goal of case-based reasoning and speed-up systems is to create a potential solution to a specific problem as fast as possible under the assumption that a slower retrieval method already exists. Not all case based reasoners count as speed-up learning systems – especially those systems, which are using the case base as the only source for problem knowledge

(Lopez de Mantaras et al., 2006). Houeland and Aamodt (2010) calls case-based reasoning as a lazy learning approach. Lazy learning is "[…] that choices regarding the solution of a problem will be postponed as long as possible until the problem query is posed and as much information as possible is available" (Houeland and Aamodt, 2010, p.142). The following two main approaches can be used to avoid the utility problem for lazy learners: first, an efficient indexing and second the deletion of cases in retaining phase. As Houeland and Aamodt (2010) argue, these two approaches are against the original idea of case-based reasoning because this is somehow a "knowledge reduction" without knowing if a deleted case might have been useful in future or not. Therefore, simple indexing and case deletion should be avoided in case-based reasoning. Houeland and Aamodt (2010, p.154) have shown that "many practical CBR systems do not require the use of these eager optimizations and can be limited by committing to decisions prematurely". Additionally, Smyth and Cunningham (1996) showed that additional cases could reduce the efficiency of a case-based reasoning system, but it increases the quality to a certain point. They also showed that the retrieval efficiency decreases where at the same time the adaptation efficiency increases.

2.2.4.4.2 Problem-solving Competence

Despite the fact that case base enlargement has its positive side (as mentioned in section before), the case-based reasoning community tries to use other artificial intelligence and machine learning approaches to deal with harmful knowledge and the utility problem in particular (Lopez de Mantaras et al., 2006). One prominent technique used as *coping strategy* is the *nearest neighbour classification* to identify certain patterns - e.g. the condensed nearest neighbour rule (CNN rule) approach by Hart (1968) for removing harmful examples from training data (Lopez de Mantaras et al., 2006; Smyth and McKenna, 1999).

As mentioned in the former section, case deletion is an important technique when dealing with harmful cases. Case deletion is one technique of the *information* *filtering model* called *selective retention* introduced by Markovitch and Scott (1993). The model is a framework for information filtering to avoid harmful knowledge in learning systems. The deletion of cases in case-based reasoning is related to selective retention filtering model. Based on this, Smyth and Keane (1995) introduced a *competence model* for case deletion in case-based reasoning systems. These competence model asses every case how it affects the problem-solving competence the case based reasoning system using certain criteria. The model categorises the cases into four main classes: pivotal, spanning, supporting, auxiliary. *Pivotal cases* should not be deleted, because it would result in a reduction of competence (knowledge loss). In contrary, *auxiliary cases* do not influence the competence level of a case based reasoning system – a deletion would only reduce the efficiency of the system. *Spanning cases* act as links between cases – they do not directly influence to competence level. However, they might become relevant if linked cases will be deleted. Finally, *supporting cases* are a special form of spanning cases – they often exist as a group. This means that each case within a group is somehow linked and related. The deletion of some cases of such a group would not affect the competence level – a deletion of the whole group would reduce the competence level of a case-based reasoning system. Smyth and McKenna (1999) evaluated such a competence model with the result that such an approach works well compared to traditional techniques. Nevertheless, Leake and Wilson (2000) showed that besides competence also compactness and adaptation performance matters too.

2.2.4.4.3 Case-base Maintenance

Based on the former section it becomes clear that researchers began to realise that there is a need for maintenance processes and policies. "Case-base maintenance implements policies for revising the organization or contents (representation, domain content, accounting information, or implementation) of the case-base to facilitate future reasoning for a particular set of performance objectives" (Leake and Wilson, 1998, p.197). Nowadays, case-based maintenance is recognised as an essential element it the retaining step of case-based reasoning (Lopez de Mantaras et al., 2006).

Leake and Wilson (1998) introduced a framework for characterising case-base maintenance systems based on certain *policies*. Later Wilson and Leake (2001) improved this earlier work and applied the refined dimensions to all knowledge containers – the result is a framework which aims to understand the "general problem" of case-based reasoning maintenance. As an example, one policy category is called *data collection* – "this is the information that will be used to determine whether maintenance operations should be performed" (Wilson and Leake, 2001, p.199). Such case-base maintenance policies are an important element today's case-based reasoning approaches. The work of Wilson and Leake indicates the importance of case-based maintenance. As a consequence of this direction of case-based reasoning research and the growing importance Reinartz, Iglezakis and Roth–Berghofer (2001) proposed an extension of the original case based reasoning cycle of Aamodt and Plaza (1994).

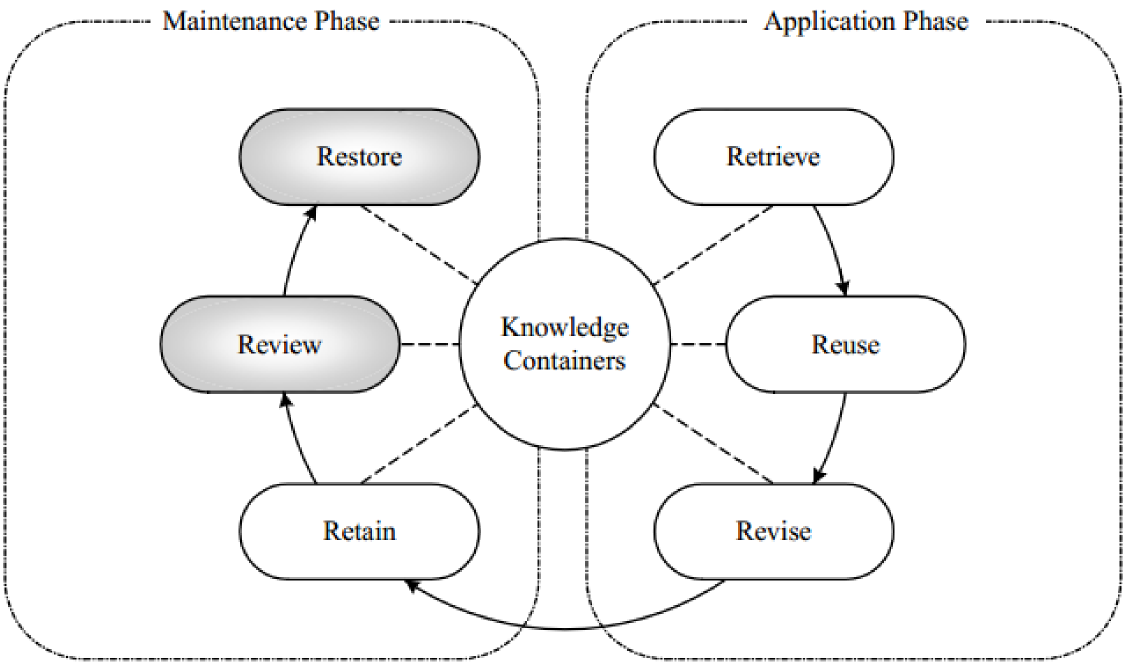

Figure 2.7: The Six-Step CBR Cycle (Reinartz, Iglezakis and Roth–Berghofer, 2001)

As shown in Figure 2.7, Reinartz, Iglezakis and Roth–Berghofer (2001) extended the case based reasoning cycle with a *review* and *restore* step. "The review step considers the current state of the knowledge container and assesses its quality" (Reinartz, Iglezakis and Roth–Berghofer, 2001, p.216). In the review step, certain case properties are used to evaluate the quality of the cases. These properties are *correctness*, *consistency*, *uniqueness*, *minimality* and *incoherence*. "The restore step uses modify operators to change the contents of the CBR system" (Reinartz, Iglezakis and Roth–Berghofer, 2001, p.216). These *modify operators* are used to increasing the quality of the case base – the potential modifications are: remove, specialise, generalise, adjust, alter, combine, abstract, cross and join cases.

2.2.5 Multi- Method CBR Approaches

This section introduces the most prominent multi- method approach of case-based reasoning.

2.2.5.1 Ontology-based CBR

It is advisable to provide the case-based reasoning system with domain knowledge beforehand to reduce the knowledge acquisition bottleneck. *"The more knowledge is embedded into the system, the more effective [it] is expected to be"* (Recio-García, Díaz-Agudo and González-Calero, 2008, p.54). This domain knowledge embedding is where ontologies can come into place. Ontologies can provide this knowledge. To use the power of ontologies in a Case-based Reasoning system a combined ontology-based and case-based reasoning approach is needed. Ontology-based systems can benefit from structural CBR and vice versa as discussed in Bergmann and Schaaf (2003) and Bichindaritz (2004). Ontology-based CBR *"[…] can take advantage of this domain knowledge and obtain more accurate results"* (Recio-García, Díaz-Agudo and González-Calero, 2008, p.54).

There exist several approaches (Bello-Tomás, González-Calero and Díaz-Agudo (2004), Recio-García, González-Calero and Díaz-Agudo (2014), Roth-Berghofer and Bahls (2008), Bach and Althoff (2012), Dıaz-Agudo and González-Calero (2001), Wang, Hu and Zhang (2003), Dıaz-Agudo, González-Calero, Gómez-

Martın and Gómez-Martın (2005)) that implement a combined Ontology-based CBR. Unfortunately only a limited number of approaches that go beyond taxonomic CBR (Bergmann, 1998) including properties/relations in ontologies. Such an approach has been introduced by Hefke (2004), which is part of the Knowledge Management Implementation and Recommendation (KMIR) Framework (see further: Ehrig, Haase, Hefke and Stojanovic, 2004; Hefke and Abecker, 2006a; b; Hefke, Zacharias, Abecker and Wang, 2006; Hefke, 2008).

Chen and Wu (2003) introduced an RDF-based markup language for case-based reasoning called CaseML. They (2003, p.201) define cases as "[…] an ordered pair where P is the problem space and S is the solution space". CaseML tries to overcome the issues of traditional case languages – as there are: The lack of a *standard vocabulary for describing cases* to enable the interchange of cases among different systems, of a global *convention of integrating domain vocabularies* and of a *flexible case language* for various CBR approaches (structured and unstructured). As shown in Figure 2.8 the CaseML ontology contains a *CaseBase* concept, which acts as a container for *Cases*. A *Case* is linked to *Problems*, *Solutions* and (adaptation) *Rules*.

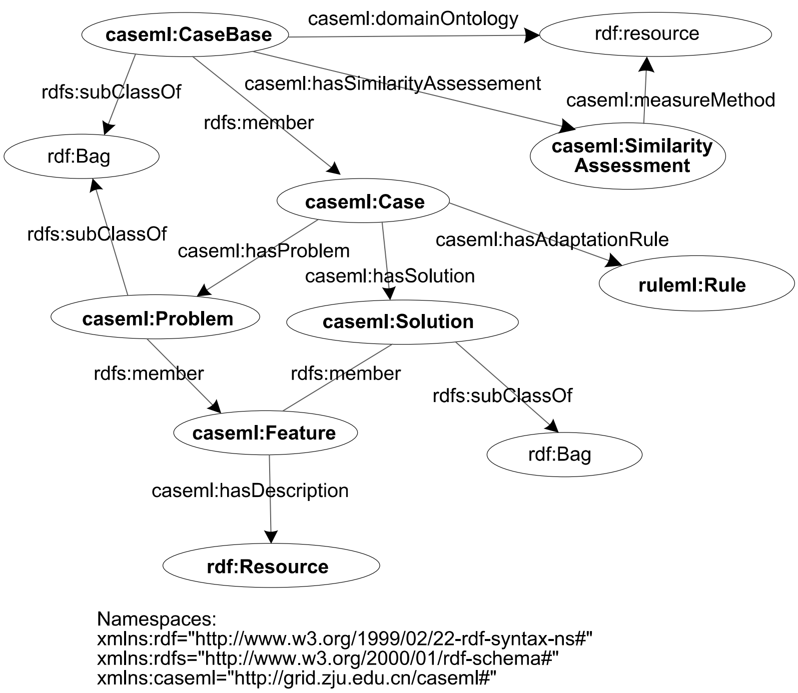

Figure 2.8: RDF Case Base Representation of CaseML (Chen and Wu, 2003)

As Fidjeland (2006) argued, CaseML defines only a small vocabulary with limited expressiveness compared to other approaches. Fidjeland (2006) introduced in his master thesis an OWL vocabulary for Creek (see Section 2.2.6.10). Creek is a case-based reasoning system introduced by Aamodt (1991, 2004). The OWL vocabulary (Fidjeland, 2006) contains basic elements for describing case concepts. The basic concept "creek:Case" has basic relations like "hasFinding", "hasSolution", "hasCaseStatus" and "hasSubcase". Apart from the vocabulary, Fidjeland (2006) introduced a possibility for sharing the case base and the domain model using an OWL representation. Fidjeland (2006) used Jena (McBride, 2001) and jCreek for implementing the proposed OWL vocabulary approach.

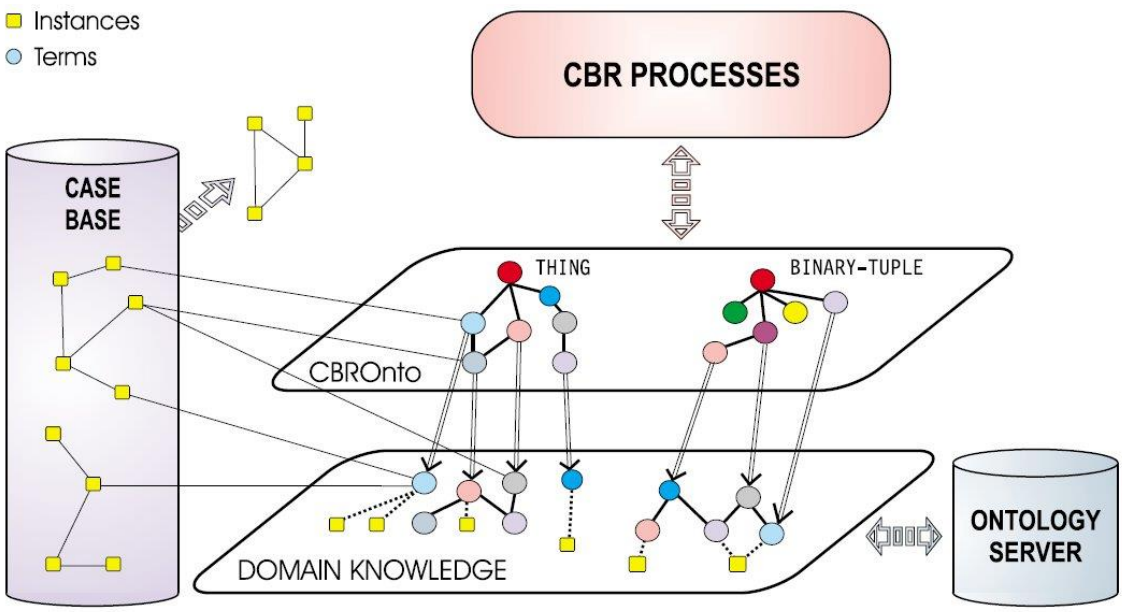

Figure 2.9: CBROnto as Integration Layer of Domain Knowledge (Dıaz-Agudo and González-Calero, 2001)

Dıaz-Agudo and González-Calero (2001) proposed an ontology for describing case-based reasoning processes in a domain in depended way called CBROnto. As depicted in Figure 2.9 CBROnto acts as an integration layer of domain knowledge and case-based reasoning tasks.

Bichindaritz (2004) introduced a case-based reasoning framework called Mémoire for biological and medical cases. That CBR system uses a semantic web standard based interchange language, "[…] bridge the gap between the multiple case-based reasoning systems dedicated to a single domain […]" (Bichindaritz, 2004, p.47). Bichindaritz (2004, p.61) pointed out the benefit of such an ontology-based case-base and interchange possibility as follows: "The perspective of unlimited cooperation between these systems is extremely promising for the improvement of healthcare and biomedical research […]". It is therefore not surprising that the medical and clinical domain has been prominent in the recent past. In the following four main research paths of recent studies, two domains and two methods, in the field of OBCBR are listed[6]:

1. Medical and clinical domain: Shen et al. (2015) propose an OBCBR and multi-agent-based clinical decision support system. The used ontology "[…] employs the domain knowledge to ease the extraction of similar clinical cases and provide treatment suggestions to patients and physicians" (Shen et al., 2015, p.307). Sene, Kamsu-Foguem, and Rumeau (2015) propose an OBCBR approach based on the taxonomic reasoning for telemedicine in the oncology domain with the inclusion of natural language processing (NLP). Delir Haghighi et al. (2013) introduce a development and evaluation of an OBCBR system in emergency medical management.

2. Early warning and emergency domain: Ju et al. (2016) propose an incentive-oriented early warning system (EWS) using OBCBR for predicting co-movements of oil price shocks and the macroeconomy. They underpinned the importance of having a clear procedure model with an end-user and expert involvement when implementing an OBCBR system. Amailef and Lu (2013) introduce an OBCBR implementation for intelligent m-Government emergency response services. It is notable that this

[6] Some verbatim passages presented in this following section, which have been achieved in this thesis project, have been published in the following own authored publication (Martin et al., 2016)

implementation gives end users the possibility to adjust extempore certain similarity weights during retrieval phase and allows them to evaluate the proposed solution (outcome) during the retaining phase.

3. Semantic rule-based CBR: Bouhana et al. (2015) introduce an OBCBR approach for personalised itinerary search systems for sustainable urban freight transport with the inclusion of personalised rules. Bulu, Alpkocak, and Balci (2012) describe a CBR approach for ontology-annotated mammographies. They use a "Semantic Query-enhanced Web Rule Language (SQWRL) to process retrieval of similar masses from annotated mammography collection in OWL" (Bulu, Alpkocak and Balci, 2012, p.11194). SQWRL enhances semantic web rule languages with "SQLlike operations" for the retrieval in an OWL ontology.

4. Fuzzy-ontology-oriented CBR: El-Sappagh, Elmogy, and Riad (2015) introduce a CBR approach using an OWL2 fuzzy ontology that is utilized in a diabetes diagnosis application scenario. The results are promising: "The resulting system can answer complex medical queries related to the semantic understanding of medical concepts and handling of vague terms" (El-Sappagh, Elmogy and Riad, 2015, p.179). They successfully compete against traditional CBR approaches. El-Sappagh, Elmogy, and Riad (2015, p.206) conclude that their "[…] fuzzy-semantic retrieval algorithm outweighs all of the JCOLIBRI algorithms, and it covers their limitations".

2.2.5.2 Contextual Case-based Reasoning

d'Aquin, Lieber and Napoli (2005) introduced an approach that addresses the difficulties when integrating different domain specific ontologies as a knowledge base in ontology-based case-based reasoning. The work uses C-OWL (Contextual OWL) to map concepts in different ontologies representing different contexts or viewpoints (Bouquet et al., 2004). In this case, contexts are defined as "[…] local [and not shared] models that encode a party's view of a domain" (Bouquet et al., 2004, p.1). The approach of d'Aquin, Lieber and Napoli (2005) to represent different viewpoint in the domain and adaptation knowledge.

Wang, Hu and Zhang (2003) introduced an approach to integrate an enterprise based ontology into a case-based reasoning system. This enterprise ontology is used as an integration layer of different knowledge containers in an enterprise e.g. ERP or workflow systems.

2.2.5.3 A Case-based Reasoning Schema

Coyle, Hayes and Cunningham (2002) introduced a schema for describing cases of CBR systems in XML called CBML. "Two documents are needed to represent a case in CBML, the structured document describes the internal format of a case, and the case document contains the content of the case. The case structure document describes the hierarchy and cardinality of the features that can exist in a case." (Coyle, Hayes and Cunningham, 2002, p.2).

CBML can represent two types of possible features. The so-called simple feature represents the feature-value (attribute-value) pair representation. The values of the simple features can be restricted to a specific type (such as number or string). The complex feature type can be used to build a hierarchical structure that contains again simple or complex features (Coyle, Hayes and Cunningham, 2002). Apart from that, Coyle, Doyle and Cunningham (2004) introduced an approach to representing similarity measures in an XML format based on CBML as an addition.

2.2.5.4 Semi-structured CBR

Recio et al. (2005) introduce an approach how to retrieve textual cases by information retrieval techniques. Their work is demonstrated using the case-based reasoning framework jCOLIBRI (Bello-Tomás, González-Calero and Díaz-Agudo, 2004) (Recio-García, González-Calero and Díaz-Agudo, 2014) (Recio-García, Díaz-Agudo and González-Calero, 2008). The motivation to introduce a textual CBR is stated because the domain knowledge is written down in unstructured textual documents.

Selected textual CBR (TCBR) approaches maps unstructured documents to structured case descriptions or semi-structured cases (Recio et al., 2005). To map the unstructured data to the cases, several approaches use domain ontologies.

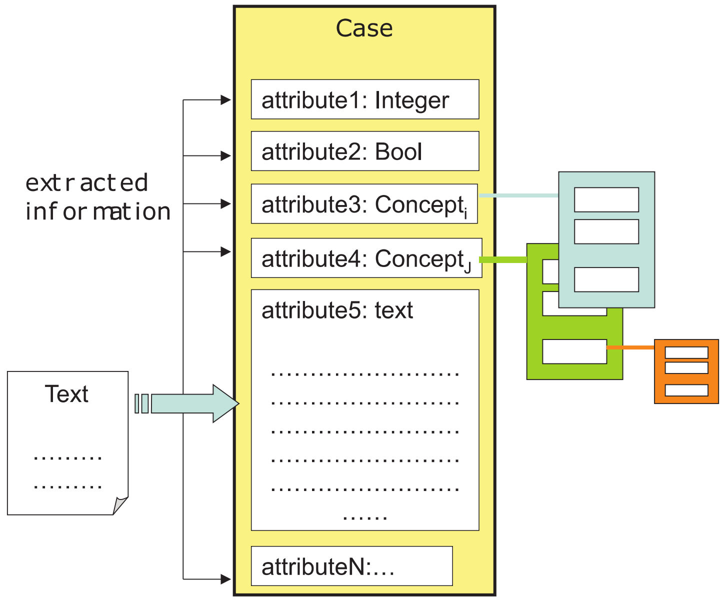

Figure 2.10: jCOLIBRI Semi-structure Case Representation (Recio et al., 2005)

2.2.5.5 Hierarchical Case Representation

Hierarchical case representations are needed to represent cases on different abstraction level. “*The basic idea behind [a hierarchical case representation] is to represent a case at multiple levels of detail, possibly using multiple vocabularies*” (Bergmann, Kolodner and Plaza, 2005, p.211). Bergmann and Wilke (1996) introduced a general framework for hierarchical cases. Figure 2.11 shows the different kind of cases: *concrete*, *abstract* and *hierarchical* case.

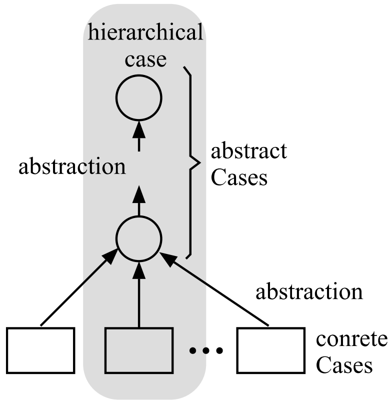

Figure 2.11: Different Kind of Cases (Bergmann and Wilke, 1996)

Abstract cases contain combined and refined descriptions of possible solutions from *concrete* cases. Within abstract cases, there can exist an abstract hierarchy (e.g. represented as a tree) – abstract cases with a different abstraction. Bergmann and Wilke (1996) showed that abstraction of cases in a cased-based reasoning system could be successfully applied in *configuration, design and planning*.

2.2.6 Case-based Reasoning Tools and Frameworks

There exist several frameworks for CBR inter alia COLIBRI (Bello-Tomás, González-Calero and Díaz-Agudo, 2004) and jCOLIBRI2 (Recio-García, González-Calero and Díaz-Agudo, 2014), myCBR (Roth-Berghofer and Bahls, 2008) and (Bach and Althoff, 2012) or CAKE (Bergmann et al., 2006; Maximini and Maximini, 2007). In the following, the most prominent frameworks are introduced.

2.2.6.1 jCOLIBRI and jCOLIBRI2

jCOLIBRI is a framework for creating CBR systems (Bello-Tomás, González-Calero and Díaz-Agudo, 2004). The framework is available under the LGPL (LGPLv2) license in version 2 and is implemented in Java (Recio-García, González-Calero and Díaz-Agudo, 2014). “The main goal of jCOLIBRI2 is to provide a reference platform for developing CBR applications” (Recio-García, González-Calero and Díaz-Agudo, 2014, p.3). jCOLIBRI2 is a multi-layer application. There exist two main components – the COLIBRI studio and the jCOLIBRI engine itself. The studio component can be used to design, construct and configure a case-based reasoning application. The engine and the framework also provide apart from the CBR core functionalities several extensions for textual CBR including information extraction functionalities. Additionally, the engine uses a persistence layer that provides connectors to relational databases, plain text and ontologies using an ontology bridge (Recio-García, González-Calero and Díaz-Agudo, 2014).

2.2.6.2 COBRA

COBRA (Conversational Ontology-based CBR plAtform) is a CBR application platform using an eclipse based editor (Assali, Lenne and Debray, 2010). It aligns the cases to a domain-specific ontology using inference rules. COBRA uses certain parts of jCOLIBRI and Assali, Lenne and Debray (2010) added a layer to support heterogeneous case bases.

2.2.6.3 CBR Shell

The AIAI CBR Shell is a system for "[…] automatic polling of fields for sensitivity to goal finding and the stochastic hill-climbing of ever-fitter combinations of field weights" (Wheeler and Aitken, 2000, p.94). The tool is a closed source available "in demonstration form". It shows the potential of K nearest neighbour and threshold retrieval, multiple diagnostic algorithms, genetic algorithm weight learning and cross-platform implementation.

2.2.6.4 myCBR

Roth-Berghofer and Bahls (2008) introduced a CBR plugin to the ontology editor Protégé called myCBR. The first version of the myCBR plug is designed as a rapid prototyping tool and similarity retrieval engine (Roth-Berghofer and Bahls, 2008). myCBR consists of three main features and views: (1) The explanation editor can be used to describe concepts, which will be used as case characterisation; (2) the retrieval editor provides an interface to the conceptual and backward-explanations. The backward-explanations are a description of the retrieval process, including the local and global similarity features; (3) the similarity measure editor provides a possibility for editing local and global similarity measurements. Stahl and Roth-Berghofer (2008) used the myCBR framework to configure a Web-based CBR application. The protégé based editor can be used to create XML files, containing the similarity and domain model, which can be used in a Web application containing a CBR engine (Stahl and Roth-Berghofer, 2008).

The current version of myCBR is version 3. myCBR 3 is a new development and is now OSGi based, which means that it can be integrated into existing OSGi environments (Bach and Althoff, 2012). The myCBR 3 workbench is a rich client platform (RCP) of the Eclipse development IDE (Roth-Berghofer et al., 2012). myCBR 3 offers a software development kit written in Java to build its CBR applications (Roth-Berghofer et al., 2012). This approach has been successfully applied in decision support, configuration (computer settings), and diagnosis and information composition (Bach and Althoff, 2012).

2.2.6.5 CReP

Manenti and Sartori (2010) presented Case Retrieval Platform (CReP), a framework that provides tools to describe cases, similarity functions on case description parts using a hierarchical structure and string matching.

2.2.6.6 IUCBRF

Bogaerts and Leake (2005) introduced an open source CBR framework written in Java for developing modular CBR applications. The system was a basis for several other CBR systems in the research community.

2.2.6.7 Empolis - Information Access Suite

Empolis have developed a commercially used CBR system. This system is one of "*the most widely used*" CBR-based systems in Europe. The CBR component acts as an "*underlying methodology*" of the Information Access Suite provided by Empolis. This Information Access Suite has been used in a wide range of "*commercial application domains*" (Bergmann et al., 2009, p.3).

2.2.6.8 CAKE

Bergmann et al. (2006) introduced a generic collaboration support architecture called Collaborative Agent-based Knowledge Engine (CAKE). The CAKE can be used for the selection of agents and (sub-) workflows with the usage of CBR technology. "The CAKE architecture, for instance, combines workflow technology, agent technology, and structural CBR to select appropriate agents and workflows in knowledge-intensive application domains using CBR" (Bergmann et al., 2009, p.3).

2.2.6.9 eXiT*CBR

eXiT*CBR is a CBR tool for medical prognosis (López et al., 2011; Pla, López, Gay and Pous, 2013). Pla, López, Gay and Pous (2013, p.1) "[…] propose a user-friendly medically oriented tool for prognosis development systems and experimentation under a case-based reasoning methodology". Besides several generic CBR tools, e.g., jCOLIBRI, eXiT*CBR has been developed under the focus of a particular purpose (medical prognosis) and domain (medicine). eXiT*CBR.v2 extends the isolated CBR system eXiT*CBR.v1 in such a way that collaborative data can be used.

2.2.6.10 CREEK

Creek is a CBR system introduced by Aamodt (1991, 2004). It "*[…] is a knowledge-intensive approach to problem-solving and learning*" (Aamodt, 1991, p.137). There exists a Java-based implementation of CREEK called TrollCreek, which has been used in the petroleum industry (Bergmann et al., 2009).

2.3 Case-based Reasoning and Process Execution

Case-based reasoning has been applied successfully in workflow and process environments. Madhusudan, Zhao and Marshall (2004) and Kim, Suh and Lee (2002) use CBR to support workflow design. The work of Kaster, Medeiros and Rocha (2005) uses CBR in combination with a decision support system. Weber and Wild (2005) enable ad-hoc modifications of workflows using CBR and Rinderle, Weber, Reichert and Wild (2005) proposed a CBR system enabling ad-hoc modifications of workflows using semantic information. Van der Aalst et al. (2005) proposed a case handling system using explicitly and implicitly structured cases. Wargitsch, Wewers and Theisinger (1997) proposed a system called WorkBrain. This system's idea is the elaboration of workflows using certain workflow elements or fragments – called workflow building blocks. In this approach CBR is used at the beginning to configure the instance of the workflow – it does not provide the flexibility to determine the process during runtime. Van Elst, Aschoff, Bernardi and Schwarz (2003) describe a system, which has been elaborated during a research project called FRODO. In FRODO, they introduce a task concept ontology for weakly structured workflows. These workflows can be modified during run-time (instance level). Weber, Wild and Breu (2004) introduced in their work a system called CBRFlow, which combines workflow execution and conversational case-based reasoning (CCBR). CBRFlow uses a case-based reasoning component to handle exceptions to business rules during run-time – it enables "[…]modifications to a predefined workflow model and to provide incremental learning capabilities" (Weber, Wild and Breu, 2004, p.436). Bergmann et al. (2006) and Maximini and Maximini (2007) described a case-based reasoning system architecture called CAKE (Collaborative Agent-Based Knowledge Engine). CAKE can be used for the selection of agents and (sub-) workflows with the usage of CBR technology.

2.3.1 CBR for Workflow Retrieval, Adaptation and ad-hoc Changes

Bergmann and Gil (2011) introduced an approach for retrieving existing workflows based on a query. In contrast to other approaches, this approach the workflows are graph based and semantically annotated (Bergmann and Gil, 2011).

Minor et al. (2010b, p.279) introduced a process-oriented approach using case-based reasoning "[…] to support the reuse of change experience". The introduced approach acts as workflow enactment service, which supports the workflow modifications regarding "[…] ad-hoc changes in order to fulfil change request and late-modelling" (Minor et al., 2010b, p.294). Currently, the workflow modification has to be done manually – "[…] the user has to transfer the solution for adapting the current workflow […]" (Minor et al., 2010b, p.288).

A case "[…] consists of a pair of subsequent revisions of a workflow instance […]"(Minor et al., 2010b, p.288). It is possible to identify a problem part, containing a workflow instance *before a modification* has been made, and a solution part containing the *modified version* of the workflow instance. Based on the case description it is possible to run a query against the described workflow instances when a modification needs to be made (Minor et al., 2010b).

An adequate case representation containing both parts of a workflow instance, control flow and context needs to be defined to represent workflow instances as cases. Minor et al. (2010b, p.279) used two different ways to represent the two parts in their approach. The *context* of the workflow instance represented by attribute-value pairs as known from structural case-based reasoning as introduced by Bergmann (2002) (Minor et al., 2010b). Apart from that the context is modelled in an OWL *ontology* to describe the interdependencies (Minor et al., 2010b). The similarity is assessed using local and global similarity calculation as introduced in Section 2.2.4.1. The *control flow* of the workflow instances is represented as graphs, and the similarity assessment is based on weighted *graph* *edit distances* (Minor et al., 2010b). Finally, the overall similarity is calculated by aggregating the values of the control flow and the context (Minor et al., 2010b).

Minor, Bergmann, Görg and Walter (2010a) introduced an approach for automatic adaptation of workflows using a case-base containing previous workflow adaptions. This approach has been implemented using the CBR engine CAKE (see Section 2.2.6.8).

2.3.2 CBR for Workflow Construction

The construction (modelling) of workflows is a knowledge intensive task – it is important for process engineers to be aware of the services that are available. The work of Leake and Kendall-Morwick (2008) is a way to support scientist that are creating scientific workflows "[…] by suggesting additions to workflow designs under construction" (2008, p.270).

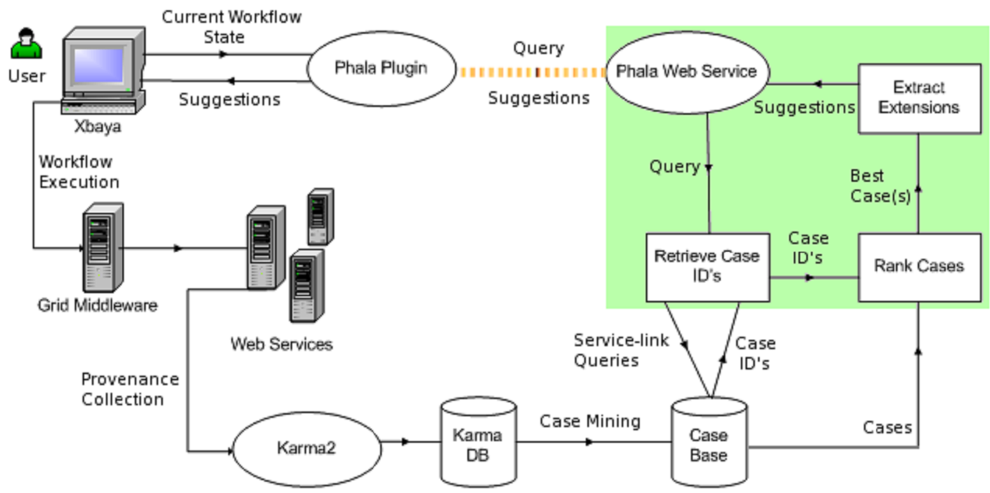

Figure 2.12: Phala (Leake and Kendall-Morwick, 2008)

The approach of Leake and Kendall-Morwick (2008) is implemented as a plugin to the Xbaya software, which is a graphical modeller for scientific workflows. Phala is a case-based reasoning application, which "[…] suggests next steps, or extensions, from a partially authored workflow to an incrementally more developed workflow" (Leake and Kendall-Morwick, 2008, p.273). Figure 2.12 shows the Phala Plugin when providing suggestions to users based on a current state of a workflow. The cases contain execution traces of previous workflows – this execution traces consists of “Sequence […], AND-join […] and AND-split […] control patterns” (Leake and Kendall-Morwick, 2008, p.273).

2.3.3 CBR for Workflow Monitoring

Kapetanakis et al. (2010) introduced a CBR-based monitoring system of workflows. The system informs the process owner or managers about potential issues and gives advice how to deal with the issue. The system is called CBR Workflow Intelligent Monitoring System (CBR-WIMS) and “[…] uses the CBR system to retrieve past useful experience about workflow problems occurred in the past by retrieving similar sequences of events/actions and context in the event log for a given workflow (or workflow part) compared to the current state and recent sequence of events/actions in the operation of the workflow” (Kapetanakis et al., 2010, p.397).

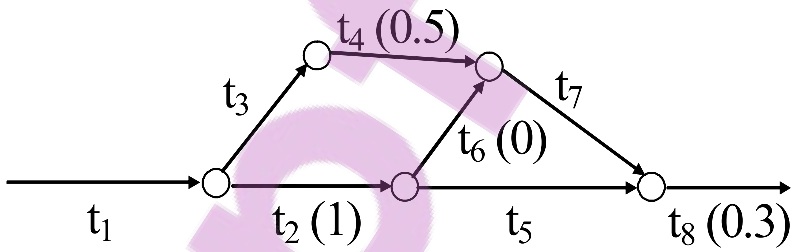

Figure 2.13: Graph Representation of temporal Relationships (Kapetanakis et al., 2010)

Such a representation of a temporal graph is illustrated in Figure 2.13. The used event log uses time intervals based on general time theory (Kapetanakis et al., 2010). The proposed system uses UML activity diagrams, which are mapped using the BPMN specification to BPEL as execution and storage language (Kapetanakis et al., 2010). The cases in the proposed system consist of the mentioned graph based representation containing “[…] events, actions, intervals and their temporal relationships” (Kapetanakis et al., 2010, p.392).

Montani and Leonardi (2012) introduced an approach to monitor business processes by retrieving similar traces of previously executed business processes.

Their work cooperates with the process mining toolkit ProM (Verbeek, Buijs, van Dongen and van der Aalst, 2010).

2.3.4 CBR for Process Life Cycle Support

Weber et al. (2009) developed an approach to support process-aware information systems (PAISs) using case-based reasoning (see Figure 2.14). This ProCycle approach tracks "[…] changes of individual process instances and the propagation of process type changes […]" (Weber et al., 2009, p.1). Process participants can retrieve and reuse contextual "[…] knowledge about previously performed changes. If similar instance deviations occur frequently, process engineers will be supported in deriving improved process models from them" (Weber et al., 2009, p.1).

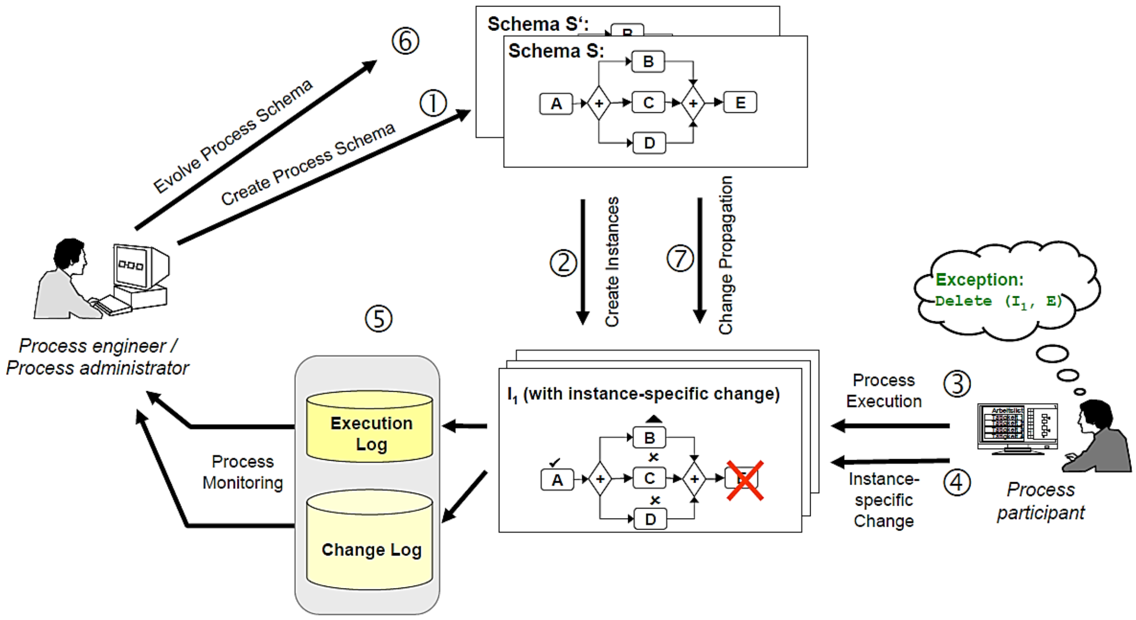

Figure 2.14: Process Life Cycle Support with Adaptive PAISs (Weber et al., 2009)

2.3.5 Business Processes and Business Rules

Business Rules are a possibility to make execution of knowledge-intensive processes more flexible and even slightly adaptable (Endl, Knolmayer and Pfahrer, 1998; Witschel et al., 2010; Martin and Brun, 2010; Feldkamp, Hinkelmann and Thönssen, 2007). These approaches follow the distinction of process knowledge and functional knowledge, where process knowledge is represented in a process model while functional knowledge is represented as business rules (Ross and Lam, 2011). Nevertheless, these approaches depend on initial process models. An initial model means that a basic structure is given and extended with knowledge-intensive tasks (Feldkamp, Hinkelmann and Thönssen, 2007). These knowledge-intensive tasks "[…] are optionally executed depending on information specific for the certain process instance" (Witschel et al., 2010, p.3). This information can be application data, process data, functional data, or further information about needed resources that are semantically described (Brander, Hinkelmann, Martin and Thönssen, 2011b).

The use of business rules for process execution has been developed and evaluated in the European funded research project FIT (Feldkamp, Hinkelmann and Thönssen, 2007) and further developed in MATURE. One outcome of the MATURE project was the agile business process management system KISSmir (Martin and Brun, 2010). The system uses a personal task management front-end, which allows the participants to log all executed tasks and resources used. It also allows adding new subtasks, which can be regarded as a rudimentary kind of case description. The demonstrator is currently being evaluated based on a real application scenario, namely the process of selecting students for admission in an academic programme. In this scenario, exceptional situations and knowledge-intensive decisions are frequent.

2.4 Results and Discussion

As mentioned above, this chapter can give only a brief overview of the literature that was investigated in-depth. The related work shows that there exist related research efforts having the same overall topics or using the same basic technologies, which gives the opportunity to rely on existing technologies and methods. Apart from that, the preliminary literature review shows that this research work is located in a highly relevant research direction.

3 Research Methodology and Design

Scientific research needs to be differentiated from solution development or engineering within the industry. Nevertheless, scientific information research should have an impact upon the industry and society by following the principles of "[…] abstraction, originality justification, and publication" (Österle et al., 2010, p.9).

This section deals with the underlying principles of research and the research methodology itself. It begins with a general investigation of information systems research (Section 3.1), followed by a general investigation on meta-research (Section 3.3). Both of these investigations guide the selection of the underlying research strategy of design science research (Section 3.2). Finally, the selected research design and process (Section 3.4) are described.

3.1 Information Systems Research

The goal of the information systems (IS) research is to simultaneously impact business and society and drive the innovation of those entities. In a widely accepted memorandum of 10 authors, including 111 supporting full professors from the European IS research community, it was stated that IS research needs to be "[…] beneficial for society and business" in an active and innovative manner (Österle et al., 2010, p.7). Conversely, the more behaviouristic IS research approach, rooted in a primarily Anglo-Saxon business school culture, is more focused on observations and user behaviour in a descriptive manner. In the above-mentioned memorandum, Österle et al. (2010) stipulated the European IS view as design-oriented IS research. This design-oriented IS research has a number of similarities to the design-science research, which became "[…] a newly emerging branch of Anglo-Saxon IS research" (Österle et al., 2010, p.8). Both design-oriented IS and design science research try to produce scientific results (artefacts) that are relevant to business and society.

3.1.1 Research Objects in Design-oriented IS Research

Österle et al. (2010, p.8) defined the research objects in design-oriented IS as follows: "IS are sociotechnical in nature and comprised of three object types, namely people (i.e. human task bearers), information and communications technology (i.e. technical task bearers) and organizational concepts (i.e. functions, structures and processes) as well as the interrelationships between them". The contribution of design-oriented IS research to the body of knowledge is made by scientific publications and "[…] by the experiences and knowledge accumulated in business concerning IS, software products, organizational concepts, methods, and tools" (Österle et al., 2010, p.8).

3.1.2 Research Objective in Design-oriented IS Research

The overall research objective of design-oriented IS research is "[…] to develop and provide instructions for action […] that allow the design and operation of IS and innovative concepts within IS (instances)" (Österle et al., 2010, p.8). Design-oriented IS research tries to build "a to-be conception" rather than primarily analysing "an as-is situation" such as in behaviouristic IS research (Österle et al., 2010).

3.1.3 Results in Design-oriented IS Research

A result of a design-oriented IS research and design science research process is that researchers create certain artefacts such as "[…] constructs (e.g., concepts, terminologies, and languages), models, methods, and instantiations […]" (e.g. implemented prototypes) (Österle et al., 2010, p.9). Such artefacts can then result in specific manifestations for e.g. "[…] axioms, guidelines, frameworks, norms, patents, software (with open source code), business models, enterprise start-ups and much more" (Österle et al., 2010, p.9).

3.1.3.1 Research Process and Methods in Design-oriented IS Research

Design-oriented IS research and design science research have a common research process with similar iterative phases (see the following Figure 3.1).

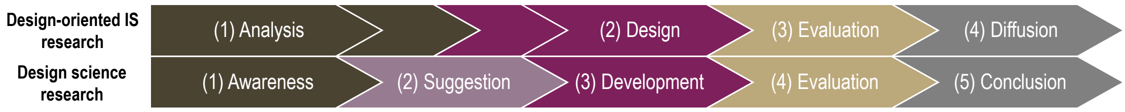

Figure 3.1: Design-oriented IS Research and Design Science Research

Österle et al. (2010, p.9) defined the research process for design-oriented IS research using four phases called analysis, design, evaluation, and diffusion. The design science research process model of Vaishnavi and Kuechler (2004) consists of five phases namely awareness, suggestion, development, evaluation and conclusion. The design science research process derived from Vaishnavi and Kuechler (2004) was used in this study and is further explained in Section 3.2. The evaluation and diffusion phases in design-oriented IS research and the evaluation and conclusion phases in design science research can be clearly mapped. The awareness phase of the design science research fits into the analysis phase of design-oriented IS research. Parts of the design science suggestion phase may fit into the analysis phase of design-oriented IS research, but may also fit into the design phase because a suggestion can be considered a tentative design of an artefact. Although the research phases are not entirely similar, both research processes use similar methods. Methods in the analysis phase "[…] are surveys, case studies, expert interviews, and IS analysis (e.g., database analysis)" (Österle et al., 2010, p.9). For the design phase "[…] demonstration or prototype construction, modeling with CASE tools, reference modeling, and method engineering […]" is used (Österle et al., 2010, p.9). Moreover, the evaluation can be done using "laboratory experiments, pilot applications (i.e., instantiation of prototypes), simulation procedures, expert reviews, and field experiments (i.e., instantiations in a number of user organizations)" (Österle et al., 2010, p.9).

3.2 Design Science Research

Design science research (DSR) can be seen as an artefact creation and investigation process. Such artefacts can be constructs, models, methods, instantiations or design theories (Hevner and Chatterjee, 2010). Hevner and Chatterjee (2010, p.5) defined design science research as "[…] a research paradigm in which a designer answers questions relevant to human problems via the creation of innovative artifacts, thereby contributing new knowledge to the body of scientific evidence. The designed artifacts are both useful and fundamental in understanding that problem." This artefact creation process generates and accumulates knowledge (Owen, 1998). "The fundamental principle of design science research is that knowledge and understanding of a design problem and its solution are acquired in the building and application of an artefact" (Hevner and Chatterjee, 2010, p.5).

The general design science research methodology based on Vaishnavi and Kuechler (2004) contains five steps (see Figure 3.2): awareness of the problem, a suggestion for a solution, the development of the artefact(s), the evaluation of the artefact(s) and a conclusion. This work followed the methodology of design science research (DSR) as introduced by Vaishnavi and Kuechler (2004) was enhanced with elements from Peffers et al. (2007). As displayed in Figure 3.2, the process begins with (1) an awareness phase in which the research problem is addressed and motivated in a problem relevance description. The outcome of the

(2) suggestion phase is a tentative design of the approach and the system. For iterative and agile development, it is suitable to execute the (3) development in incremental cycles. Iterative and agile approaches divide the requirements into smaller portions and demonstrate the solution (also called increments) using small use cases (Métrailler, 2011). After performing the incremental development phase, the artefact should be evaluated as a whole within the (4) evaluation step. Finally, the results are generally presented in scholarly and professional publications in the (5) conclusion step.

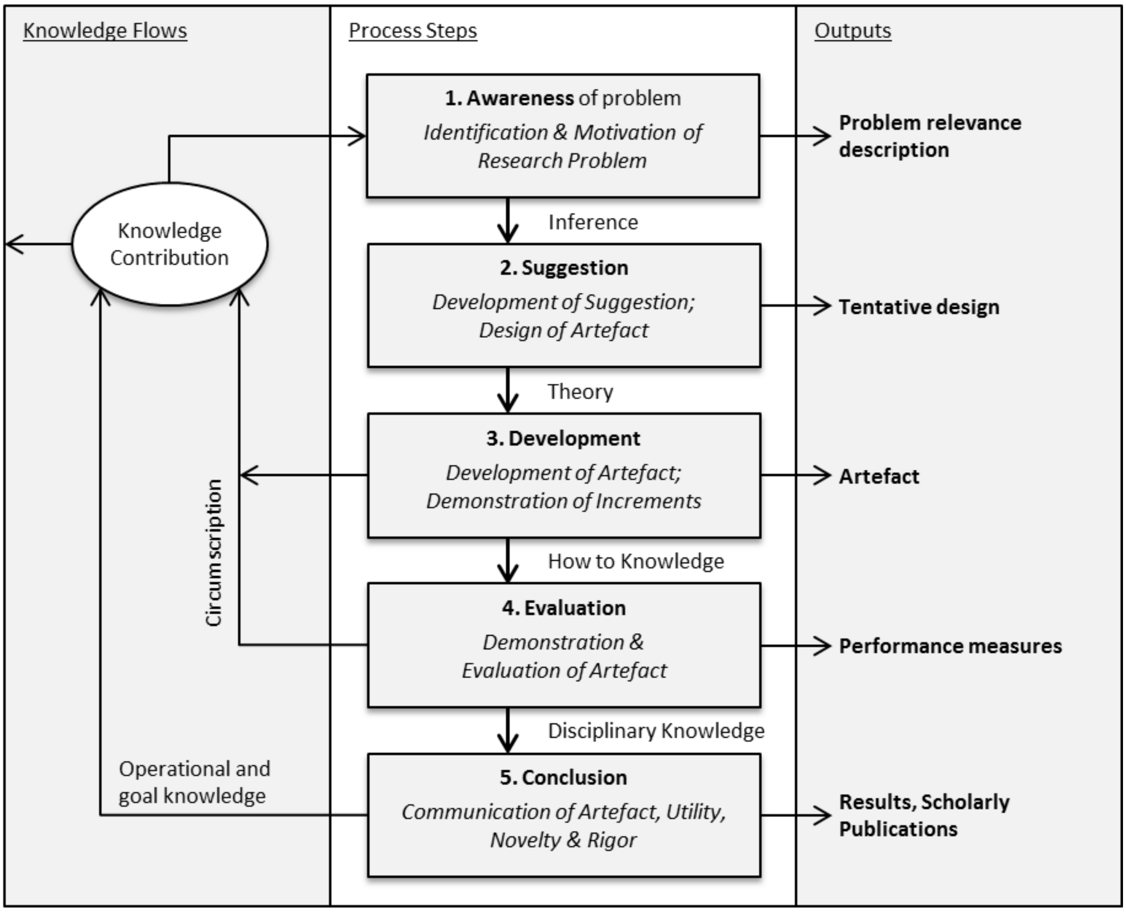

Figure 3.2: General Methodology of Design Science Research (adapted from Vaishnavi and Kuechler (2004) and enhanced with elements from Peffers et al. (2008))

3.3 Meta Research Methodology

The meta-research methodology is the underlying basis for every research project. In the following, this is described using a possible meta-research layering model (adapted from Saunders, Lewis and Thornhill (2007)).

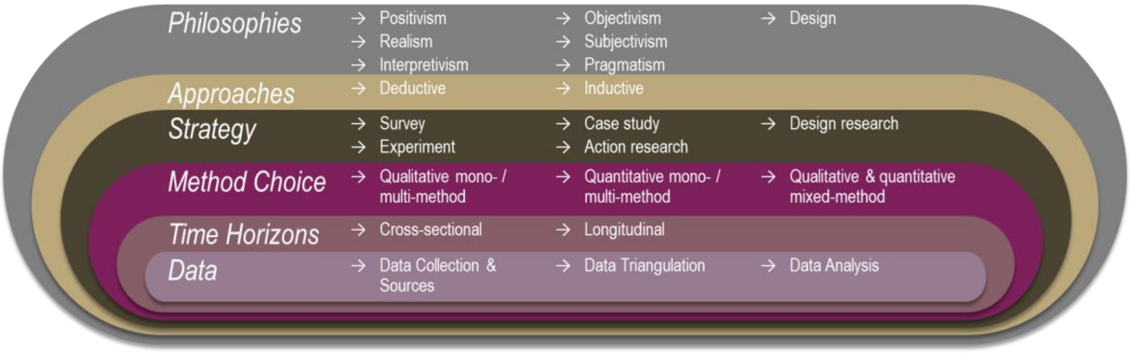

Figure 3.3: Research Layering (adapted from Saunders, Lewis and Thornhill (2007))

The research layering (see Figure 3.3), provides an overview of the different aspects of a research design and possible philosophies, approaches and strategies. It introduces layers – beginning with philosophy, then the approach, the strategy, the choices and the time horizons – to the research techniques and procedures. The original research onion of Saunders, Lewis and Thornhill (2007) was adapted with regards to the layering. It is therefore obvious that the layering and underlying sequence are not determining factors – the research layering provides a guideline for the research undertaking.

3.3.1 Theory and Research Strategy

As presented in Figure 3.4 and Figure 3.7, the research layering was divided into two separate layerings. This section, along with the corresponding Figure 3.4, focuses on the philosophy, approaches and strategy of a research project.

Figure 3.4: Theoretical Perspectives and Research Strategies (adapted from Saunders, Lewis and Thornhill (2007) and Denzin and Lincoln (2011))

The method selection for the collection and analysis of data, insights and knowledge, will be discussed in Section 3.3.2.

3.3.1.1 Research Philosophy

According to Saunders, Lewis and Thornhill (2007) there are three major thought processes regarding research philosophies (basic beliefs):

1. The first is *epistemology*. "[It] concerns what constitutes acceptable knowledge in a field of study" (Saunders, Lewis and Thornhill, 2007, p.102). The underlying question for *epistemology* based on Guba and Lincoln (1994, p.108) is: "What is the form and nature of reality and, therefore, what is there that be known about it"? In short, "How do I know" (Durant-Law, 2005, p.14)?
2. *Ontology* "[…] is concerned with nature of reality. To a greater extent than epistemological considerations, this raises questions of the assumptions researchers have about the way the world operates and the commitment held to particular views" (Saunders, Lewis and Thornhill, 2007, p.108). The underlying question for *ontology* based on Guba and Lincoln (1994, p.108) is: "What is the nature of the relationship between the knower […] and what can be known"? In short, "What exists" (Durant-Law, 2005, p.14)?
3. Finally, *axiology* is "[…] a branch of philosophy that studies judgements about value. Although this may include values […] in the fields of aesthetics and ethics, it is the process of social enquiry […]" (Saunders, Lewis and Thornhill, 2007, p.110). The underlying question for *axiology* is based on Vaishnavi and Kuechler (2004, p.21): "What values does an individual or group hold and why"? In short, "What is valuable" (Durant-Law, 2005, p.14)?

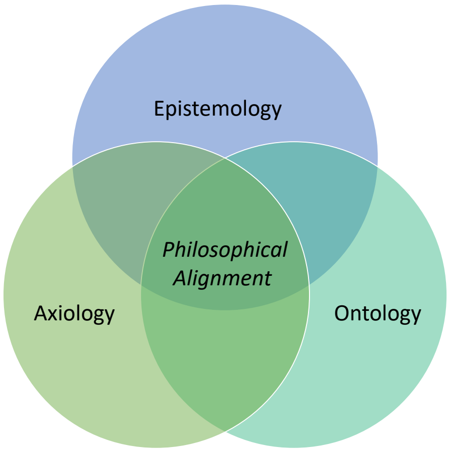

Figure 3.5: Philosophical Trinity (adapted from Durant-Law (2005))

It is often difficult to decide in which research philosophy an entire research work, namely a thesis, takes place and "the debates on both epistemology and ontology have had a competitive ring to them. The debate is often framed regarding a choice between either the positivist or the interpretivist research philosophy" (Saunders, Lewis and Thornhill, 2007, p.120). In practice, it is unrealistic to choose one approach for an entire work. It is more important that a researcher knows his or her philosophical stance, by finding the "sweet spot". This "sweet spot" is referred to by Durant-Law (2005, p.15) as a philosophical alignment (see Figure 3.5) "[…] where the overlap between ontological, epistemological and axiological positions is maximised". Nevertheless, the following prominent research philosophies can be identified:

Positivism deals with the social reality, meaning that a result of an approach "[…] can be law-like generalisations similar to those produced by the physical and natural scientists" (Remenyi and Williams, 1998, p.32).

- The *realism* philosophy can be regarded as "that what the senses show us as reality is the truth. […] The theory of realism is that there is a reality quite independent of the mind" (Saunders, Lewis and Thornhill, 2007, p.104).
- Researchers following the *interpretivism* research philosophy argue that "[…] the social world is far too complex to lend itself to theorising by definite 'laws' the same way as the physical sciences, […] it is necessary for the researcher to understand differences between humans in our role as social actors. This emphasises the difference between conducting research among people rather than objects such as trucks and computers" (Saunders, Lewis and Thornhill, 2007, p.106).

- *Objectivism* "[…] portrays the position that social entities exist in reality external to social actors concerned with their existence" (Saunders, Lewis and Thornhill, 2007, p.108).
- *Subjectivism* "[…] holds that social phenomena are created from the perceptions and consequent actions of those social actors concerned with their existence" (Saunders, Lewis and Thornhill, 2007, p.108).
- In *pragmatism* the research questions are the most important element that determines which research philosophy should be adopted. "One [research philosophy] may be 'better' than the other for answering particular questions" (Saunders, Lewis and Thornhill, 2007, p.110).
- Besides the two prominent research philosophies (positivist and interpretivist) in natural and social science, Gregg, Kulkarni and Vinzé (2001) introduced, based on Guba and Lincoln (1994), *socio-technologist/developmentalist* as a meta-research assumption. Vaishnavi and Kuechler (2004) adapted this assumption and denoted it as a *design* research philosophy.

Table 3.1: Major Research Philosophies and Beliefs compared (adapted from Gregg, Kulkarni and Vinzé (2001) and Vaishnavi and Kuechler (2004))

Positivism	***Interpretivism***	***Design***
Epistemology: What is the nature of knowledge?		
Objectivity is important. Detached observer of truth.	Subjectivity: Interactive link between participant(s) and researcher. Explicit values and findings.	Objective, interactive and iterative construction within a context. "Knowing through making".
Ontology: What is the nature of reality?		
One reality. Knowledge based on probability.	Multiple social constructed realities.	Known context; multiple technical and socially constructed realities.

Axiology: What is of value?		
Truth; universal and beautiful; prediction.	Understanding; situated and description.	Control; creation; progress and improving; understanding.
Methodology: What is the research approach for obtaining the desired knowledge and understanding?		
Observation; primarily quantitative, decontextualized and statistical.	Participation; primarily qualitative, hermeneutical and dialectical.	Primarily developmental; Measure impact of an artefact.

3.3.1.2 Research Approach

Saunders, Lewis and Thornhill (2007) distinguished between the two research approaches (s. Figure 3.6):

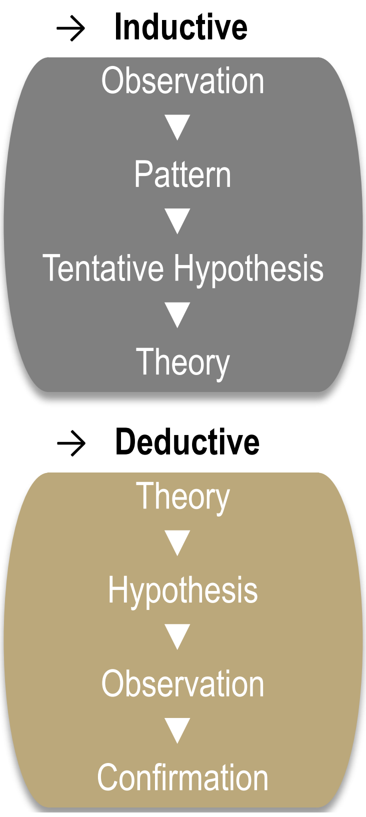

Figure 3.6: Induction and Deduction (adapted from Trochim (2006))

- From the specific to the general (Trochim, 2006): The *inductive* approach has its origin in the social sciences. It begins with an observation with the goal of identifying patterns based on cause-effect relationships. The data collection could be done, for example, through interviews. After collecting the data, the analysed data is used to formulate a hypothesis and devise a theory.
- From the general to the specific (Trochim, 2006): The *deductive* approach begins by deducting a hypothesis from theory, expressing the hypothesis in operational terms, making observations, examining the outcome (confirmation or falsification) and if needed, modifying the theory (Robson, 2002).

3.3.1.2.1 Research Approach in Design-oriented IS

Following a design-oriented (information systems) research strategy, e.g. design-science research, a deductive research approach is primarily applied. A formal (using mathematics) or semi-formal (e.g. conceptual) deduction would be the ideal situation (Österle et al., 2010). However, in design-science research, it is rarely the case that an artefact can be evaluated formally (Österle et al., 2010). It is more likely that "[…] design-oriented IS research takes advantage of natural-language (i.e. argumentative) deduction, taking into account existing theories and models" (Österle et al., 2010, p.9). In contrast to this, a single case study inferencing would be an example of an inductive approach within design-science research (Österle et al., 2010). However, this is also rarely the case.

3.3.1.3 Research Strategy

The research strategy can be seen as a process or a plan for a research project. Saunders, Lewis and Thornhill (2015, p.177) have defined research strategy "[…] as a plan of how a researcher will go about answering her or his research question". The research layering lists the following research strategies: experiment, survey, case study, action research, grounded theory, ethnography and archival research. Of course, this is not an all-encompassing list and the following paragraphs will only describe the most prominent ones.

Action research has been defined by Avison, Lau, Myers and Nielsen (1999, p.94) as the following: "Action research combines theory and practice (and researchers and practitioners) through change and reflection in an immediate problematic situation within a mutually acceptable ethical framework". Saunders, Lewis and Thornhill (2007) have identified four common sub-themes of action research. First, it is research in action and conducted with those who experience the issues directly. Secondly, there exists a partnership between practitioners and researchers, whereas the research can act as an internal or external consultant. Thirdly, an iterative process of diagnosing, planning, taking action and evaluating occurs. Fourthly, the gained knowledge should be used to inform other contexts (e.g. within the organisation).

Robson (2002, p.178) has defined *case study* as "[…] a strategy for doing research which involves an empirical investigation of a particular contemporary phenomenon within its real life context using multiple sources of evidence". Saunders, Lewis and Thornhill (2007) view case studies as explanatory and exploratory research which use various data collection techniques. A case study is able to generate answers to the questions "why", "what" and "how".

Experiments can be seen as a "[…] classical form of research that owes much to the natural sciences" (Saunders, Lewis and Thornhill, 2007, p.136). Experiments tend to be used in exploratory and explanatory research and try to answer the "how" and "why" questions (Saunders, Lewis and Thornhill, 2007). Saunders, Lewis and Thornhill (2007) have provided a summary of what experiments typically involve: Experiments begin with a definition of a theoretical hypothesis. A selection of samples from known populations are allocated to the conditions, the experimental group and the control group. The variables are then manipulated and controlled.

Design Science Research, as described in Section 3.2, can be seen as an artefact creation and investigation process in contrast to observations on existing artefacts. This creation process can be referred to as actions which generate and accumulate knowledge (Owen, 1998). The artefact is the main object of investigation and the overall goal of the process.

3.3.2 Research Methods of Collection and Analysis

For conducting research, an adequate choice of a research method is essential. Examples (see Figure 3.8) of such research methods are experiments, testing, simulations, observations, questionnaires, interviews and case studies.

Figure 3.7: Research Methods of Collection and Analysis (adapted from Saunders, Lewis and Thornhill (2007))

Researchers must make certain decisions (see Figure 3.7) concerning, qualitative and quantitative research methods, the time horizon, the data collection, sources and analysis.

3.3.2.1 Research Method Classification and Choice

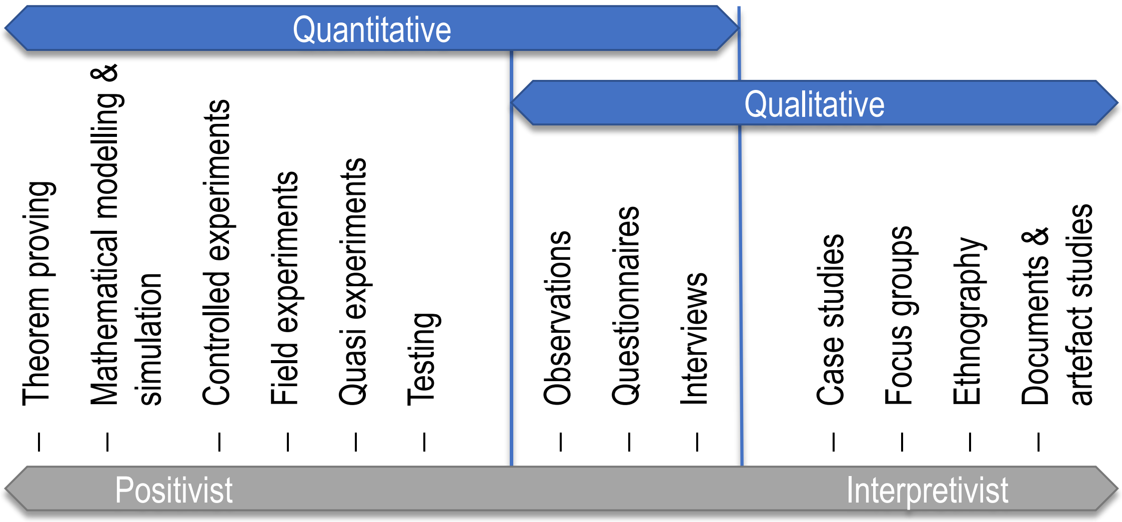

Figure 3.8: Classification of Research Methods (adapted from De Villiers (2005))

As introduced in Section 3.3.1, research is based upon certain paradigms and philosophical assumptions. Positivism deals with reality, meaning that the results are generated using empirical methods and are regarded as absolute (De Villiers, 2005). Interpretivism assumes that the reality is complex and depended upon the given context. Therefore, the results of an interpretivistic research tend to be subjective. Nevertheless, interpretivistic research is regarded as adequate for investigating social and social-technical phenomena (De Villiers, 2005). Qualitative methods produce a smaller sample of data that is less quantifiable

(Guba and Lincoln, 1994) and qualitative research has an interpretive nature (De Villiers, 2005). In addition, qualitative methods produce data based on statistical methods (De Villiers, 2005). Nevertheless, the distinction is not ideal, but limited and narrow.

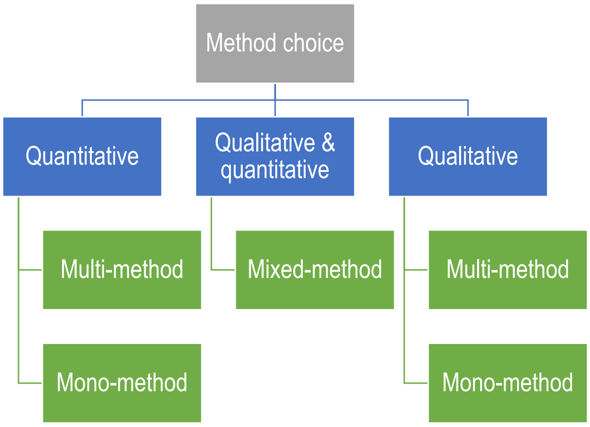

Figure 3.9: Method Choice (adapted from Saunders, Lewis and Thornhill (2007))

Ideally, "[…] quantitative and qualitative research may be viewed as two ends of a continuum, which in practice are often mixed" (Saunders, Lewis and Thornhill, 2015, p.165).

Figure 3.9 presents the method classification as adapted from Saunders, Lewis and Thornhill (2007), which can be used to describe whether a research project has a quantitative or qualitative method choice based on one method (mono-method) or more than one method (multi-method). As previously mentioned, it is also often the case that quantitative and qualitative methods are selected, which is then known as mixed-method choice.

3.3.2.2 Time Horizons

Time horizons have been distinguished, according to Saunders (2007, p.148), as being between cross-sectional and longitudinal. *Cross-sectional studies* investigate “a particular phenomenon […] at a particular [and a single point in] time”, whereas *longitudinal studies* investigate certain phenomenon at multiple time points over a longer period.

3.3.2.3 Data Collection and Sources

Many researchers tend to use a more simplified explanation as to what the difference is between primary and secondary data. *Primary data* is data that has been or will be collected by the researchers themselves using a quantitative or qualitative data collection method. *Secondary data* is data that has been or will be collected by someone else and is a data source that already exists. The differentiation focusses on whether the (secondary) data is collected second-hand or (primary) data is collected and analysed first-hand. It appears that it is less important how the data is collected regarding the research problem, method and context. Although, the reliability and trustworthiness of the data can be ensured by the researchers themselves when the entire data collection processes is controlled first-hand. Nevertheless, whether to use primary or secondary data should be decided based on the research fit (Hox and Boeije, 2005). Based on Hox and Boeije (2005), the differentiation between primary and secondary data can be defined in conjunction with Figure 3.10 as follows:

Definition 3.1: Primary and Secondary Data

Primary data is data that has been collected and analysed first-hand for a specific research problem and with a first-hand research method selection that fits the research problem. *Secondary data* is data that has been collected for a different research goal and reused for a different research question or in a different context.

Researchers may work or have already worked on multiple research projects with similar research problems and similar approaches for addressing these problems. Such research projects generally need certain data in order to create an application scenario of the research as an application-oriented character or perform an evaluation if needed by the research design. It is, therefore, obvious that the data that has been collected and analysed for similar research problems with similar research questions in different contexts should be available. It is, therefore, also obvious that such first-hand data can be beneficial to the current research project by providing data source triangulation. Such first-hand data can be regarded as primary and secondary data. However, in order to avoid a misunderstanding, the following definition defines first-hand secondary data as an additional data type in conjunction with Figure 3.10.

Definition 3.2: First-hand Secondary Data

First-hand secondary data is data that has been collected and analysed first-hand for similar research problems with similar research questions in different contexts and a research method selection that fits the similar research problem.

First-hand secondary data may already be published in scientific publications or project reports, or captured in the same time-frame as the research that uses the first-hand secondary data.

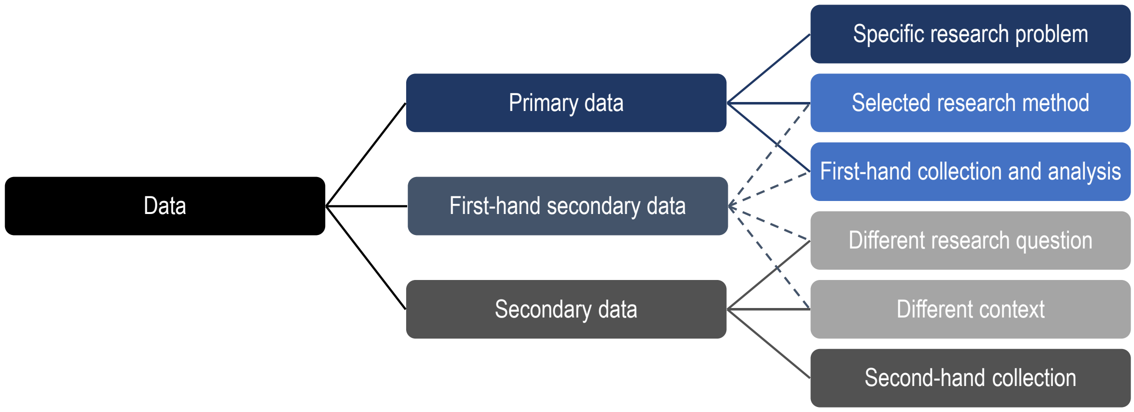

Figure 3.10: Primary and Secondary Data

3.3.2.4 Data Triangulation and Analysis

Data triangulation ensures the validity of the analysis of a phenomenon and helps to gain a deeper understanding of that phenomenon (Cohen and Crabtree, 2006). A triangulation can be achieved by multiple data collection methods, sources, investigators or theoretical perspectives (Cohen and Crabtree, 2006). "By combining multiple observers, theories, methods and data sources, [researchers] can hope to overcome the intrinsic bias that comes from single-methods, single-observer, and single-theory studies" (Denzin, 2009, p.313). Based on Cohen and Crabtree (2006) and Patton (2015, p.661), the following types of triangulation are identified:

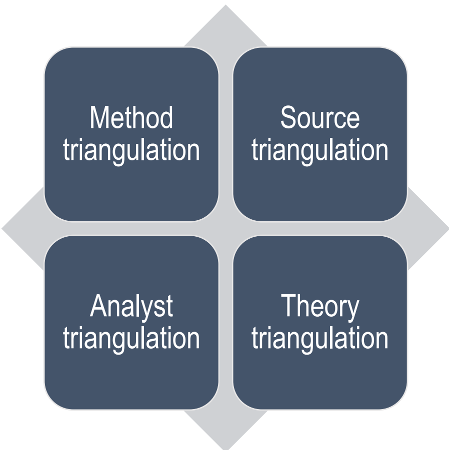

Figure 3.11: Data Triangulation

1. Method triangulation - multiple data collection methods.
2. Source triangulation - different data sources from different time horizons or data capturing settings.
3. Analyst triangulation – different observers and/or analysts.
4. Theory triangulation – different theoretical perspectives (theory).

3.4 Research Design and Process

In previous sections, the general meta-research was described wherein the following needed elements for this thesis are selected:

- Philosophy: Design
- Approach: Inductive
- Strategy: Design Science Research Strategy
- Method: Qualitative Multi-Method

The research was conducted using the design science research cycles based on Hevner, March, Park and Ram (2004) and the methodology based on Vaishnavi and Kuechler (2004). "Design-science research must produce a viable artefact in the form of a construct, a model, a method, or an instantiation" The research findings were derived from the elaboration of these artefacts. These artefacts, theories and processes were developed in design cycles (see Figure 3.12). Additionally, requirements were gathered during the relevance cycle to ensure the business's suitability in design science research. The rigor cycle ensured that the research work was well grounded using existing research work and methodologies.

This research project used the design, relevance and rigor cycles in conjunction with the with the design science research methodology based on Vaishnavi and Kuechler (2004).

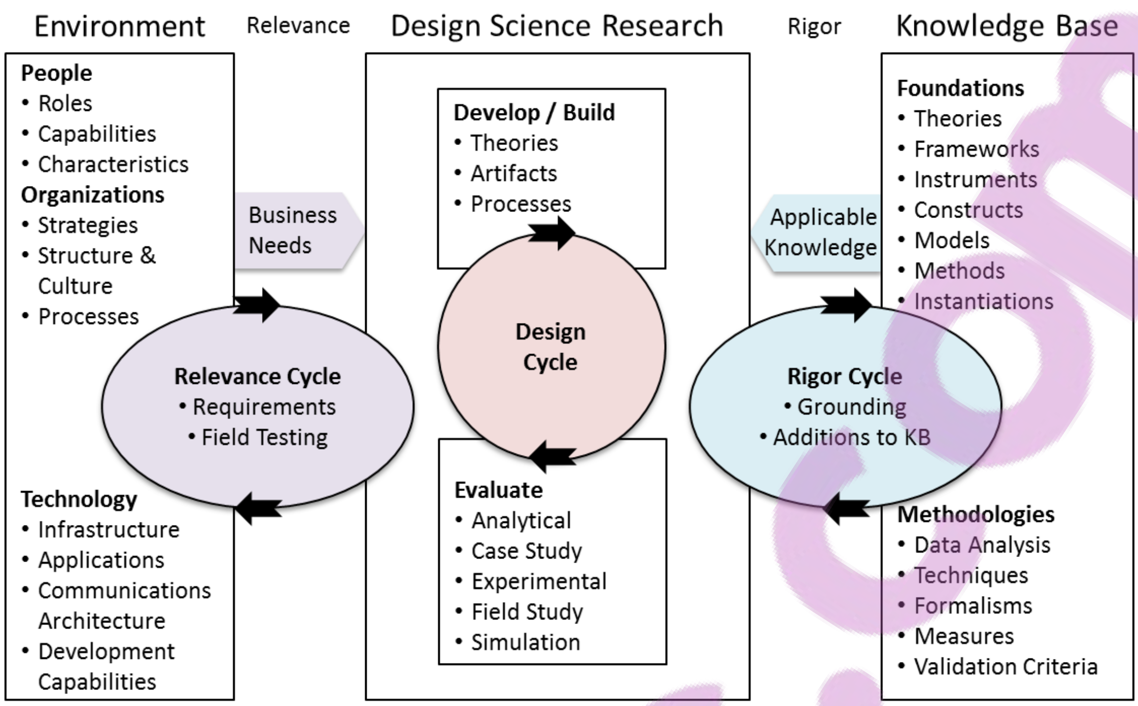

Figure 3.12: Design Science Research Cycles (adapted from Hevner and Chatterjee (2010) and Hevner et al. (2010; 2004))

3.4.1 Research Methods, Strategy and Choice

This study was designed as a multi-method research choice – it involved several research and engineering methods (sometimes called research strategies – see Saunders, Lewis and Thornhill (2007)). These were:

- Design Science Research: As mentioned before, design science research is the overall research method used within this study.
- Application scenario (case study): The case study provided the business and real-life context of this research project. "A case study is an empirical inquiry that investigates a contemporary phenomenon within its real-life context" (Yin, 2003, p.13). The application scenario, which is called the admission process, was created in the development phase. It was used in the development phase to demonstrate the prototype system iteratively and to act as a source of requirements. Finally, it was used in the evaluation phase when the artefact was investigated.

- Requirements analysis (sometimes called requirements engineering): Requirements analysis and engineering is an engineering method and part of a software development process (Sommerville, 2011). Requirements engineering consists of several techniques or methods, such as use case analysis or personas (Cheng and Atlee, 2007). In this study, the requirements analysis was the first step performed within the development phase of the case model (Chapter 6.4.1) and final prototype development (Chapter 8) using the application scenario and interviews.
- Literature review: The literature review provided a theoretical basis for this study; It ensured that this work "creates" something new. The literature review was linked the specific research objectives (Biggam, 2008) and research questions. One outcome of the literature review was a technical report, which was used as an introduction to case-based reasoning (CBR).
- Interviews: The goal of the interviews within this study was "[…] to gather valid and reliable data […]" (Saunders, Lewis and Thornhill, 2007, p.310) and answer the research questions. According to Bingham and Moore (1941), an interview is a "conversation with a purpose" (cited by Kahn and Cannell, 1957, p.97). The first interview(s) were conducted during the awareness phase in order to elaborate upon and verify the application scenario and describe the problem. Additionally, the interviews were a data source for the requirement analysis of the case model (Chapter 6.4.1), case-based reasoning services (Chapter 7) and final prototype development (Chapter 8).
- Experiments: "The purpose of an experiment is to study causal links" (Saunders, Lewis and Thornhill, 2007, p.136). The experiments helped answer the "how" research questions. Within this study, experiments were used in the evaluation phase.

3.4.2 Data Collection, Sources and Analysis

The following research methods or strategies mentioned above were used for data collection and analysis: The *application scenario, requirements analysis* and *interviews* provided qualitative and primary data from a real-life context. The *literature review* was used to obtain secondary data.

3.4.2.1 Types of Data and Sources

In this study, three types of data sources were used. The primary data source type for this study was first-hand secondary data, which was the main data source for the elicitation and design of the application scenario. In addition, secondary data sources, as well as literature and reports were also used. Moreover, primary data gained from interviews and analysed case data were also used for this study. The application scenario was derived from real-world use cases and the literature. Pre-existing results, use cases and existing application data from related research projects were transferred to and re-analysed for this project.

- Admission process scenario (main application scenario): The admission process for the Master of Science programmes at the University of Applied Sciences and Arts Northwestern Switzerland FHNW was a starting point for the research. The application scenario (see Section 4.2) was first analysed during the research project MATURE[7]. I was a member of the project team and actively involved in developing the various use cases. Even after the MATURE project was completed, the different stakeholders acted as domain specialists and application and evaluation partners in this research project.

[7]The Integrating Project MATURE was co-funded by the European Commission under the Information and Communication Technologies (ICT) theme of the 7th Framework Programme FP7 (contract no. 216356) - mature-ip.eu

- Offering and project management scenario: Additional requirements were derived from the research project [sic!][8], in which a CBR system for the offer process and project management of a software company was developed. I was the project leader for this research project, which had almost the same duration as this dissertation. The [sic!] research project ended in 2015. This research project also made it possible to derive requirements and have access to first-hand secondary data and primary data specifically for this thesis. This offering and project management scenario is used in this thesis for the evaluation and is described in Section 9.4.1.
- Process learning scenario: Finally, the research project LearnPAd, where I was a member of the project team, provided confirmatory evaluation results based on a different application scenario - the e-learning support of servants in public administrations. This research project delivered additional first-hand secondary data for the evaluation phase of this dissertation to underline the acceptance of the suggested and implemented CBR approach. This process learning scenario was used for this thesis' evaluation and is described in Section 9.4.2.

3.4.2.2 Primary Data

As described in the method section, two primary data collection techniques, (1) interviews and (2) document and artefact study, were used in this research work.

The qualitative interviews were conducted during the awareness phase of this study in order to create the application scenario of the admission process, as well as during the suggestion and implementation phases. The stakeholders of the admission process were interviewed. The interview participants (domain specialists) were selected because they work within the context of the process and

[8][sic!] is funded by the Swiss Confederation's innovation promotion agency (CTI).

were chosen based on their availability. There were seven universal potential participants working within the context of the application scenario. Moreover, the minimal sample size was three. The interviews began with a semi-structured qualitative questionnaire followed by an open discussion concerning certain aspects of the application scenario or the suggested approach. The questionnaire is available in the appendix (see Appendix-C: Excerpt from the Interview Documentation) of this study. The raw answers, including audio recordings, are confidential but paraphrased within this study. Interviews with stakeholders:

- The overall description of the application scenario and the case data were verified by the dean of the study programme (process owner) during an interview on the 5th December 2014. This interview was recorded and freely transcribed.
- The detailed process model and the initial version of the suggested approach were verified by two process members during two interviews, one on the 31st March 2015 and the other on the 10th April 2015.

The document and artefact study was applied as a qualitative research method in order to gain case data and process data to create the application scenario within the awareness phase. This study used generalised, artificial and anonymized data that was derived from student data (e.g. education, work experience). Since there was no need to identify real persons or relationships, the data were able to be anonymized and made artificial without distorting the research results. Additionally, the primary data collected was anonymized in order to comply with the collective employment agreement and the federal act on data protection, while the original raw data has not left the original storage and must remain confidential.

3.4.2.3 Secondary Data

The admission process was initially elicited based on the results of the MATURE research project. As mentioned, I was a member of the project team and was actively involved in creating the underlying use case. Thus, I had access to the raw data for this application scenario. The application scenario was made publically available within two deliverables of the MATURE research project:

- Nelkner et al. (2011): Nelkner, T., Hu, B., Martin, A., Brander, S., Braun, S., Riss, U., Attwell, G., Hinkelmann, K. and Berrio de Diego, M., 2011. *D2.3 / D3.3 Design and Delivery of Prototype Version V2 of PLME / OLME*.
- Cook et al. (2012): Cook, J., Schmidt, A., Bradley, C., Barnes, S.-A., Bimrose, J., Brander, S., Braun, S., Brown, A., Kump, B., Kunzmann, C., Mazarakis, A., Nelkner, T., Pearson, C. and Taylor, I., 2012. *D6.4 Summative Evaluation Report*.

The case data and a process model were derived from real data. Written recordings from interviews and notes from workshops were able to be used in this study. Everything was completely anonymized before storage in order to comply with the regulations of the application partner and the federal act on data protection. The following first-hand secondary data were used for this study:

- Initial process model and description
- Case data (tasks, task descriptions, emails, documents, related process and task patterns and concrete user data)
- Written remarks from interviews with different stakeholders
- Written remarks from workshops and written observations from evaluations with end users

The CBR approach and the suggestion were developed for two application scenarios and used cases simultaneously. Besides the admission process, the offer process of a software company made it possible to confirm the usefulness of the suggested approach iteratively. The approach, as described in Chapter 5, was developed in this dissertation project and was also developed and verified using the application scenario of the [sic!] research project. This thesis project used first-hand secondary data from a related research project, but no primary data were used from the software company. The first-hand secondary data used were:

- References to publications containing the suggested approach (called ICEBERG) of this study, applied and implemented in a different context/use case
- References to evaluation results

Finally, in the research project Learn PAd, the suggested approach was implemented in a different use case and evaluated with end-users of public administrations. The main deliverable that describes the implementation and evaluation of the suggested approach of this study was:

- Thönssen, Witschel, Hinkelmann and Martin (2016): Thönssen, B., Witschel, H.-F., Hinkelmann, K. and Martin, A., 2016. *Experience Knowledge Mechanisms and Representation*.

The evaluation results of the Learn PAd project were used in this study as first-hand secondary data as an additional source regarding evaluation data triangulation. Additionally, the results of the Learn PAd concerning this approach were published in a delivarable (Thönssen et al., 2016) and will be published in a book chapter (Emmenegger et al., 2017, In Press). The first-hand secondary data used from the Learn PAd research project were:

- References to evaluation results
- References to a publication containing the suggested approach of this study applied and implemented in a different context and use case

3.4.3 Adherence to the Design-Science Research Guidelines

Hevner et al. (2004, p.83) has presented research guidelines for design-science research projects. These guidelines were used to evaluate whether this thesis was in adherence to design-science research. These guidelines and the responses to these guidelines are presented (Hevner et al., 2004, p.83) below.

Guideline 1 – Design as an Artefact – “Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation”:

> This research work provides the following viable artefacts: (1) the ontology-based CBR and process execution approach as model, (2) the ontology-based CBR and process execution implementation as instantiation, (3) a case model, (4) case-based reasoning services as instantiation, (5) an ontology as model and instantiation and (6) a procedure model.

Guideline 2 – Problem Relevance – “The objective of design-science research is to develop technology-based solutions to important and relevant business problems”:

> The problem relevance description and application scenario, the summative evaluation and the derived requirements ensure that the concepts of this thesis are a solution to relevant business problems. In addition, the implementation of the prototype indicates that the artefacts developed were technology-based.

Guideline 3 – Design Evaluation – “The utility, quality and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods”:

> The evaluation is a significant component of this thesis. The prototype was demonstrated under real world circumstances.

Guideline 4 – Research Contributions – “Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies”:

> The research provided verifiable contributions as presented in Chapter 1.8 and mentioned as publications in Chapter 10.1.3.

Guideline 5 – Research Rigor – "Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact":

This research was based on an extensive technical report and literature framework, as presented in Chapter 2.

Guideline 6 – Design as a Search Process – "The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment":

As presented in Chapter 3.4.4, the performed development step was highly iterative. Apart from that, it was possible to run additional cycles if the results did not satisfy the expectation of the evaluation partners and requirements of the application scenario.

Guideline 7 – Communication of Research – "Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences":

As explained in Chapter 1.8 and 10.1.3, several research publications were created from this research. Furthermore, the results of this thesis were presented to business partners of the mentioned research projects in Chapter 3.4.2.

3.4.4 Research Procedure

The aforementioned methodology of design research guided this research project and ensured that the specific output of this design science approach was generalizable. Figure 3.13 provides an overview of the research process and shows the outcome of the main steps including three artefacts and the main artefact (the ICEBERG-PE approach and the prototype).

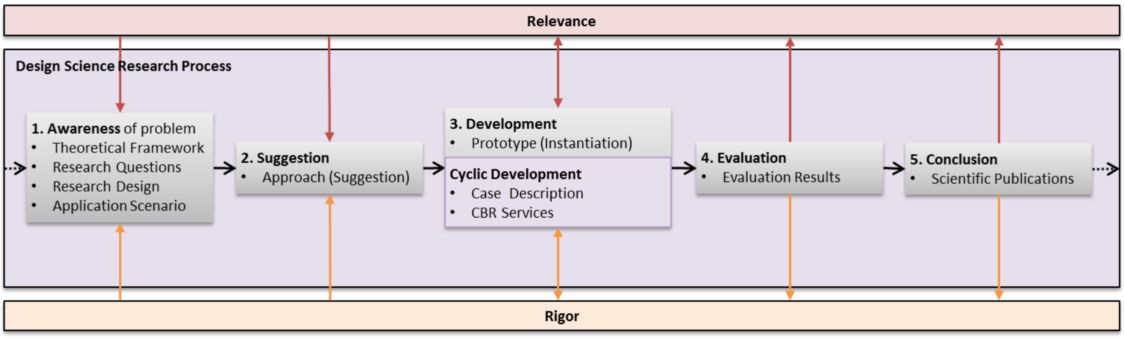

Figure 3.13: An Overview of the Design Science Research Process and Main Outputs

In detail, the five design science research steps are defined as follows (see Figure 3.14):

I. The **awareness of the problem** led to the theoretical framework presented in Chapter 2 (the literature review). The application scenario of the admission process was elaborated together with the domain specialists in order to place the research project in real-world context. This was done in an iterative manner based on interviews. Moreover, the domain specialist involvement supported the problem relevance of this work.

II. The **suggestion** phase was done based on the theoretical framework. The main activity of this phase was the creation of the approach, which is described in Chapter 4. This Chapter 4 contains a conceptual framework, an ontology framework, and the underlying methodology.

III. The **development** phase was divided into three activities based on the four research questions. The activities were not isolated because they correlated with other activities. The development phase occurred over several iterations. How the research questions (RQ) are addressed is described as follows:

- RQ 2 (Case description) – Case Model: The case description research question is addressed in "The Case Model" Chapter 5 and was derived from the theoretical framework. The case model artefact was verified and demonstrated using the application scenario.

- RQ 3 (Case-based reasoning services) - Case-based Reasoning Services: The similarity, adaptation and learning mechanism was derived from the theoretical framework and developed by the case model. This is addressed in Chapter 7.
- RQ 1 (Prototype) – Approach and Prototype: The new case-based reasoning and process execution approach and prototype was the main artefact of this design science research. The ICEBERG-PE approach was described as a concept and instantiated within a prototypical system as a proof of concept. The implementation is addressed in Chapter 8; The suggested approach is described in Chapter 4.

This development phase runs as iterative development process. As Sommerville (2011) suggests, the work was divided into smaller pieces (sub-artefacts) and developed in an iterative way.

IV. The **evaluation** phase was used to observe and measure how well the ICEBERG-PE approach was able to provide a solution to the problem. The evaluation was done using an exploratory scenario – the admission process for a University programme – together with the domain specialists. The ICEBERG-PE approach was evaluated in depth based on the application scenario which contained objective criteria, within a business environment. The evaluation results indicated whether to iterate back or to continue to the conclusion.

V. The **conclusion** activity includes presenting "[…] the problem and its importance, the artefact, its utility and novelty, the rigor of its design, and its effectiveness to researchers and other relevant audiences, such as practicing professionals" (Hevner and Chatterjee, 2010, p.30). Parts of it were communicated to relevant audiences of several journals and conference papers (Martin, Emmenegger and Wilke, 2013; Witschel et al., 2015; Cognini, Hinkelmann and Martin, 2016; Martin et al., 2016; Emmenegger et al., 2017, In Press). The communication activity was executed during the entire study in order to involve the research audience. This was achieved by presenting relevant results as soon as they became available. In the end, the findings were summarised and the dissertation was finalised by the conclusion which presents the possible limitations and proposals for future research.

3.4.5 Validity, Reliability and Trustworthiness

As described in Section 1.4.2, this work attempted to answer three research questions during the creation of the design science research artefacts (case model, CBR services, ontologies and the approach itself). Several "techniques" were applied in order to ensure that this study was reliable and the results were valid:

1. Application scenario (case study technique): The admission process application scenario was created based on existing and running processes. Additionally, the application scenario was reviewed by domain specialists. This was done to ensure that all of the research questions could be proven based on a real-world scenario.
2. Domain specialist involvement: The involvement of domain specialist ensured that the research was addressing relevant business problems.
3. Functional testing: The answers to the research questions were validated during the development phase. This was done using test data, elaborated upon during the requirement analysis based on the application scenario and the theoretical framework.
4. Evaluation: The main "technique" that ensures validity, reliability and trustworthiness of a design science research project is the evaluation

itself – explained in Section 3.4.4 (Research Procedure). The approach was evaluated in this phase using the prototype (as a proof of concept). The evaluation addressed research questions two, three and four.

5. Research project cooperation: This research was conducted as part of and in cooperation with research projects (see Section 3.4.2 "Data Collection, Sources and Analysis) with related research questions.
6. Intended scientific publications: It was intended that the results of this thesis be further published in scientific publications. This would ensure the acceptance of this research work within the scientific community, since it would provide a further sign of validity, reliability and trustworthiness.

3.4.6 Research Ethics

All activities of this research work were done in accordance with the UNISA guidelines – all activities complied with the policy on research ethics (University of South Africa, 2007). All sources were cited using the Harvard referencing style in a correct, complete and consistent manner based on the Harvard referencing guideline of the Anglia Ruskin University (2011). The data were collected from the university information system of the University of Applied Sciences and Arts Northwestern Switzerland FHNW and were anonymized, from which artificial cases were generated. The collected data was anonymized in order to comply with the Federal Act on Data Protection (FADP) of the Swiss Confederation, while the original raw data did not leave the original storage because it must remain confidential. The data were anonymized in such a way that any similarity to persons and any relation to entities were purely coincidental and unintentional. The artificial cases were used during the interviews and for evaluation purposes. Finally, this study received written approval from the Research Ethics Review Committee of the School of Computing.

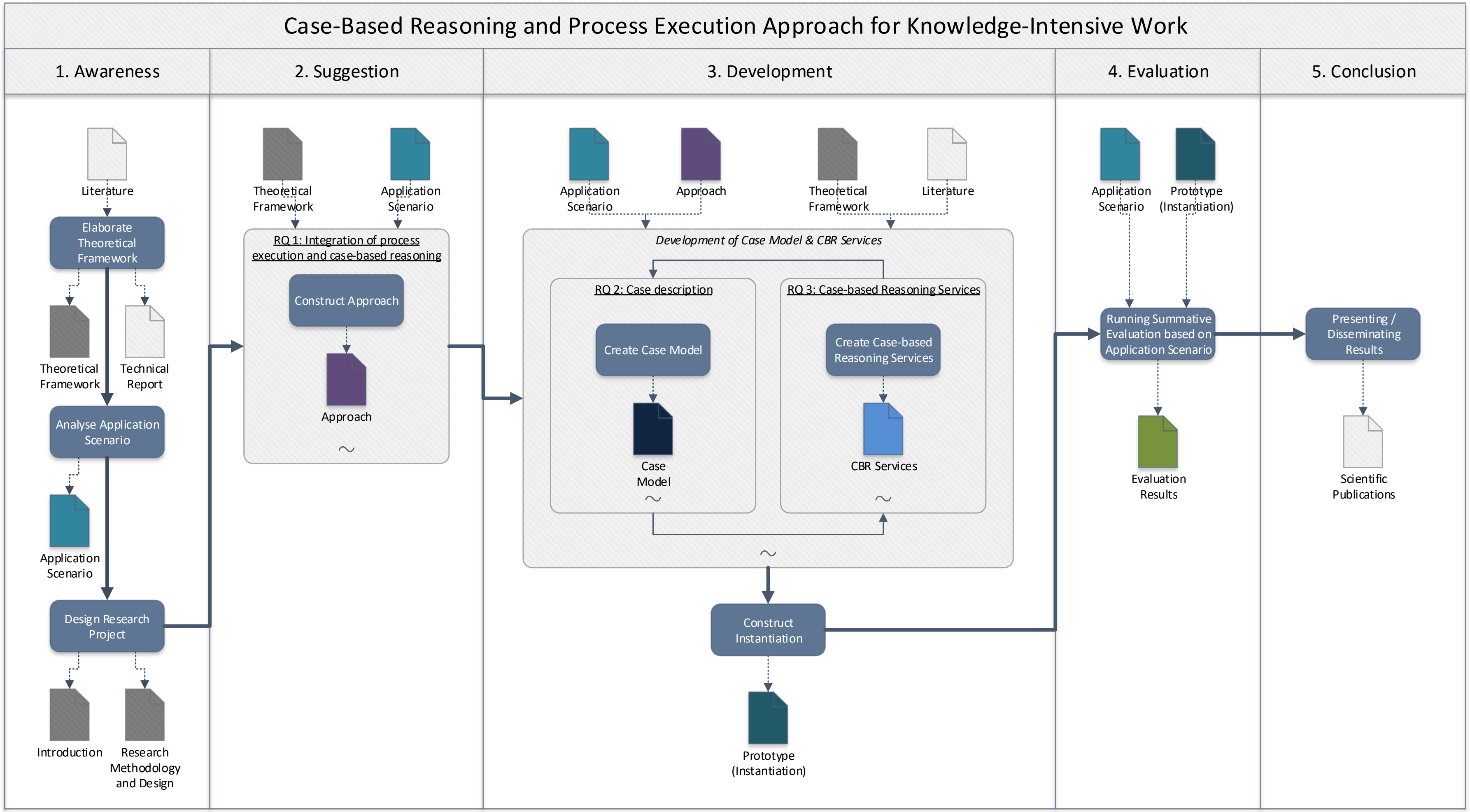

Figure 3.14: Research Procedure

4 Problem Relevance and Application Scenario

This chapter introduces an application scenario that will be used throughout the entire thesis to illustrate the research problem, the suggested approach, the implementation and the evaluation. In addition to the main application scenario, two additional use cases will be briefly introduced within this chapter, which are used within the evaluation chapter in order to ensure data (source) triangulation.

4.1 Data Source for Application Scenario

The first-hand secondary data from the admission process were complemented with primary data based on interviews, documents and artefact studies to create an underlying application scenario for this study, referred to as the admission process application scenario.

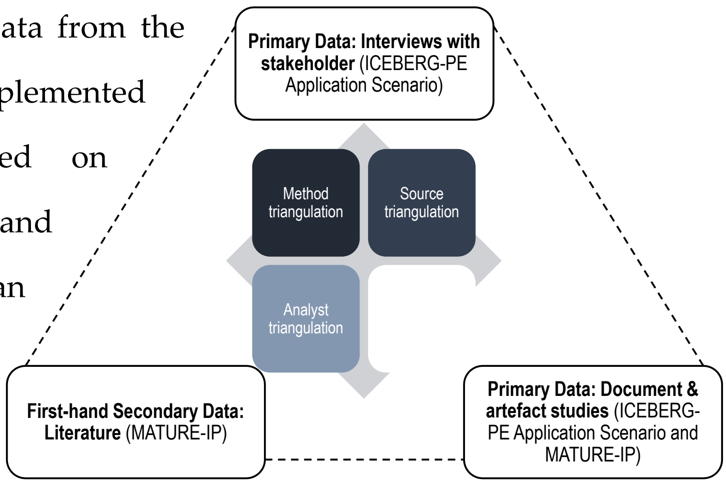

Figure 4.1: Application Scenario Triangulation

The data source triangulation of the admission process application scenario, as displayed in Figure 4.1, was ensured by various methods (method triangulation:

interview, document and artefact studies and literature), data sources (source triangulation) and analyses (analyst triangulation: apart from myself, other research project members supported the analysis).

The initial application process model was presented to the public in a research project deliverable (Nelkner et al., 2011). Additionally, case data and remarks from observations and workshops with stakeholders were presented in a research project deliverable (Cook et al., 2012).

The data that was captured and presented here as an application scenario and process model, as presented in Figure 4.2 and Figure 4.3, were verified by stakeholders of the application scenario. This was done using the following interview sessions:

1. The data was verified by the process owner (the dean of the master programme) during an interview on the 5th December 2014. This interview was recorded and freely transcribed.
2. The process model was verified by two process members during two interviews, one on the 31st March 2015 and a second one on the 10th April 2015.

Finally, the application scenario was partially published in the following own authored publication by Cognini, Hinkelmann and Martin (2016).

4.2 Application Scenario - the Admission Process

The study admission of the Master of Science (MSc) programmes at the school of business/FHNW University of Applied Sciences and Arts Northwestern Switzerland served as the main application scenario for this research work. The admission process is a highly knowledge-intensive process that was performed in Switzerland by each university individually. In this scenario, knowledge-intensiveness was expressed by the high variability of applicants coming from diverse universities, countries and degrees.

This admission process was used to verify the application of a prospective student of the Master of Science (MSc) programme in the Business Information Systems (BIS) programme based on the admission requirements. The admission requirements encompass the following elements:

- **Academic qualification**: A prospective candidate must hold an academic degree in compliance with the following requirements:
  - **Bachelor degree in a related field of study**: A prospective candidate must hold a Bachelor of Science or Bachelor of Arts in Business Informatics, Information Systems, Computer Science, Business Administration or a related field of study.
  - **Bachelor degree obtained from an accredited institution**: The candidate must hold a bachelor's degree from an accredited institution.
  - **Good or excellent grades**: Moreover, only candidates with good or excellent grades will be accepted.
- **Working experience**: Since the MSc BIS is located at a Swiss university of applied sciences and arts, every prospective student must have working experience of at least one year, preferably within a related field of study.
- **Good linguistic abilities in English**: Since the MSc BIS programme is an international programme with modules and courses taught in the English language, it is expected that every prospective student have adequate linguistic abilities in English.

Figure 4.2 shows a representation of the admission process of the Master of Science (MSc) programme in Business Information Systems (BIS) in BPMN 2.0. This representation was initially created by the dean of the MSc BIS programme and then enhanced and verified during an interview (see Appendix-C: Excerpt from the Interview Documentation). The admission process begins when a prospective student's application arrives. The application is analysed during the first activity "prepare eligibility check", where the study assistant prepares the eligibility check. She or he collects all of the information needed to allow the dean of the programme to check the eligibility of the candidate; This is then achieved by the dean during the "check eligibility" activity. If the candidate is eligible, she or he is invited for an oral interview by the study assistant during the "invite for interview" activity. Otherwise, a rejection letter is sent. The main goal of the interview is to verify the eligibility of the applicant. This eligibility validation activity is highly knowledge-intensive. If, after the interview, the candidate is accepted by the admission commission during the "decide for acceptance" activity, the administration department determines the tuition fee during the "determine tuition fee" activity. Finally, an acceptance letter is sent to the candidate by the study admission during the "send acceptance letter" activity. If the candidate is not eligible, a rejection letter is sent.

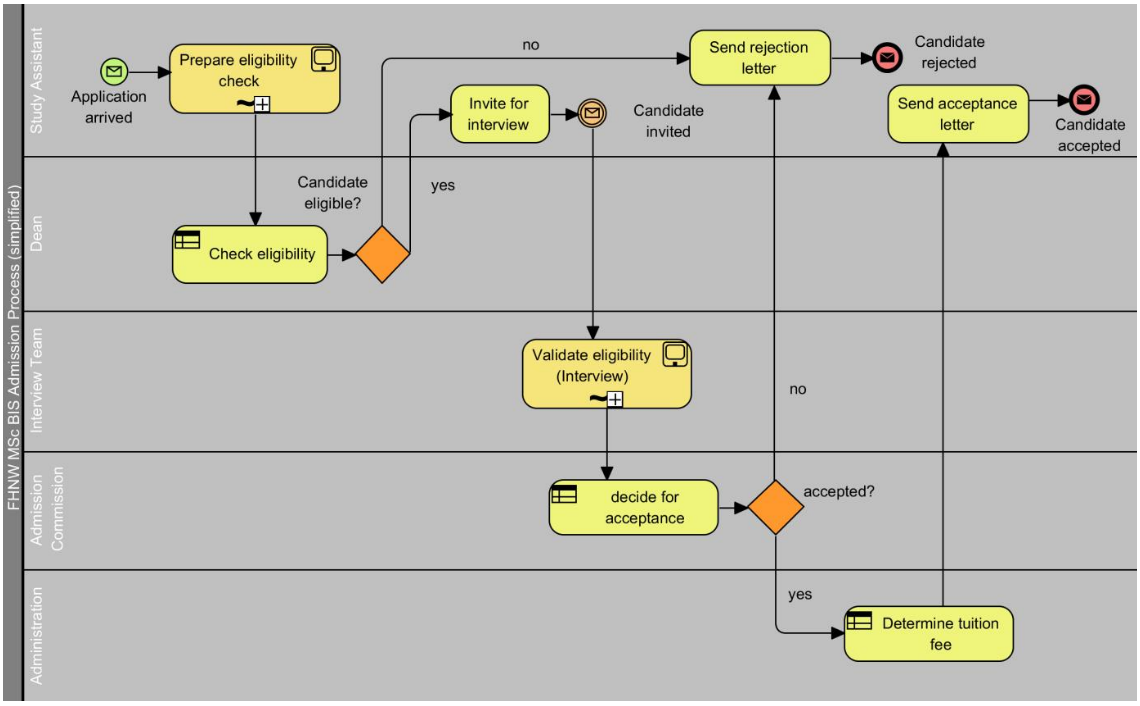

Figure 4.2: FHNW MSc BIS Study Admission Process

Although this process is modelled as a structured process, the activity "prepare eligibility check" is modelled as a complex ad-hoc sub-process. As previously mentioned, the activities modelled in Figure 4.3 were identified in interviews with the stakeholder.

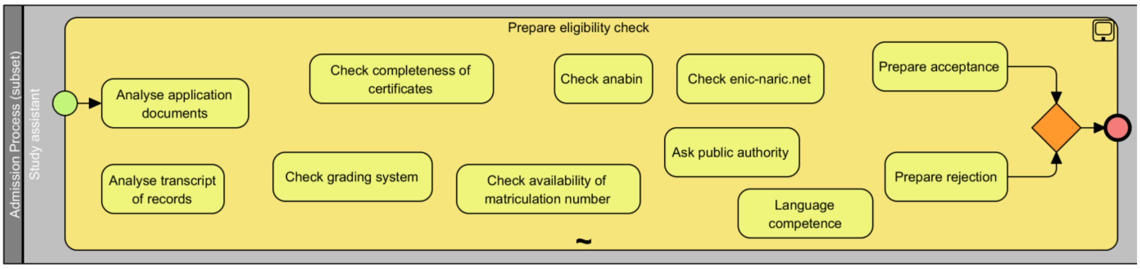

Figure 4.3: Prepare eligibility check as described by the dean of the master programme

In this activity, it is determined whether the bachelor's degree of an applicant qualifies for the master programme. Because candidates are international, they represent a wide variety of degree and certificates. If the bachelor's degree is unknown, the transcript of record is analysed. This means that it is verified whether the university, from which the candidate received his or her bachelor's degree, is accredited.

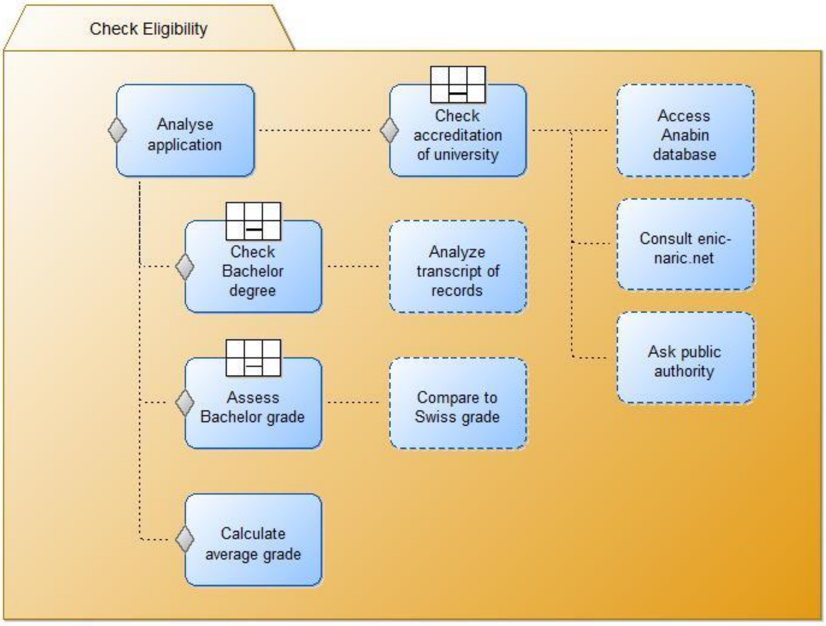

Figure 4.4: Check eligibility as CMMN case[9]

[9] Modelled by the MSc BIS dean.

In case the university is unknown, the study assistant can access an academic database containing qualification information known as Anabin. The Anabin database, however, is incomplete. Moreover, there are several other databases and on-line resources as well. For example, enic-narci.net provides access to resources in which many countries have listed their accredited universities. The selection of this resource depends upon the country where the applicant received his or her certificates. If the university cannot be found within any resource, the study assistant can ask the public authority for confirmation.

Furthermore, the eligibility depends on the average grade of the bachelor's degree, which must be at least a "B". If the average grade is not mentioned in the transcript of record, it is calculated by the study assistant. For unknown grading systems, one must discover their comparable Swiss grades. It is not clear in advance which activity is not required, nor in which order they were executed. For a number of activities, the entry criteria are known while other activities depend upon the judgement of the performer.

Figure 4.4 shows the check eligibility activity, partially modelled using CMMN (OMG, 2014, 2016). This figure reveals the interdependencies of certain activities. If the entry criteria of a task are known, the task is modelled with solid lines. If the execution of a task depends on human judgement, it is modelled as a discretionary task with dotted lines. Nevertheless, the potential activities are not final, nor is it pre-defined that the student assistant will perform certain activities. The invocation of certain activities highly depends upon the actual application case.

4.3 Case and Process Execution Data

The data collected and presented in the following were anonymized to comply with the collective employment agreement and the federal act on data protection. Any similarity to persons and any relation to entities were purely coincidental and unintentional. The original raw data did not leave the original storage and must be kept confidential. Table 4.1 shows the basic data of the application case "B" containing the name and metadata information of a fictitious applicant; The demonstrative cases are denoted by letters "A" to "D". Further, their additional information is provided by the study assistant and performed activities during this process instance. Finally, problems and solutions that can occur in similar cases are also listed. The problems and solutions were determined during an evaluation with administration staff of FHNW. They were used as an initial indication of case-based reasoning (CBR) problems and solutions and the corresponding potential case model.

Table 4.1: Basic Data - Case B

Name	***Susan Fisher***
Nationality	US
Degree	Bachelor of business administration (BBA) in «Management»
Final degree university	Davenport University, USA
Additional information	Student has been working in Switzerland for 4 years
Performed activities	Analyse Application, Check Approval, Prepare Response (Acceptance)
Problems	University is not in list of anabin[10] website
Solutions	Ask student for proof of accreditation or/and call swissuniversities[11], Swiss ENIC[12]

[10] Anabin is a database of the standing conference of the ministers of education and cultural affairs of the states in the federal republic of Germany for the recognition for foreign university diplomas.

[11] Swissuniversities is an association founded by the universities, universities of applied sciences and universities of teacher education in Switzerland as replacement of the former conferences CRUS, KFH and COHEP based on the "Federal Act on the Funding and Coordination of the Higher Education Sector (Higher Education Funding and Coordination, HEdA)".

[12] Swiss ENIC is a service centre of swissuniversities that issues recommendations of recognition for foreign university diplomas.

Table 4.2 lists the remarks or suggestions concerning case "B" derived from interview notes initially described in Cook et al. (2012). They were remastered and re-analysed for this thesis.

Table 4.2: Remarks/Suggestions – Case B

Remarks or suggestions
Remark about process model: should start with «check approval» because this is typically most critical and – if university is not approved – quickly leads to closing the case (with rejection)
Participant states that she has created her form for capturing criteria that need to be discussed in interview and that later allow to trace and justify decisions (e.g. about semester fees)

Table 4.3 lists the activities concerning case "B" that has been performed by the process participants including the problems and solution that might have been identified, resources and certain observations or remarks during the execution of the process. In the appendix of this documents further potential cases are described that has been used to evaluate the ICEBERG-PE approach (see Appendix-A: Case Data).

In addition to the potential case data, Table 4.4 lists the data and documents that were submitted by the applicants. This case data can be regarded as data that was available during workflow initialization.

Table 4.3: Case Data – Case B

Activity	*Role*	*Problem*	*Solution*	*Resources*	*Observations, remarks*
1. Analyse application documents	Study assistant				
2. Check approval of qualification/accreditation of university	Study assistant	University not in anabin list	Call Swiss ENIC Ask student for proof of accreditation		
3. Prepare acceptance	Study assistant				
4. Determine tuition fee	Administration	Workflow proposes 7500 TF. How to tell the system that 700 would be correct?	Enhance the flexibility of the system. Extend desc. of task.		
5. Accept application formally	Study assistant			Acceptance letter template	Interviewee always sends a letter to students that confirms receipt of their application documents; then each case is double-checked with the dean before sending acceptance letter

Table 4.4: Data and Documents from Applicant

Personal Details	***Academic Qualifications***	***Secondary School***	***Professional Experience***	***Documents***
Gender	BSc Degree (Name)	School name	Function/Role	Letter of Motivation
Last and First Name(s)	University/institution (BSc)	Place/Country	Enterprise/Organisation	BSc Degree
Street and Number	Place/Country (BSc)	Qualification obtained	Place/Country (Professional Experience)	Transcript of Records for BSc Degree
Post Code and Town	Matriculation No	Date of Award (Secondary School)	Duration (Professional Experience)	Documentary evidence of required level in English
State/Canton	BSc Degree Award Date	Place of Residence upon Graduation	Research Experience	Copy of Passport/Residence Permit
Country	Grade BSc-Thesis	Post code and Town (Place of Residence upon Graduation)		Copy of Secondary School Education
Date of Birth	Average Grade (BSc)	Canton/State (Place of Residence upon Graduation)		Curriculum vitae
Place of Birth	MSc Degree	Place/Country (Place of Residence upon Graduation)		Photograph
Country of Birth	University/institution (MSc)			Certificate of employment/proof of work
Mother Tongue language	Place/Country (MSc)			References
Marital status	MSc Degree Award Date			
Nationality	Grade (MSc)			
AHV No	Additional Qualifications			
Phone	Linguistic Abilities (certificate)			
E-mail				

4.4 Scenario and Data Analysis

As mentioned in Section 4.2, the process begins by checking an application formally in two knowledge intensive tasks prepare eligibility check executed by the study assistant and the eligibility check itself executed by the dean of the programme. This formally check results in a pre-selection of whether a candidate will be invited for interview or not. This (prepare) eligibility check phase consists of knowledge intensive tasks where knowledge workers try to learn and perform these tasks using prior experiences which are possibly available.

Up to now the admission process has been performed using conventional infrastructures such as paper based dossier including hard-covered case files, a computer based file system, a university administration application and a standard content management system. This means that the application case data is physically and electronically available, without a significant number of metadata items. Some structured data is usually transferred to the university administration application after completion of the application. However, the prior case data can be accessed using full-text search using conventional information retrieval techniques only. As a result, the experience management is performed by the knowledge worker itself by gaining experiences while performing one case after the other – there is no experience management software implemented up to now. Nevertheless, the knowledge workers are performing their file based experience management while capturing their learned procedures using notes added to a case. However, the admission process has been in place since the master program began and has been improved upon each year. Nevertheless, the stakeholders of the admission process expect that the implementation of an experience management software would improve the effectivity and efficiency of the admission process.

4.4.1 Variety of Cases

Since the Master of Science in Business Information Systems (MSc BIS) is recognised as an international programme with students from all over the world, it is an inviolable principle that all applications are treated as equal as possible, regardless if it is a foreign or a local application.

The study administration, the dean of the programme and the admission commission, is exposed to cases with a different composition. Although the variety of the cases is enormous, it is possible to identify similarities when comparing certain sub-elements of certain cases or when abstracting certain elements. This variety becomes evident when analysing the randomly selected sample of 66 cases of previous applications to the MSc BIS programme (see Figure 4.5):

- Sample size: 66 cases
- Applicant's nationalities: 30 countries
- Bachelor degree subjects: 33 different subjects
- Bachelor degree generalised subjects: 25 different subjects
- Bachelor degree institutions: 48 universities
- Bachelor degree institutions located: 26 countries
- Working experience: 64 different jobs or internships

The sample of 66 cases shows that the variety of the cases in this sample is significant. The applicants have 30 different nationalities, and they denoted 33 different bachelor’s degree (for confidential reasons only 25 generalised degree subjects are listed in Figure 4.5) received from 48 different universities located in 26 different countries. Besides, the variety of the working experience is almost the same the sample size of the cases itself. Unfortunately, it was not possible to list the denotations of the working experience due to confidential reasons. Based on a deeper analysis considering a vast number of applications gathered over the recent years, it is possible to process difficult cases in a more efficient way using the content of previous similar cases.

Nationality (30)

- Afghanistan
- Australia
- Austria
- Bangladesh
- Brazil
- Canada
- Costa Rica
- Croatia
- Ecuador
- Finland
- Germany
- Ghana
- Guatemala
- India
- Kosovo
- Lebanon
- Nepal
- Nigeria
- Pakistan
- Philippines
- Poland
- Russia
- Serbian
- Slovakia
- Spain
- Sri Lanka
- Switzerland
- Turkey
- Tanzania
- Vietnam

Bachelor Subject (25)

- Administration and Marketing
- Business Administration
- Business Economics
- Business Engineering
- Business Information Systems
- Commerce and International Business
- Computer Science
- Economics
- Electrical Engineering
- Electronics and Communications
- Finance and Banking
- Food and Health
- Global Business Management
- Industrial Physics and Electronics
- Informatics Engineer
- Information Science
- International Management
- Journalism
- Marketing
- Mathematics Engineering
- Mechatronic Engineer
- Political Sciences
- Project Management and IT
- Public Relations
- Tourism

University (48)

- AGH University of Science and Technology
- Al-Hikmah University
- AMA Computer College
- Baden-Wuerttemberg Cooperative State University
- Bern University of Applied Sciences
- BZU Multan
- City University of Seattle
- Copenhagen Business School
- Covenant University
- Federal Polytechnic, Ilaro
- FHS St. Gallen
- Geneva School of Business Administration
- HES-SO Valais
- Igbinedion University,Okada
- Istanbul Technical University
- Jagannath University
- Kalaidos Fachhochschule
- Kurukshetra University
- Ladoke Akintola University of Technology
- Lucerne University of Applied Sciences and Arts
- Martin-Luther-University Halle-Wittenberg
- Oxford Brookes University
- Punjab Technical University
- Purbanchal University
- Robert Gordon University Aberdeen
- Sagesse University
- South East European University
- Swiss Management Center University
- Turku University of Applied Sciences
- Universidad Nacional de Costa Rica
- University of Aberdeen
- University of Applied Science Vienna
- University of Applied Sciences and Arts Northwestern Switzerland
- University of Berne
- University of British Columbia
- University of Economics Ho Chi Minh City
- University of Georgia
- University of London
- University of Swat
- University of the Valley of Guatemala
- University of Wollongong in Dubai
- Utah Valley University
- Uttar Pradesh Technical University
- Vidyasagar University
- Visvesvaraya Technological University
- Voronezh State University
- Zagreb University of Applied Science
- Zurich University of Applied Sciences

University Country (26)

- Austria
- Bangladesh
- Canada
- Croatia
- Costa Rica
- Denmark
- Finland
- Georgia
- Germany
- Germany
- Guatemala
- India
- Lebanon
- Macedonia
- Nepal
- Nigeria
- Pakistan
- Philippines
- Poland
- Russia
- Switzerland
- Turkey
- United Arab Emirates
- United Kingdom
- USA
- Vietnam

Figure 4.5: Variety of 66 Randomly Selected Admission Cases

The case content consists of documents and data provided by the applicant as listed in Table 4.4. This case content is the primary source for processing a new application. Primary elements are university certificates including a transcript of records, certificate of employment and further documents serving as proof of the authenticity of the provided information. Apart from that, the knowledge workers of the admission process (study assistant, Dean of the programme, interviewees and commission members) are already implementing a data file based experience management system. These data files contain experience items such as the problem and solution items including procedural information as showed in Table 4.3. Although there is not electronic experience management system implemented and linked to a workflow management system, it is possible to process demanding cases as shown in the next sub-section.

4.4.2 Case-based Inferencing

As mentioned in Section 4.2, the aim of the admission process and the initial tasks is to verify whether an applicant fulfils the basic admission criteria. These criteria are working experience, good linguistic abilities in English and an acceptable academic qualification, which includes a bachelor's degree in a related field of study obtained from an accredited institution and good or excellent grades. A number of elements of the basic requirements are compensable, while others have to be fulfilled. For instance, it is not possible to be enrolled in the programme without the required linguistic abilities in English, since the programme is entirely in English. The bachelor's degrees and work experience are rated on a scale signifying "excellent", "better than average", "average", "below average" and "not acceptable". Using this scale it is possible to compensate a slight subject mismatch of the bachelor's degree with a subject match in the work experience if the duration of the job is significant. Such an analysis and compensation are accomplished during an instantiation of the admission process. Evaluation and preparations concerning a valid bachelor's degree and acceptable grades are done at the beginning of the admission process during the eligibility check tasks, as described in Section 4.2. This verification of the academic qualification encompasses a rating of the institution, a conversion of the grade(s) and comparisons of the curriculum's content.

- **Demanding Example**: As revealed by the application sample analysis (Figure 4.5), the study assistant was confronted with a wide variety of different academic institutions located all over the world. If a new application arrives, the study assistant must verify the accreditation of the institution where the candidate obtained his or her bachelor's degree. The

> standard procedure is to consult the anabin database, which contains ratings of institutions. If the new application contains a university which is not listed in anabin, the study assistant needs to discover the level of accreditation based on an alternate path. If the institution of a new application was previously verified by the study assistant, he or she should be able to retrieve a previous case containing the result of the previous analysis. Assuming that the new application contains an institution that has not been verified and is not listed in any database, it would be helpful if certain characteristics can be used to find similar cases. An application case from an applicant holding a degree from the "ABC University" located in the USA might contain similarities to case "B", as presented in Table 4.1. It may be possible that the presented procedural description in case "B" can also be applied when assessing the current case.

The mentioned example indicated that it is possible to learn from previous cases, which can later be adapted to future situations. The scenario and data revealed that the retrieval and adaptation of previous cases can be done directly or using case-based inferences. This case-based inferencing can be considered analogical and similarity-based inferencing using the underlying "[…] assumption that when a new situation 2 is similar to an old situation 1 then we can plausibly predict that an outcome 2 similar to outcome 1 is correct" (Plaza, 2009, p.18).

4.5 Problem Relevance and Objectives

Through an analysis of cases of the application scenario described within the previous section and based on interviews with the stakeholder of the application scenario, it was possible to derive generalised objectives for the approach and the modelling of the case content and characterisation.

The application scenario indicated that such a process can be supported by a particular way of process execution, business process management or process modelling. In this context, CMMN is more expressive than BPMN for modelling non-structure process parts. For example, activities are executed because of human judgement and activities dependent upon the case situation can be distinguished between. New instances of a process might lead to unforeseen situations that require new activities, which cannot be modelled in advance. Therefore, an adaptive approach is necessary.

A case-based reasoning system can provide the required functionalities based on the case-based reasoning cycle: retrieve, reuse, revive and retain. The application can be regarded as a case. The ad-hoc process can be regarded as a case model. If a case requires additional activities, these can be added to the case model. The objectives for case content and case characterisation were determined from the application's scenario and are summarised in the following paragraphs.

As a hypothesis, the case content is a representation of the process that is executed in order to deal with a certain situation (a new application of a potential student). In the terminology of CBR, the content can also be referred to as the solution or lesson space (Bergmann, 2002). Based on the admission process application scenario, the following high-level objectives were defined for the case content:

- Maintenance: The case content should reflect an update functionality in order to capture ad-hoc knowledge from previous situations.
- Resources: The case content should contain information resources such as documents and data objects.
- Representation: The case content should be presented in an adequate way. Often this is referred to as a graphical representation using a graphical modelling language. However, this is highly debatable in both practice and research, if a graphical representation is required at all.

The case characterisation enriched the case model with additional information which served as a processable basis for a similarity measure. Based on the admission process application scenario, the following objectives for the case characterisation were derived:

- Structure: The cases need to be described in a structured way.
- Reusability: The case characterisation should be described with a reusable vocabulary, which can be provided in an enterprise ontology.
- Process execution information: To assign tasks to appropriate performers, the case characterisation should include process information such as variables or roles.

In essence, based on the admission process application scenario, *there is a need for a certain process execution flexibility, information availability and access to previous cases containing case meta-data as well as information about performed activities, decisions and the decisions paths*.

It is not claimed that the objectives derived from the admission process application scenario are complete at all. As an example and as previously mentioned, it is highly debatable in both practice and research, whether a graphical representation of a model is even required.

An approach was suggested based on this application scenario, which will be discussed further in subsequent sections and if necessary, more accurate objectives are listed for the specified chapters answering the research questions.

5 Ontology-based CBR and Process Execution Approach[13]

This chapter concerns the main conceptual artefact of this thesis - the suggested approach. Simultaneously, the main research question guides the investigations as well as the suggestion in this chapter. As a result, the following research question can be answered conceptually:

RQ 1: How can case-based reasoning be integrated with process execution?

This research question is answered conceptually within this chapter after introducing related work and devising an approach and a methodology.

5.1 Introduction

Retrieving and maintaining existing knowledge and experience is an important aspect for different entities. This is especially the case when knowledge-intensive and agile activities occur, as described in the admission process application scenario. A powerful experience management system proved that it is possible

[13] Some verbatim passages presented in this following section, which have been achieved in this thesis project, have been published in the following own authored publication (Martin et al., 2016).

to reuse experiential knowledge and discover past decisions, which can be crucial for a process' success. Case-based reasoning (CBR) can be an adequate method for retrieving experiential knowledge in an experience management system (Bergmann, 2002).

The admission process application scenario (see Chapter 4) indicated that the accessibility of previous cases containing experiential knowledge is crucial for an entity such as described within the scenario. Moreover, the application scenario revealed that different individuals performing different roles have different perceptions of previous cases in a case-based research repository. We derived the requirement by presenting relevant information according to the varying concerns of different individuals as well as within the analysis of the order process in a software company (Martin et al., 2016).

Based on the analysis of the two application scenarios a case-based reasoning approach (called ICEBERG) was developed together with a demonstrator, which was applied at a software company (see Martin, Emmenegger and Wilke (2013), Witschel et al. (2015) and Martin et al. (2016)). The acronym ICEBERG has been derived from **i**nterlinked **c**as**e**-**b**as**e**d **r**easonin**g** similar to the commonly used metaphor in applied psychology, pedagogic and interpersonal communication. The metaphor has been described by Ernest Hemingway (1932) and reflects the Pareto principle (80/20). Sigmund Freud also used the iceberg metaphor to explain the differences between conscious and unconscious human action (Ruch and Zimbardo, 1974). Within this thesis project, however, the metaphor was used to describe the notion and goal of the approach: Using interlinked (ontology-based) case-based reasoning to bring hidden knowledge to the surface for supporting process execution.

The ICEBERG approach was introduced together with an extension for process execution called ICEBERG-PE, which consists of the four following elements as depicted in Figure 5.1:

- The **case-repository** is a central feature of the ICEBERG-PE approach and it contains retained and learned cases.
- The **case-based reasoning services** provide the automatic retrieval of previous cases as well as semi-automatic re-use and adaptation of the previous cases to the current situation using rules, manual revision of new cases by the users. It also provides the automatic retention of cases with regard to adding the case to the case repository.
- The **ontology** is used for CBR configuration, storage of the enterprise and domain ontology and is simultaneously the vocabulary of the CBR approach.
- The **process execution** element is the instantiation of a business process based workflow engine running a workflow definition.

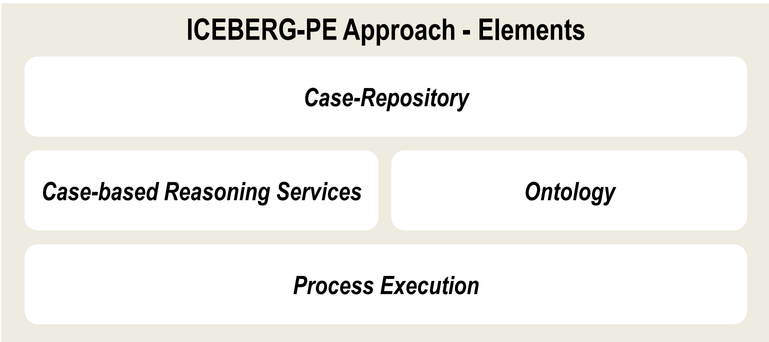

Figure 5.1: High-level View of the ICEBERG-PE Approach

In the following section, related work concerning enterprise ontologies and ontology-based CBR is presented. In the subsequent sections the four elements, as depicted in Figure 5.1, are introduced. The ontology element is introduced within this chapter and, as part of the case model description, in Chapter 6 as well. The case-based reasoning services are described in detail in Chapter 7. The case repository and the process execution will be displayed as part of the demonstrator in Chapter 8.

5.2 Related Work

The ICEBERG-PE approach relies on structural CBR, which uses existing knowledge (sometimes called background, contextual or domain knowledge) as vocabulary to describe a case (Richter, 1998). In this section, enterprise architecture is introduced that can be used as a source of background.

5.2.1 Enterprise Architecture

Enterprise architectures contain relevant aspects of an enterprise-business structures, IT structures and their relationships (Ross, Weill and Robertson, 2006; Lankhorst, 2009). Lankhorst (2009, p.3) has defined an enterprise architecture as "[…] a coherent whole of principles, methods and models that are used in the design and realisation of an enterprise's organisational structure, business processes, information systems, and infrastructure".

Using the ISO/IEC/IEEE 42010 (2011) conceptual model it is possible to distinguish between architecture descriptions, frameworks and description languages:

- Architecture description (AD): "An architecture description (AD) is an artifact that expresses an architecture. Architects and other system stakeholders use Architecture Descriptions to understand, analyze and compare Architectures" (ISO/IEC/IEEE, 2011). An AD is what is written down as a concrete work product. It could be a document, a model repository or a collection of artefacts. An enterprise-specific architecture description (EAD) can act as a source of background knowledge in this approach.
- Architecture framework (AF): "An architecture framework establishes a common practice for creating, interpreting, analysing and using architecture descriptions within a particular domain of application or stakeholder community" (ISO/IEC/IEEE, 2011). Enterprise architecture

frameworks (EAF) guide and support the creation and interpretation of a concrete EAD. Matthes (2011) has described more than fifty enterprise EAFs. One prominent example is the Zachman Framework (Zachman, 1987, 2008), which can be seen as a schema and a classification containing a "[…] total set of descriptive representations relevant for describing an enterprise" (Zachman, 2008, p.1). Another example of such an EAF is The Open Group Architecture Framework (TOGAF) (The Open Group, 2009b).

- Architecture description language (ADL): "An ADL is any form of expression for use in architecture descriptions" (ISO/IEC/IEEE, 2011). An example of an enterprise ADL is ArchiMate (Lankhorst, Proper and Jonkers, 2009; The Open Group, 2009a, 2012). There are also description languages for specific parts of an enterprise architecture such as BPMN (OMG, 2011) for business process modelling or BMM (OMG, 2008) for business motivation modelling.

Enterprise ADs are a valuable source of background knowledge for the enterprise. Nevertheless, an enterprise AD needs to be created and maintained. If an enterprise decides to omit an EAF, selected enterprise models such as business process models or organisation models can be considered alternative sources for background knowledge. Enterprise models contain valuable knowledge concerning the enterprise itself and provide adequate representations for different stakeholders.

5.2.2 Enterprise Ontologies

As described in Section 2.2.1.3, in CBR the cases are described using a specific vocabulary (Richter, 1998). This vocabulary can be derived from existing knowledge sources (sometimes called background, contextual or domain knowledge). "Vocabularies define the concepts and relationships (also referred to as 'terms') used to describe and represent an area of concern" (W3C, 2012b).

Based on the original definition of Gruber (1993), Studer, Benjamins and Fensel (1998, p.184) an ontology is "[…] a formal, explicit specification of a shared conceptualisation". Also, "there is no clear division between what is referred to as vocabularies and ontologies" (W3C, 2012b). However, "the trend is to use the word ontology for more complex, and possibly quite a formal collection of terms […]" (W3C, 2012b).

Bergmann and Schaaf (2003) have investigated the relation between ontology-based approaches for knowledge management and structural CBR. They have concluded that a structural CBR vocabulary is quite similar to an ontology with regard to knowledge management. "Both are formal models for restricting the possible interpretations of metadata annotations thereby providing the necessary background knowledge for semantic-based access to knowledge items" (Bergmann and Schaaf, 2003, p.621).

Within the enterprise domain, several enterprise ontologies were introduced in order to describe enterprise models or architectures. "The main purpose of an enterprise ontology is to promote the common understanding between people across enterprises, as well as to serve as a communication medium between people and applications, and between different applications" (Leppänen, 2007, p.273). The Toronto Virtual Enterprise (TOVE) project introduced two foundational (activity and resource) and numerous business (organization, product and requirements, quality and activity-based costing) ontologies (Fox, Barbuceanu and Gruninger, 1996; Fox, Barbuceanu, Gruninger and Lin, 1998). Other prominent ontologies include the enterprise ontology (Uschold, King, Moralee and Zorgios, 1998), the organisational memory (Abecker et al., 1998) and the context-based enterprise ontology (Leppänen, 2007).

5.2.2.1 ArchiMEO

As mentioned in Section 5.2.1, enterprise architecture descriptions are valuable sources of enterprise knowledge that can be used in CBR. To use the knowledge of enterprise ontologies, architectures and models, Kang et al. (2010), Hinkelmann, Merelli, and Thönssen (2010), Thönssen (2010) and Hinkelmann et al. (2015) have all suggested interlinking or relating enterprise ontologies with enterprise architectures and making these ontologies available in a way that is machine-readable.

The enterprise ontology ArchiMEO[14] was developed by the University of Applied Sciences and Arts Northwestern Switzerland FHNW. ArchiMEO is an enterprise ontology based on the ArchiMate, an enterprise architecture modeling language developed by The Open Group (Lankhorst, Proper and Jonkers, 2009; The Open Group, 2009a, 2012) and the Zachman Framework (Zachman, 1987, 2008), and is extended with selected concepts from other enterprise ontologies: TOVE (Fox, Barbuceanu and Gruninger, 1996; Fox et al., 1998), 'the enterprise ontology' (Uschold et al., 1998) and the context-based enterprise ontology (Leppänen, 2007). The foundation of the ArchiMEO ontology was laid during the development of the linked enterprise models and objects (LEMO) approach by Martin (2010) and Brun (2010). ArchiMEO has been successfully applied in risk management (Emmenegger, Laurenzni and Thönssen, 2012), contract management (Thönssen and Lutz, 2012) and CBR (Martin, Emmenegger and Wilke, 2013).

5.2.3 Ontology-based CBR and Conclusion

Chapter 2.2.5.1 introduced ontology-based CBR and listed the most prominent approaches and application domains. These approaches, frameworks and tools emphasised the potential of accessing and utilising an ontology-based repository. This literature chapter indicated that CBR itself can benefit from ontology-based vocabularies. Specifically, Potes Ruiz, Kamsu-Foguem, and Noyes (2013) as wee

[14] ArchiMEO is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License and available for download: http://www.ikm-group.ch/archimeo

as Kamsu-Foguem and Noyes (2013) have expressed that an ontology-based CBR approach can support the sharing of lessons learned by collaborative experts.

As described within this section, enterprise ontologies can benefit from enterprise architecture and models. Therefore, with regard to business applications, the use of an enterprise ontology in an ontology-based CBR approach can be regarded as the next logical step. Unfortunately, no significant attention has thus far been paid to ontology-based CBR concerning the inclusion of enterprise ontologies, the reflection of the different viewpoints and concerns of the stakeholders and the support of process execution. For this reason, a new CBR approach is introduced within the following section.

5.3 Integrated CBR Approach

As previously mentioned, CBR is a common methodology used by humans and systems to solve problems. The literature review in the Chapter 2 and the related work in the Section 5.2 has revealed that CBR methodology implemented in in information systems can successfully be applied in business and is an adequate methodology for experience management (Bergmann, 2002; Bergmann et al., 2003). During the analysis of the admission process application scenario, it was observed that individuals tend to apply a certain CBR method implicitly while performing knowledge-work oriented tasks. Therefore, it is reasonable to conclude that individuals are already familiar with this type of problem-solving method.

As mentioned within the theoretical framework (see Section 2.2), CBR relies upon knowledge gathered in advance or based on previous cyclic iterations. This knowledge is stored in a CBR system using a certain case structure in a case repository, which is also referred to as a case base. Bergmann (2002) has regarded structural CBR as best suited for experience management and knowledge-intensive tasks and as previously mentioned, structural CBR can benefit from ontology-based knowledge representation. Bergmann and Schaaf (2003) have suggested a "knowledge container" to represent background and domain knowledge. Approaches that do not consider background knowledge "[…] are often isolated and closed in the sense that they are not developed with respect to cooperation with other systems. […] Most [ontology-based CBR] systems rely on proprietary, sometimes even XML compliant, languages for the vocabulary and the cases but do not facilitate the exchange of knowledge" (Bergmann and Schaaf, 2003, p.622).

Consequently, this work suggests a structural ontology-based CBR approach for exploiting the full potential of an existing enterprise architecture and ontology, and this enables the utilization of existing domain knowledge during the execution of the CBR cycle.

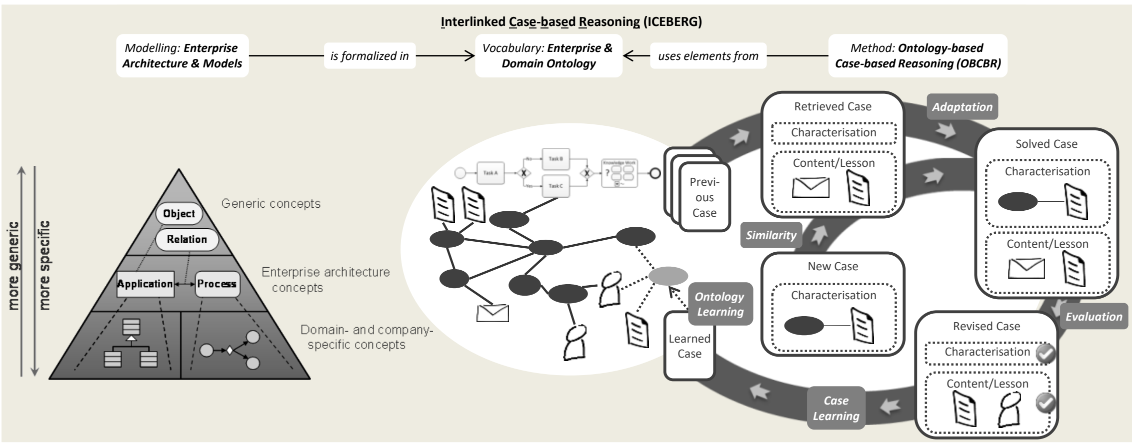

Figure 5.2: Interlinked CBR Approach

The ICEBERG approach developed in this thesis used the same technology (i.e. ontologies) for both the case repository and the knowledge container. It used an enterprise ontology ArchiMEO for its purposes, which was based on W3C standards and recommendations. The Figure 5.2 shows the interlinked CBR approach and lists its three main elements:

1. Enterprise architecture and enterprise models resulted from modelling.
2. Enterprise and domain ontology used as vocabulary.
3. Ontology-based case-based reasoning as an underlying method.

As mentioned in Chapter 5.2.1, (enterprise) architecture frameworks are commonly used to model enterprise-specific elements and integrate existing models into a coherent architectural description. ArchiMEO is a formalisation of such an architectural description. Since ArchiMEO interlinks enterprise ontologies with enterprise architectures and makes it possible to formalise descriptions of enterprise architectures or models, formalised enterprise architectures can deliver existing domain knowledge that improves CBR. The ontology-based CBR system can then use these formalised descriptions of enterprise architectures or models. An enterprise architecture and further modelled enterprise models are formalised in an enterprise and domain ontology based on ArchiMEO. This ontology serves as a vocabulary for the ontology-based CBR system. Respectively, the ontology-based CBR method uses elements from the ontology.

5.4 Ontology Framework

This ICEBERG-PE approach utilises an ontology which relies on the following structure, as depicted in Figure 5.3. As previously stated, there is no clear differentiation between the term vocabulary and ontology (W3C, 2012b), which is also valid within the field of CBR where the "[…] coincidence of [a CBR] vocabulary and an ontology […]" can be observed (Bergmann and Schaaf, 2003, p.622). The ICEBERG-PE approach follows the work of Bergmann and Schaaf (2003, p.622): "Neglecting the fact that an ontology typically serves many purposes one can say that a [CBR] vocabulary is an ontology of the domain of discourse underlying the [CBR] application".

The ontology structure in Figure 5.3 shows the dependencies (imports) of the corresponding ontologies which build, as a result, one ICEBERG-PE ontology instantiation. The ICEBERG-PE ontology is an extension of the ArchiMEO ontology, which consists of concepts defined in ArchiMate (The Open Group, 2012) and ISO/IEC/IEEE (2011) 42010 enhanced with concepts from other enterprise ontologies. The ICEBERG-PE ontologies were formalised in RDFSPlus (Allemang and Hendler, 2008) which is expressed in RDF(S) (W3C, 2014a; c) and extended with certain resources within the OWL namespace (W3C, 2012a). Additionally, it is serialised using the RDF 1.1 Turtle (W3C, 2014b) format.

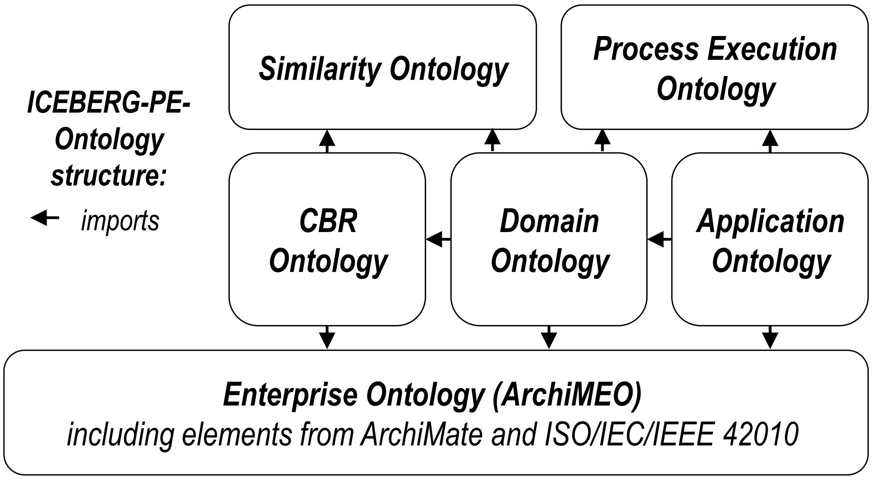

Figure 5.3: Ontology Structure of the ICEBERG-PE Approach

As depicted in Figure 5.3, the following three ontologies were built while reusing or importing concepts from ArchiMEO.

- The CBR ontology contains concepts for configuring the retrieval and the CBR system itself.
- The similarity ontology contains concepts for creating a similarity model, according to structural CBR approaches.
- The domain ontology contains enterprise-specific domain knowledge of an application scenario or use case.
- The process execution ontology contains the required concepts for interacting with a process execution engine and storing workflow relevant data.
- The application ontology specialises the domain ontology with respect to an enterprise idiosyncrasy (van Heijst, Schreiber and Wielinga, 1997).

Figure 5.4 shows the case viewpoint model (Martin et al., 2016) which we developed for the order process application scenario (see Section 3.4.2.1). This model served the need for different views and viewpoints. The case viewpoint model was an essential part of this research work since it allows for conceptual interlinkage to process execution. It also answers research question 1, partially answers research question 2.2 and, in conjunction with the ICEBERG-PE ontology structure, answers research question 3.2. The link between the process execution and the knowledge worker's integration into the working environment is described in Section 5.5.

In addition to the mentioned technical integration, it was recognised that different stakeholders have different concerns and therefore, have diverse interests regarding experienced case knowledge (Martin et al., 2016). It was also recognised that a single case characterisation with similarity measures and the corresponding weights cannot support all of the knowledge workers' different interests.

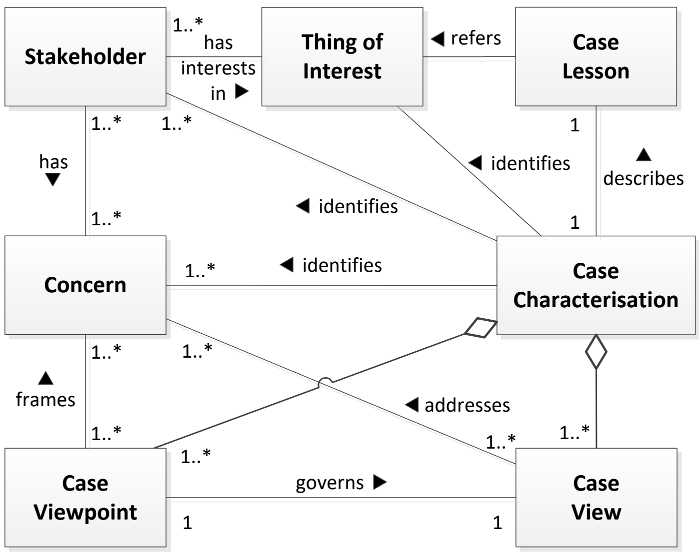

Figure 5.4: Case Viewpoint Model (Martin et al., 2016)

As emphasised by The Open Group (2012), the problem of how to satisfy the needs of different stakeholders has been researched since the 1990s. According to ISO/IEC/IEEE 42010, an architecture viewpoint represents the interest of one or more stakeholders, whereas a view expresses the "[…] Architecture of the System of Interest from the perspective of one or more stakeholders to address specific Concerns […]" (ISO/IEC/IEEE, 2011). The notion of views was introduced by the CIMOSA EAF (ESPRIT Consortium AMICE, 1989). ArchiMate (The Open Group, 2012, p.74) has defined a view "as a part of an architecture description that addresses a set of related concerns and is addressed to a set of stakeholders. A view is specified by means of viewpoint […] Simply put, a view is what you see and a viewpoint is where you are looking from". The case viewpoint model of ICEBERG (Martin et al., 2016) consists of the following elements, which were derived from ISO/IEC/IEEE (2011):

- Thing of interest: The thing of interest is something in which a stakeholder has interest and is described as part of a case lesson or case characterisation. It can be a system, experiential knowledge or information need. This term is used as a placeholder.
- Case lesson/content: The case lesson, also called case content, "[…] can contain information that is not the solution itself but useful to find a solution […]" (Bergmann, 2002, p.50), including links to information sources containing previous solutions that are useful to find a solution for the current situation.
- Case characterisation: The case characterisation "[…] describes all facts about the experience that are relevant for deciding whether the experience can be reused in a certain situation" (Bergmann, 2002, p.50).
- Stakeholder: "Stakeholders are individuals, groups or organisations holding concerns for the [thing of interest]" (ISO/IEC/IEEE, 2011).
- Concern: A concern is any interest in the experiential knowledge system or thing of interest.
- Case viewpoint: A case viewpoint is "[…] a set of conventions for interpreting, using and analysing one type of […]"case view and is

derived from an architectural viewpoint (ISO/IEC/IEEE, 2011). A case viewpoint frames a specific set of concerns.

- Case view: A case view expresses the case (including the lesson) of the thing of interest "[…] from the perspective of one or more stakeholders in order to address specific concerns using the conventions established by its viewpoint" (ISO/IEC/IEEE, 2011).

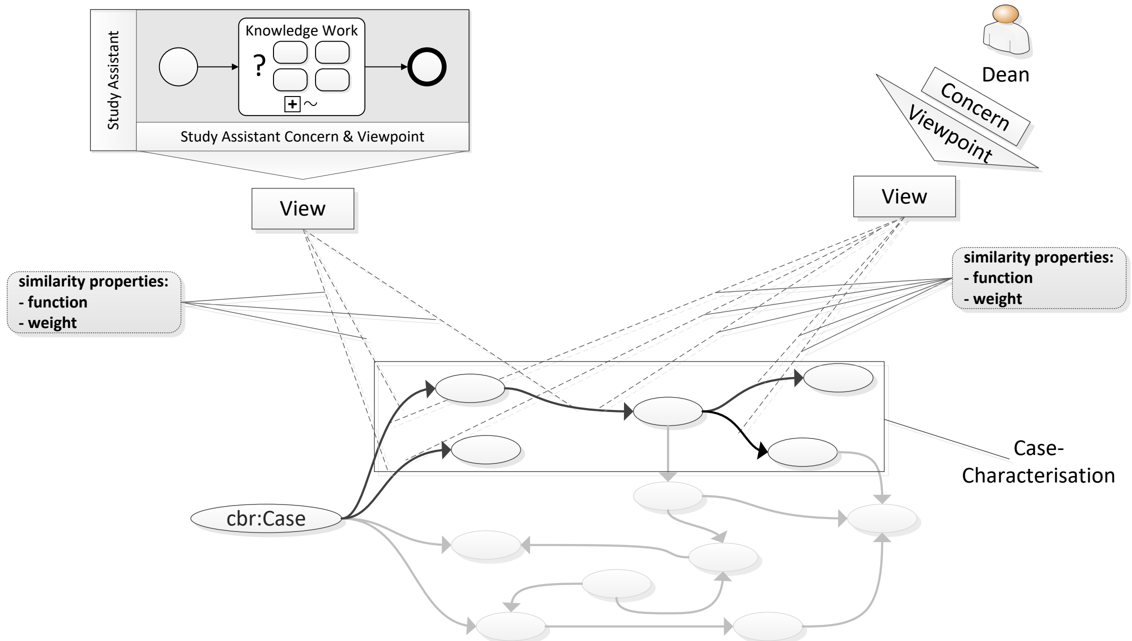

Figure 5.5: Example Views on Case Characterisation (adapted from Martin et al. (2016))

Figure 5.5 depicts an example of different concerns leading to different viewpoints and views on the case characterisations based on the roles (person performing a role or process roles) of various stakeholders. A different view means that part of the case characterisations (classes and relations), the similarity measures (functions) and the weights of the properties (simple and relational ones) differ depending on the corresponding viewpoint and concerns. On the left in Figure 5.5, a snippet of the process is exhibited. If a stakeholder who is performing a role consults the CBR system while performing a user or a manual task, the similarity service will be invoked under the usage of the corresponding view. If invocation happens without direct stakeholder involvement using a service task, then the CBR system will attempt to identify the current context and choose the corresponding view. For retrieval based on a similarity service, a specific characterisation viewpoint must be chosen and only those characterisation elements assigned to the corresponding view should be used to formulate a query statement.

5.5 Conceptual Framework

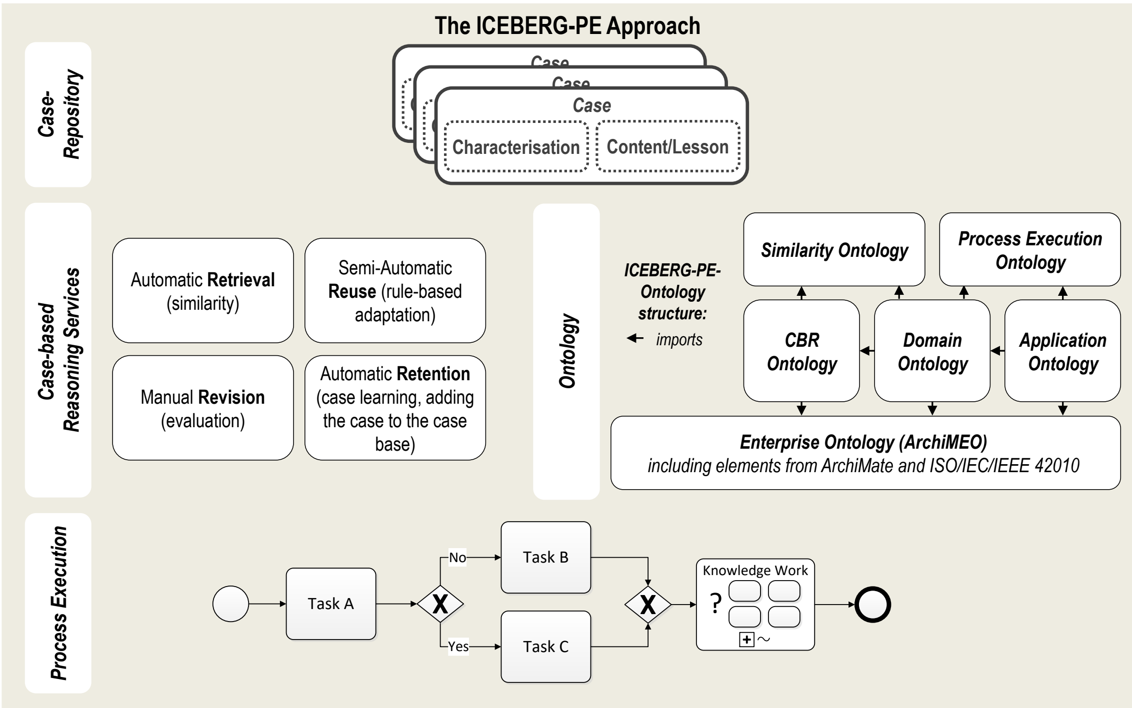

Figure 5.6: Elements of the ICEBERG-PE Approach

This ICEBERG-PE approach consists of four main elements (as depicted in Figure 5.6) which combine CBR and process execution in order to support knowledge work. In the following sections, these four elements will be introduced on a conceptual level. In the corresponding chapters, these elements will be presented on the technical level and as a prototypical implementation.

5.5.1 Ontology

Ontology is the core element of the ICEBERG-PE approach. As previously described within Section 5.4, the ICEBERG-PE ontology was used to define, configure and describe various aspects of the introduced approach itself. The ontology was used for the CBR configuration, storage of the enterprise and domain ontology and was simultaneously the vocabulary of the CBR approach. The ontology element of the approach allowed for semantic correctness, avoided redundancy, ensured accessibility, and allowed the users to orientate themselves based on their enterprise vocabulary and individual perceptions.

5.5.2 Case-Repository and Cases

As a central feature of this work, the CBR approach contained a case base. In this research work, the term *case repository* was used to avoid misunderstandings.

> **Definition 5.1: Case Repository**
> A case repository is an instantiation of the case base concept (Definition 2.8), which contains retained cases.

In contrast to traditional structural or object oriented CBR approaches, the repository in ICEBERG-PE was implemented using a graph database using Semantic Web standards. This made it possible to describe the cases using the vocabulary, which is derived from the ontology framework. This description is referred to as *case characterisation* (see Definition 2.9). It contained features and values which are linked to the ICEBERG-PE ontology.

The *case content*, sometimes called case lesson, captures the lesson learnt, experiential knowledge, process fragments and the related information resources of previous solutions or lessons that are useful for finding a solution for the current situation.

> **Definition 5.2: Case Content**
> A case content/lesson is an instantiation of the experience/knowledge item concept. A case content contains "[…] information that is not the solution itself but useful to find a solution […]" (Bergmann, 2002, p.50) and links to information sources containing previous solutions that are useful to find a solution for the current situation.

The case model is investigated and described in Chapter 6.

5.5.3 Case-based Reasoning Services

The CBR services are the main application logic within this research work. The elements depicted in Figure 5.6 are the four main CBR supporting categories of the ICEBERG-PE services. These supporting categories are related to the CBR-cycle. The ICEBERG-PE services provide automatic *retrieval* of previous cases, semi-automatic *reuse* and adaptation of the previous cases to the current situation using rules, manual *revision* of new cases by the users and automatic *retention* of cases regarding adding the case to the case repository (case base). The ICEBERG-PE services are investigated and described in Chapter 7.

5.5.4 Process Execution

In order to combine a CBR approach with a workflow management system and answer the first research question 1 by an implementation (apart from the conceptual investigation), an interface between these two components was realized. This interface between the workflow management system and the CBR system was built from the user's perspective based on the task management component. A task management component can also be referred to as a task list (OASIS, 2012) or worklist (Wohed et al., 2009). This CBR interface to task management system needs to be integrated into the working environment of the knowledge and/or process worker. In doing so, this approach reflects the following two settings:

1. Stand-alone CBR system: The stand-alone option can be seen as the typical usage of a CBR system, where every user retrieves similar cases after entering search criteria. The user describes the problem, current situation or current task using a case vocabulary as case characteristics. In this setting, the task enactment is then done by the user himself.
2. Invocable CBR system: The invocation option is implemented as a service that is invoked during the execution of a process or workflow. In this setting, the CBR system is interlinked with the task list of the workflow

management system. The ICEBERG-PE component incorporates with the workflow engine and pre-populates the case characterisation using workflow and case-relevant data.

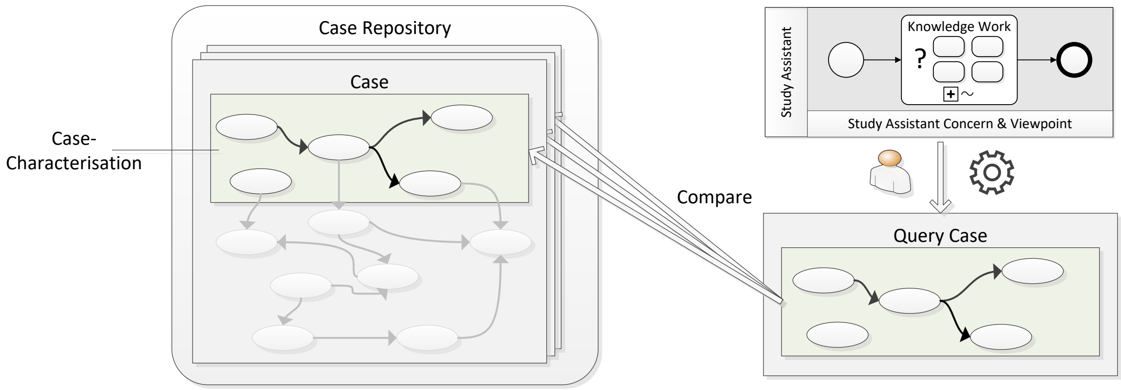

Figure 5.7: Interface to Case-based Reasoning (adapted from Martin et al. (2016))

Figure 5.7 shows how a query case is compared (based on a similarity mechanism and configuration as described in Chapter 7.2) to existing cases based on the ontology-based case characterisation. The query case is either manually or automatically defined depending upon the aforementioned settings.

5.6 Methodology

The following methodology consists of two procedure models depicting how the ICEBERG approach and the process execution extension were instantiated based on an application scenario or use case. Ju et al. (2016) has emphasised the importance of having a clear procedure model with an end-user and expert involvement.

The initial model was called the ICEBERG procedure model and was elaborated upon with three application scenarios: the admission process (see Section 4.2), the offering and project management of a software company (see Section 3.4.2 and 9.4.1) and the process learning of the Learn PAd project (see Section 3.4.2 and 9.4.2). After the first incremental design science research cycle, the generalised procedure model was instantiated, refined and concretized specifically for process execution (see Section 5.6.2).

5.6.1 General Procedure Model

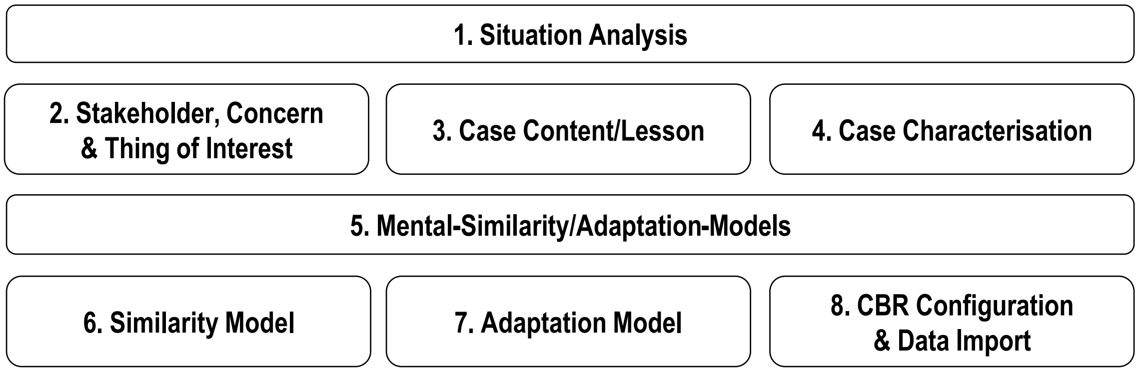

Figure 5.8: Interlinked Case-based Reasoning Procedure Model (adapted from Martin et al. (2016))

This general procedure was elaborated on during the first iteration of this research project. Figure 5.8 presents the general procedure model containing the following eight steps:

1. Situation analysis: In order to fully understand the use case for the CBR approach, a first situation analysis can be performed. Requirement elicitation and creativity techniques are applied. It is advisable that the situation analysis can be done in a focus group or workshop-like setting. It was possible to observe that story-telling can be adequate to achieve impressive results. However, it is advisable that an additional and profound process analysis be elaborated. It is also advisable to consult existing enterprise architecture descriptions, from which further use cases can be derived.

2. Stakeholders, concerns and things of interest: Based on the situation analysis the elements stakeholders, concerns and finally the thing(s) of interest can be derived. This identification will then be used to elaborate the mental models of the different stakeholders in order to derive the view and viewpoint-based similarity model as introduced in Section 5.5.

3. Case lessons/content: Based upon the situation analysis, the case lessons/content can be elaborated together with the stakeholders. It is

advisable that exemplary cases with initial data be created and discussed. Additionally, it is advisable to define how this initial data is acquired or imported.

4. Case characterisation: In conjunction with the case content, a generalised case characterisation is created. At this stage, it is essential that an existing enterprise architecture be used to build an initial case characterisation vocabulary.
5. Mental similarity/Adaptation models: When implementing CBR, it is the ultimate goal to represent mental similarity and adaptation models in a consolidated form as a configurable model as adequate as possible. Before such a configurable model can be implemented, the various mental models need to be elicited and consolidated from the stakeholders. The mental models will be consolidated but are still dependent upon the stakeholders and their concerns.
6. Similarity model: Based on the stakeholder dependent mental models and generalised case characterisation, a CBR expert configures the similarity model dependent upon the views and viewpoints and use an existing enterprise ontology. This configuration is done by determining global and local similarity functions and assigning weights.
7. Adaptation model: In addition to the similarity model, the adaptation behaviour is configured by a CBR expert as well. The ICEBERG approach currently supports manual or semi-manual adaptation rules.
8. CBR configuration and data import: Finally, the entire system is configured by a CBR expert and initial data is imported.

This general procedure model can be applied to all types of projects with CBR involvement with a number of modifications. The flowing section introduced a modification and extension of this general procedure model with respect to process execution.

5.6.2 Procedure Model for Process Execution

The generalised procedure model, as presented within the previous section, was instantiated and concretized for process execution after the first incremental design science research cycle with the usage of the admission process application scenario. The concretized procedure model for process execution reflects, in particular, the process fragment modelling as a required additional step. The research from Sections 6.1, 6.2 and 6.3 indicate that a deeper analysis concerning process fragments and the used modelling language needs to be conducted when instantiating the ICEBERG-PE approach in practice. Therefore, the procedure model was extended for the process execution as follows (see Figure 5.9):

1. Situation analysis: To fully understand the current use case a first situation analysis is performed. Requirement elicitation and creativity techniques are applied. It is advisable that the situation analysis can be done in a focus group or workshop-like settings. Moreover, it is advisable to consult existing enterprise architecture descriptions, from which further use cases can be derived.

2. Stakeholders, concerns and things of interest: Based on the situation analysis, stakeholders, their concerns and finally the thing(s) of interest are derived. This identification will then be used to elaborate the mental models of the different stakeholders to derive finally the view and viewpoint-based similarity model as introduced in Section 5.5.

3. Process Model: After the situation analysis has been performed and the stakeholders are identified, a generalised and overall process model is elaborated. It is not required to end up with a detailed process model that can be implemented in process execution. It should provide an overview and is a basis for a later process execution configuration. Moreover, based on this initial process model, a decision needs to be made regarding which option of CBR process execution (see Section 5.5.4) would be applied.

4. Case content: Based on the situation analysis and process model, the case content can be elaborated together with the stakeholders. It is advisable that exemplary cases with initial data be created and discussed. Additionally, it is advisable to define how this initial data is acquired or imported. This will then guide the process fragment model creation in the next step.

5. Process Fragment Model: Since this approach reflects process execution and procedural knowledge, an initial analysis concerning process fragments need to be done. The process fragment model should be created based on an initial decision on complexity (see Section 6.2) and how the complexity will be assessed (see Section 6.3.1; feature comparison, empirical, theoretical or conceptual evaluations). Finally, a case modelling language (graphical or textual) or a subset of modelling language (see Section 6.3) should be selected or tailored according to the previous complexity analysis, which is used as a process fragment modelling language.

6. Case characterisation: In conjunction with the case content and the process fragment model, a generalised case characterisation is created. At this stage, it is essential that enterprise specific elements such as enterprise models, enterprise specific conceptualization or nomenclature, or even an enterprise architecture can be used to build an initial case characterisation vocabulary, which may lead already to a domain/application ontology (step 8). These enterprise specific elements, which can be used as case characterisation input can be accessible implicitly or explicitly.

7. Mental similarity/Adaptation models: When implementing CBR, it is the ultimate goal to represent mental similarity and adaptation models in a consolidated form as a configurable model as adequate as possible. Before such a configurable model can be implemented, the various mental models need to be elicited and consolidated from the stakeholders. The

mental models are consolidated but are still dependent to the stakeholders and their concerns.

8. Domain/Application Ontology: A specific application and domain ontology is elaborated for a process execution instantiation, which is based on an enterprise ontology including application and domain concepts. This domain/application ontology creation is suggested with stakeholder involvement. It is debatable whether this creation step should be made explicit or the creation of a domain/application ontology will be done seamlessly during the case content and characterisation creation. However, eventually a domain/application ontology needs to be available before the creation of the similarity model (step 9).
9. Similarity model: Based on the stakeholder dependent mental models, generalised case characterisation and the ontology, a CBR expert configures the similarity model dependent upon the views and viewpoints. This configuration is done by determining global and local similarity functions and assigning weights.
10. Adaptation model: Apart from the similarity model, the adaptation behaviour is configured by a CBR expert as well. The ICEBERG-PE approach currently, supports manual or semi-manual adaptation rules.
11. CBR configuration and data import: Subsequently, the CBR system is configured by a CBR expert, and initial data is imported.
12. Process Execution Configuration: Finally, the CBR system is integration with the process execution environment. A process engineer creates an executable and generalised process model if required and a possible process execution services. These process execution services interlink the CBR system with the process execution system.

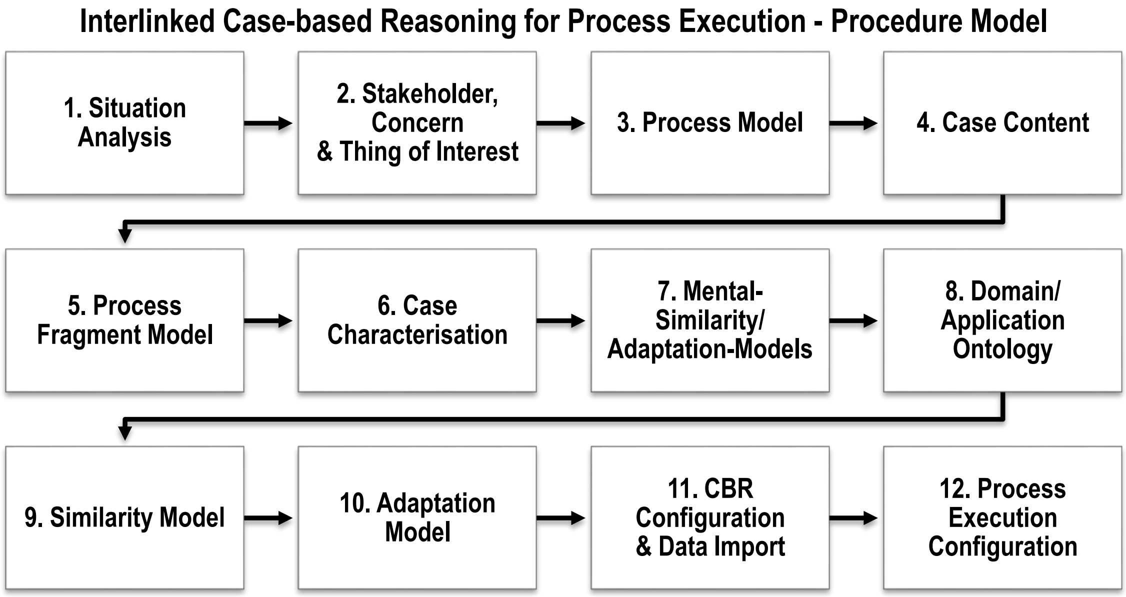

Figure 5.9: Interlinked Case-based Reasoning Procedure Model for Process Execution

5.7 Conclusion

This chapter introduced the overall ICEBERG-PE approach, which was implemented and evaluated in order to answer the overall research question. The approach was derived from related work in conjunction with the admission process application scenario.

The chapter provided a conceptual answer to the following research questions, which is showcased by the implementation of the prototype (Chapter 8) and evaluated by triangulated data sources (Chapter 9).

- The conceptual framework of the approach depicted how case-based reasoning can be integrated with process execution (RQ 1), in conjunction with the ontology framework.
- The ontology framework, referred to as the ICEBERG-PE ontology, provided a partial and conceptual answer for how domain knowledge and contextual information could be used for the retrieval of cases and suggestion or adaptation of case items (RQ 3.2).

- The case viewpoint model, as part of the ontology framework, delineated how the process execution context could be integrated into the case description (RQ 2.2).

The next chapter describes the investigation of the case model and conceptually answers research questions 2, 2.1 and 2.2.

6 The Case Model

The case model represents the concept, which is referred to as an instantiation in the following as the case itself. An instantiated case consists of a case *characterisation* and a case *content* (see Definition 2.9 and Definition 6.1).

> **Definition 6.1: Case Model**
> A case model consists of a description of the case characterisation (Definition 2.9) and the case content (Definition 6.1).

In the following a case characterisation and a potential case content is described that has been elicited using the application scenario. This characterisation has been verified by two stakeholders (process members of the admission process application scenario; see Section 4.2). Apart from that, the case-based reasoning configuration has been created in collaboration with the stakeholders of the application scenario and is based on the ICEBERG case-based reasoning approach introduced here.

This chapter answers the following research questions (see Section 1.4.2):

RQ 2: What should the case description for knowledge-intensive work consist of?

RQ 2.1: How can functional and process knowledge be included in a case description that is cognitively adequate to humans?

RQ 2.2: How can the process execution context be integrated into the case description?

This chapter starts with an investigation concerning process execution context, followed by a description of complexity and cognitive adequacy, then potential case content modelling languages are presented and finally the case model of the ICEBERG-PE approach is introduced.

6.1 Case Description including Process Execution Context

Abecker et al. (1998) defined the concept of context-intensive knowledge supply by differentiating three types of ontologies: enterprise, domain and information ontology:

- The enterprise ontology provides contextual information. It is "[…] used to describe the creation context and the intended utilization context of knowledge items" (Abecker et al., 1998, p.44).
- The domain ontology contains a content description of an enterprise.
- The information ontology provides structure and access to information objects. It "[…] comprises all aspects of information and knowledge sources that are not content-specific" (Abecker et al., 1998, p.44).

As an initial hypothesis, a concise case description is needed to support knowledge-intensive work and to reflect the process execution context.

6.1.1 What is Context?

The following research question refers to the process execution context:

*RQ 2.2: How can the process execution **context** be integrated into the case description?*

The definition of context can be regarded as a difficult undertaking (Thönssen and Wolff, 2012). Ben Mena et al. (2007, p.58) argue that it is not possible to "[…] speak about context in an absolute way". Instead, Mena et al. (2007, p.58) state that context depends closely on the domain and its application nature.

A reason for identifying context and a definition of it has been provided by Dey and Abowed (1999), as follows:

Definition 6.2: Context
"Context is any information that can be used to characterise the situation of an entity" (Dey and Abowd, 1999, p.3; Dey, 2001, p.2).

Such an entity can be everything of concern. Thönssen and Wolff (2012) list several types of context data related to business entities. There are business processes; workflow relevant data; business constraints; and application, resource, social and geographical contexts. They came to the conclusion that an "[…] enterprise architecture provides the biggest overview of the enterprise context data" (Thönssen and Wolff, 2012, p.343). Thönssen and Wolff (2012) further suggest the use of ontologies for modelling context.

Definition 6.3: Enterprise Context
Enterprise context is the conceptualisation of all elements affecting enterprise objects and characterising the situation of them.

6.1.2 What is Process Execution Context?

RQ 2.2: How can the ***process execution*** ***<u>context</u>*** *be integrated into the case description?*

Business process management is primarily focused on the flow. "BPM involves the flow of control and the sequencing of state changes" (Palmer, 2011, p.79), whereas adaptive case management (ACM) (see 2.1.2 Flexibility of Case Management) considers the context when knowledge workers execute knowledge-intensive tasks and thus is context-aware (Palmer, 2011).

According to Palmer (2011, p.82), ACM is the capturing of both the “what” (data, files, records or most often links to the physical sources of those) and the “how” (metadata, audit trail, as well as the context of decisions and actions). "Case management, by its definition, is a system of record of what happened. It captures the context as well as links the information as the case evolves" (Palmer, 2011, p.83). Moreover, it provides a "[…] long-term record of how work is done" (Palmer, 2011, p.86).

The flow of a process, what is predetermined in structured business processes, is sometimes denoted as ***process logic*** (von Halle and Goldberg, 2009; Hinkelmann, 2014). Process logic can refer to knowledge about processes (such as process flow, roles or resources) and is contained in the process model (Hinkelmann, 2016).

The information about how something is done (know how), is denoted as ***business logic***. It can refer to knowledge in processes (skills, experience and know how) at runtime (Hinkelmann, 2016). Apart from that, business logic represents business thinking about the way important business decisions are made (von Halle and Goldberg, 2009, p.6).

Contextual elements need to be reflected by considering additional conceptual modelling methods to include more than only the flow of a process (process logic) in a case description. The process logic explains what has happened. Based on an analysis of contextual information, it is possible to identify how a certain situation has been reached. Business logic can provide contextual information explaining how a certain situation has been reached and frames knowledge, used in process logic, which explains why a certain situation has been reached and why a certain situation exists.

The Workflow Management Coalition (WfMC, 1995) identified three types of data: workflow control data, workflow relevant data (also called case data) and application data. Thönssen and Wolff (2012) considered workflow relevant data as contextual data, and they showed that many process-oriented, case-based reasoning frameworks support the usage of this data type. Workflow relevant data is application data that is accessible from the process execution (also called workflow) engine, and can be created and updated by the engine. In contrast, the application data can only be manipulated by the application itself; even this application is invoked by the execution engine (WfMC, 1995).

Definition 6.4: Process Execution Context

Process execution context is enterprise context that is used by a process execution engine (Definition 2.4) denoted as workflow relevant data and business logic that is used by knowledge workers.

6.1.3 How Can Context be Integrated into the Case Description?[15]

Another aspect of the following research question is the integration of context in the case description:

RQ 2.2: How can the ***process execution context be*** ***integrated*** *into the case description?*

In CBR, a case contains elements of concern (the case content) and a descriptive part (characterisation). For this, existing knowledge (sometimes called background, contextual or domain knowledge) can be used as vocabulary to describe cases (Richter, 1998).

Vocabularies consist of concepts and relationships (also called terms), which can be used to describe "an area of concern" (W3C, 2012b). Such a vocabulary needs to be acquired in advance, and this is a demanding task in time consumption and knowledge worker involvement, which can lead to an expensive situation for enterprises in particular. To overcome this, several studies suggest the utilisation of ontologies (Díaz-Agudo and González-Calero, 2000; Recio-Garía and Díaz-Agudo, 2007; Gao and Deng, 2010) and the use of enterprise architecture descriptions. The underlying idea is the re-use of already existing and agreed-upon conceptualisations of (enterprise) entities using an accepted and machine-processable format that additionally provides reasoning capabilities.

Enterprise ontologies can provide a vocabulary for enterprise-specific case descriptions (Martin, Emmenegger and Wilke, 2013). Moreover, such an

[15] Some verbatim passages presented in this following section, which have been achieved in this thesis project, are published in the following own authored publication: Martin et al. (2016)

enterprise ontology should be rooted on existing enterprise-specific conceptualisation. The alternative, however, would be building such an enterprise ontology from scratch or using an arbitrary, not specified and not accepted enterprise ontology, which could lead to enterprise vocabulary inconsistency.

To overcome this, Kang et al. (2010), Hinkelmann et al. (2010), Thönssen (2010) and Martin et al. (2013) have suggested formalising enterprise architecture descriptions and enterprise models (such as business process models or organisational charts) in an enterprise ontology. An enterprise architecture is a description of an "[…] enterprise's organisational structure, business processes, information systems, and infrastructure" (Lankhorst, 2009, p.3) and is an excellent source of enterprise-specific knowledge to describe cases. Apart from the enterprise architecture description, selected enterprise models are potential sources of enterprise vocabulary and knowledge. Both enterprise architecture and enterprise ontologies allow the reuse of existing vocabulary without storing it redundantly, which avoids possible inconsistencies and additional maintenance effort.

Apart from general enterprise-specific knowledge, which is based on enterprise architecture descriptions, Thönssen and Wolff (2012) suggest the use of an ontology-based representation of context. In the past several context ontologies have been introduced, including SUMO (Niles and Pease, 2001), CONON (Xiao Hang Wang, Da Qing Zhang, Tao Gu and Hung Keng Pung, 2004) and SOUPA (Chen, Perich, Finin and Joshi, 2004).

Therefore, the processes execution context, which is enterprise context (Definition 6.4), can be integrated using information derived from an enterprise ontology and with the usage of a vocabulary that is linked to the same enterprise ontology. The ontology framework of the ICEBERG-PE approach (see Section 5.4) provides such an ontology to include process execution context in an adequate manner.

6.2 Complexity and Cognitive Adequacy

When using cases in a case-based system, the case description should be made explicit and represented in a way that allows machine processing; in parallel, it should remain understandable (cognitively adequate) to humans. The implication is that the case description language should be as natural as possible to gain a wide acceptance from the end user. Therefore, in this section, the cognitive adequacy and potential complexity of cases is investigated with the guidance of the following research question:

*RQ 2.1: How can functional and process knowledge be included in a case description that is **cognitively adequate** to humans?*

The cognitive load theory (see Chandler and Sweller, 1991; Sweller, van Merrienboer and Paas, 1998) is an instructional theory that explains how cognitive resources can be concentrated on the actual learning tasks rather than the preliminaries for the learning task itself.

"The theory assumes a limited capacity working memory that includes partially independent subcomponents to deal with auditory/verbal material and visual/2- or 3-dimensional information as well as an effectively unlimited long-term memory, holding schemas that vary in their degree of automation" (Sweller, van Merrienboer and Paas, 1998, p.251).

The cognitive load theory assumes that the working memory is limited and a working memory overload results in a decrease of learning (schema construction), quality and time consumption of an individual.

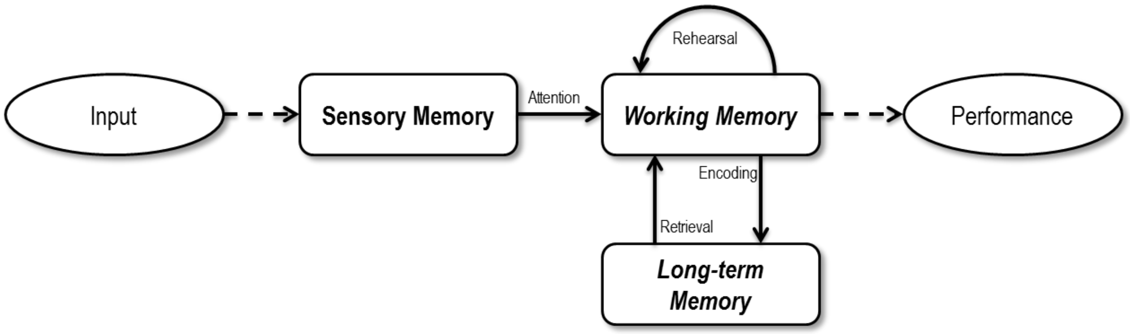

Figure 6.1: Structure of the Memory System (based on Atkinson and Shiffrin, 1968; Claes, Gailly and Poels, 2013)

Figure 6.1 shows the structure of the human memory system based on Atkinson and Shiffrin (1968) and with the terminology of Claes, Gailly and Poels (2013). Based on Atkinson and Shiffrin (1968), the human memory system consists of three structural components:

1. Sensory memory: "Incoming sensory information [(input)] first enters the sensory register [(sensory memory)], where it resides for a very brief period of time, then decays and is lost" (Atkinson and Shiffrin, 1968, p.90). The information will then automatically be transferred to the working memory, based on an attention-based selection process.

2. Working memory: The working memory receives the selected information from the sensory memory. "Information in the short-term store [(working memory)] decays completely and is lost within a period of about 30 seconds, but a control process called rehearsal can maintain a limited amount of information in this store as long as the subject desires" (Atkinson and Shiffrin, 1968, pp.90–91). The information in the working memory will be enhanced and completed with information, which is retrieved from the long-term memory (Claes, Gailly and Poels, 2013).

3. Long-term memory: "The long-term store [(long-term memory)] is a fairly permanent repository for information, information which is transferred [(encoded and copied)] from the short-term store [(working memory)]" (Atkinson and Shiffrin, 1968, p.91).

The working memory has a limited capacity. The limit of the human working memory is estimated by "the magical number" seven, plus or minus two "chunks" of information (Miller, 1956). "The amount of information stored in one chunk depends on the expertise of the subject on the specific task" (Claes, Gailly and Poels, 2013, p.170). This means that experts can access more existing information and already constructed schemas, which leads to more efficient chunking. Experts "[…] can store more information in a single chunk of working memory (one schema of an expert provides access to more information than a novice's schema)" (Claes, Gailly and Poels, 2013, p.170).

To adapt this to the initial research question, a "[…] case description that is cognitively adequate to humans" is a case description that considers the limited capacity of the working memory. This reflection of the limited working memory can be gained by heading towards a cognitive fit as described in the following section.

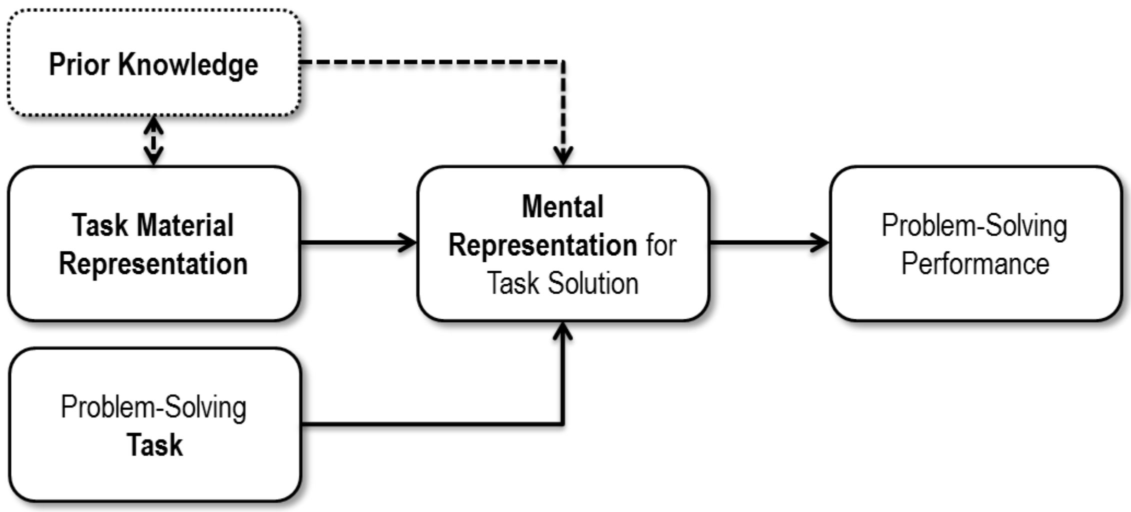

Figure 6.2: Cognitive Fit in Problem Solving (adapted from Shaft and Vessey (2006))

The cognitive fit theory assumes that a representation that fits an individual reduces the cognitive load. In other words, the best fit between the (problem-solving) task and the (problem) representation is requested to maximise the (problem-solving) performance (see Figure 6.2). "The cognitive fit theory states that when the task material representation fits with the task to be executed, people tend to be more effective and more efficient in executing the task […]" (Claes, Gailly and Poels, 2013, p.171).

Claes et al. (2015) combined the several cognitive theories (cognitive load theory, cognitive fit theory and human memory theory) and integrated them into a conceptual framework.

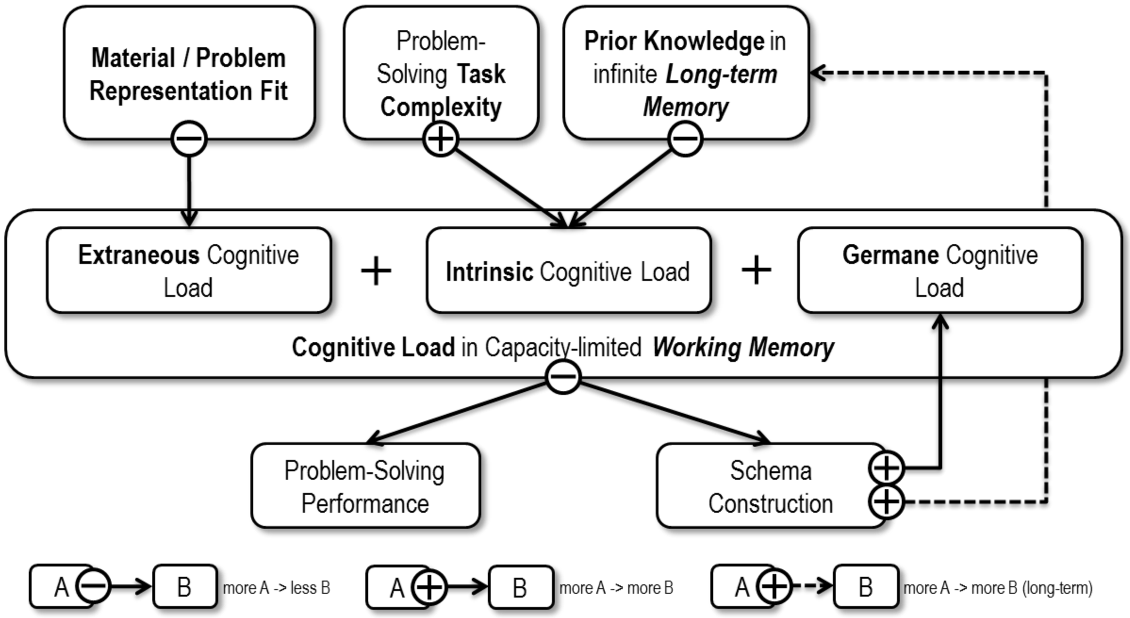

Figure 6.3: Causal Model of Cognitive Load in Working Memory (adapted from Claes et al. (2015))

Figure 6.3 shows an adapted version of the causal model of Claes et al. (2015) explaining the cognitive load in human working memory.

- Extraneous cognitive load: "Extraneous cognitive load mainly depends on the input material representation fit […]. A higher fit requires a lower cognitive load" (Claes et al., 2015, p.13). A higher fit occurs when the material, including models and documents, represents the current problem.
- Intrinsic cognitive load: "The intrinsic cognitive load increases for more complex tasks and decreases in case the [individual] possesses more relevant prior knowledge […]" (Claes et al., 2015, p.13). Prior knowledge

can be domain knowledge, modelling language expertise or methodical knowledge.

- Germane cognitive load: “Germane cognitive load is caused by loading information in working memory for the construction of cognitive schemas, which is not a prerequisite for the task, but rather the result of learning” (Claes et al., 2015, p.13).

6.2.1 What is Cognitive Adequacy?

If the working memory does not overload, as a result of the cumulative loads, then the problem-solving performance is adequate and new schema can be constructed (learning). Therefore, the following definition can be stipulated concerning cognitive adequacy of the knowledge worker.

Definition 6.5: Cognitive Adequacy

Cognitive adequacy can be reached by reducing and optimising the cognitive load (the amount of mental effort in the working memory) of the knowledge worker.

From the perspective of this study, only the extraneous cognitive load can be influenced in a short time. This causal model implies that the material/problem representational fit needs to be maximised. For doing so, the complexity of the representation needs to be optimised. Therefore, in the following section, complexity is defined and finally the similarity and complexity fit is introduced as an evolution of the material/problem representational fit.

6.2.2 What is Complexity?

A widely used definition of complexity is defined by the IEEE (1991, p.46) as follows: "The degree to which a system or component has a design or implementation that is difficult to understand and verify". This degree is difficult to evaluate. Nevertheless, Saltzer and Kaashoek (2009, pp.10–11) identified five signs of complexity:

1. *A large number of components*: The **number of elements** can affect whether a system is regarded as complex or not.
2. *A large number of interconnections*: Similar to the number of elements, a **large number** of existing or possible **interconnections** can be regarded as a "chaotic" system.
3. *Many irregularities*: Irregularities can be considered to be a huge amount of **exceptions** in arranging components or **non-repetitive** interconnections.
4. *A long description*: A description of a system can be regarded as adequate if it explains every aspect and detail of a system. In contrast, a reduction of a description to the shortest specification focusing on the core elements can be more efficient, even if this would be a loss of information. Whether an adequate description should be long, compressed without information loss (Kolmogorov complexity) or reduced is relative to understanding.
5. *A team of designers, implementers or maintainers*: A system not being understandable by just one person can be regarded as a fundamental issue, because maintenance and construction may require expertise, coordination and communication with different stakeholders.

6.2.3 Reducing Cognitive Load

Based on the previously described investigation concerning cognitive adequacy, cognitive load and complexity, Figure 6.4 shows two elements which ensure cognitive adequacy in this ICEBERG-PE approach.

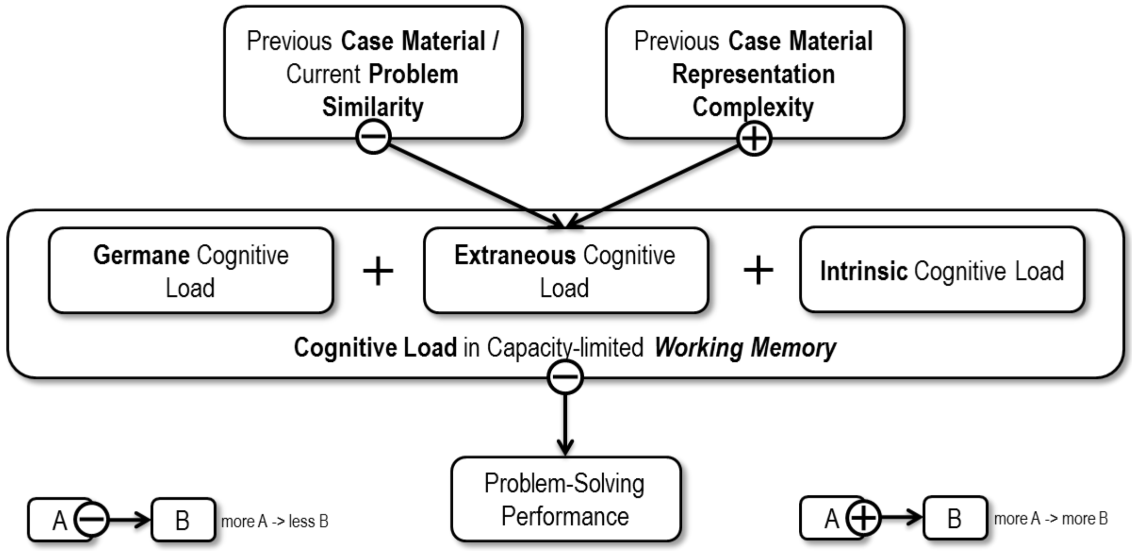

Figure 6.4: Similarity and Complexity Fit Reducing (Extraneous) Cognitive Load

The "previous case material / current problem similarity" can be ensured using a sufficient CBR similarity mechanism, which will be introduced in Chapter 7. The "previous case material representation complexity" needs to be reduced using a less complex way to describe previous cases. The case material refers to the case content and focuses on the process execution context and its graphical representation possibilities (modelling languages). The following chapter will continue with the complexity investigation concerning modelling languages as a potential case content element that is cognitively adequate.

6.3 Potential Case Content Modelling Languages

This section describes the investigation concerning potential modelling languages with regard to the cognitive adequacy and complexity as described in the previous chapter. Therefore, the following research question will be addressed here:

RQ 2.1: How can functional and process knowledge be included in a case description that is cognitively adequate to humans?

The chapter starts with a description of how the complexity of modelling languages can be assessed, which is followed by a delineation of related case content languages under the consideration of complexity and usage.

6.3.1 Complexity of Modelling Languages

Recker et al. (2009) underlined the importance of analysing the conceptual modelling languages from a complexity point of view. A reduction in complexity could affect the learnability and ease of use of a modelling approach (Recker et al., 2009). This result could increase the overall usage, longevity and success of implemented modelling approaches (Recker et al., 2009). "Even though the measure of complexity may not be perfectly accurate, a rough estimate of a method complexity is better than no information" (Recker et al., 2009, p.2).

Siau and Rossi (2011) analysed several evaluation techniques for modelling languages and came up with the following three main categories: feature comparison, theoretical and conceptual evaluation, and empirical evaluation (Recker et al., 2009):

- Feature comparison can involve a checklist-based evaluation comparing certain modelling features or requirements.
- Empirical evaluation involves surveys, laboratory experiments or case studies.

- Theoretical and conceptual evaluations can contain methods from the metamodelling domain and further techniques from ontology-based evaluation or metrics analysis. Theoretical and conceptual evaluations do not involve empirical data, are considered to be objective and are not biased by a concrete usage scenario.

Recker et al. (2009) presented a complexity evaluation (see Table 6.1) of UML and BPMN based on a metrics analysis (Rossi and Brinkkemper, 1996). They rely on the cognitive load theory of Chandler and Sweller (1991) by presenting the hypothesis that "[…] the presence of more modeling elements increase cognitive load for the modeler", building the overall assumption that "[…] a method [is] harder to learn and with more difficult rules to follow" (Recker et al., 2009, p.4).

Table 6.1: Complexity Metrics Evaluation Results (Recker et al., 2009)

Method	*Objects (Obj)*	*Relationships (Rel)*	*Properties (Prop)*	*Prop/Obj*	*Prop/Rel*	*Total Complexity*
BPMN FULL	90.00	6.00	143.00	1.52	1.33	169.07
BPMN CONCRETE	57.00	6.00	74.00	1.19	1.33	93.60
UML FULL	57.00	53.00	72.00	1.26	1.36	106.00
UML Activity diagrams	8.00	5.00	6.00	0.75	0.20	11.18
UML State diagrams	10.00	4.00	11.00	1.00	0.50	15.39

Table 6.1 shows the complexity analysis of Recker et al. (2009) based on the metrics of Rossi and Brinkkemper (1996). The results indicate that a BPMN subset, BPMN-CONCRETE, consisting of graphical instantiations only as defined by Recker et al. (2009), has a similar complexity as UML-FULL. Unsurprisingly, UML activity and state diagrams are less complex than both

BPMN sets. Recker et al. (2009, p.6) argue that the "[…] analysis indicates that the theoretical complexity of BPMN is higher than that of UML […]. [The] actual usage complexity of UML and BPMN may in fact be quite different […] in practice". Further Recker et al. (2009, p.6) argue that "[…] not all the constructs are used all the times and not all the constructs are equally important". This means that such a theoretical complexity analysis can only complement a practical analysis. To conduct a comprehensive and holistic complexity analysis, the practical complexity should be considered in addition to the theoretical complexity. "It will be key to arrive at an informed opinion not only about how a modeling method is theoretically possible to be used (the theoretical complexity), but also how it is actually being used (the practical complexity)" (Recker et al., 2009, p.6). The following section presents the BPMN modelling language as a potential language for the case content with consideration of the practical complexity.

6.3.2 Complexity of BPMN

This section describes the modelling language BPMN with the consideration of practical complexity. Based on BPMN version 1.0, zur Muehlen and Recker (2008) made an analysis of the distribution of modelling element usage. They analysed 126 BPMN diagrams from different practitioners (consultants, seminar participants and online sources) and presented the result as shown in Figure 6.6. After grouping the elements (see Figure 6.5), they came to the conclusion that the "[…] average subset of BPMN used in these models consisted of just nine different symbols" (zur Muehlen, 2008). This means that only approximately 20 percent of the whole BPMN 1.0 element set has been used by common users.

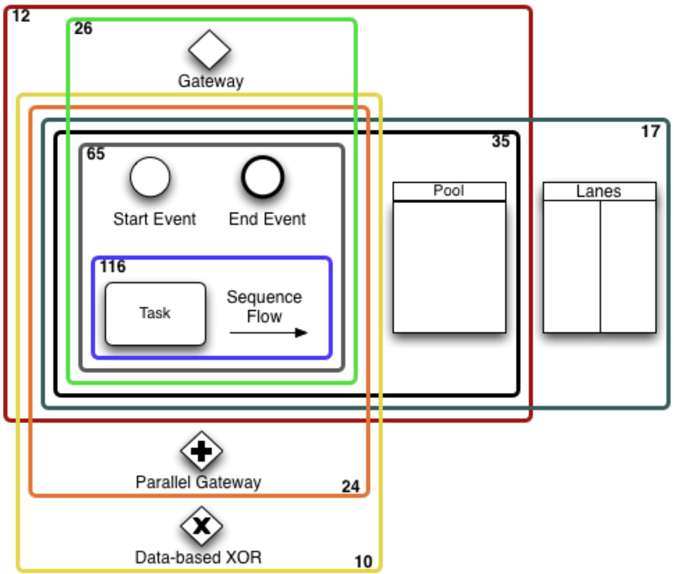

Figure 6.5: Grouping of BPMN 1.0 Elements (Muehlen and Recker, 2008)

The grouping has been elaborated by looking to the co-occurrence of certain elements and calculating the frequencies (as depicted in Figure 6.5 by the numbers in the edges). The detailed results are depicted in Figure 6.6.

Zur Muehlen (2008) came to the conclusion that:

1. **The common core set** of different BPMN elements is in average **small**.
2. **There are two types of modellers**. One group "represent[s] organizational responsibility for tasks" with pools and lanes, and another group "represent[s] the control flow rules of the process in detail" with gateways.

Overall Zur Muehlen (2008) suggested, that "standards-makers should review whether a more complete, but also more complex language is a desirable result of the standardization process". This suggestion was made before the BPMN specification 1.1 was released. The blog post from zur Muehlen (2008) and the underlying research paper from zur Muehlen and Recker (2008) initialized an intense debate about the expressiveness of the next BPMN specification (zur Muehlen and Recker, 2013).

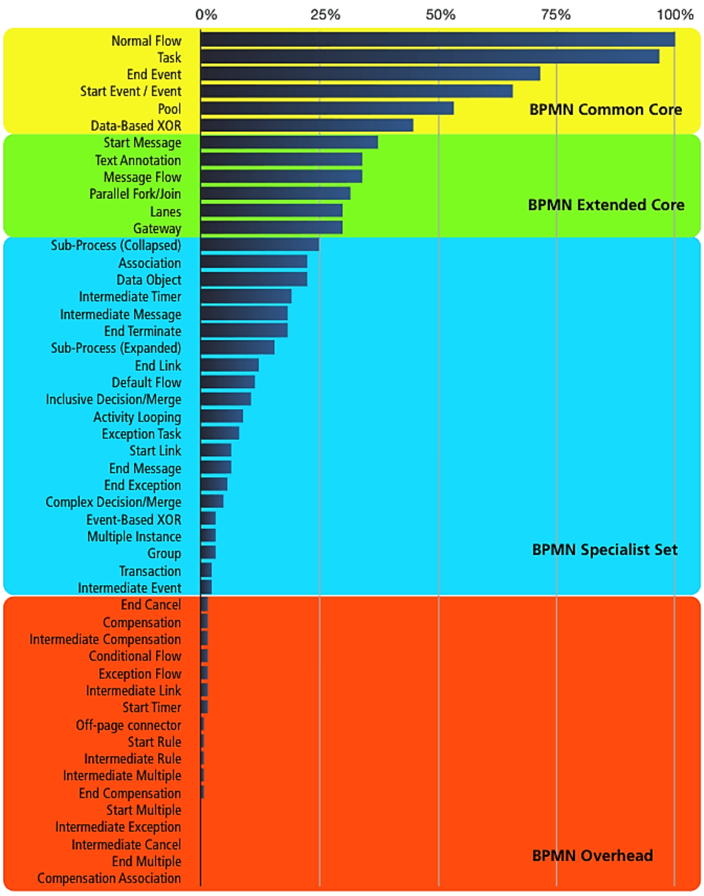

Figure 6.6: Frequency Distributions of BPMN Construct Usage (zur Muehlen, 2008)

Additionally, Robert Shapiro (2010) introduced in a webinar of the Workflow Management Coalition (WfMC) a concept of sub-classes (see Figure 6.7) just before the BPMN specification was released.

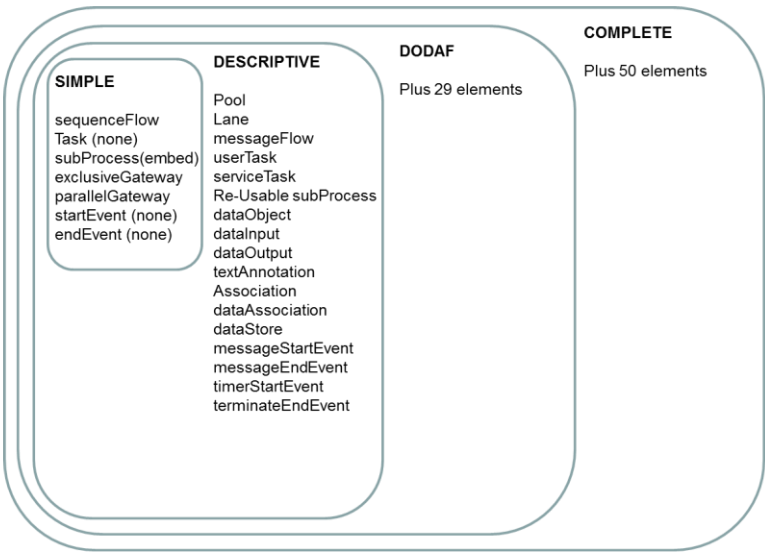

Figure 6.7: Sub-Classes within Process Modelling (Shapiro, 2010)

The OMG specified in the BPMN 2.0 standard (OMG, 2011) three conformance subclasses/levels based on the suggestions of Silver (2011) and the suggestions of Shapiro (2010). These conformance subclasses/levels represent subsets of the BPMN modelling elements for the following different use cases (see Figure 6.8): descriptive, analytical and executable.

- The descriptive level contains a limited set of modelling elements for different stakeholders in the business.
- The analytical level extends the descriptive subset with additional elements such as additional event types and exceptions.
- The executable level contains elements that are needed to support the execution of a BPMN process.

The mentioned proposals and analyses clearly set out that a specific subset is relevant in practice to avoid practical complexity. Unfortunately, based on the proposals, it is not possible to ultimately define an adequate subset. Therefore, the case model of ICEBERG-PE should provide the possibility of selecting any subset of a modelling language according to the user's needs.

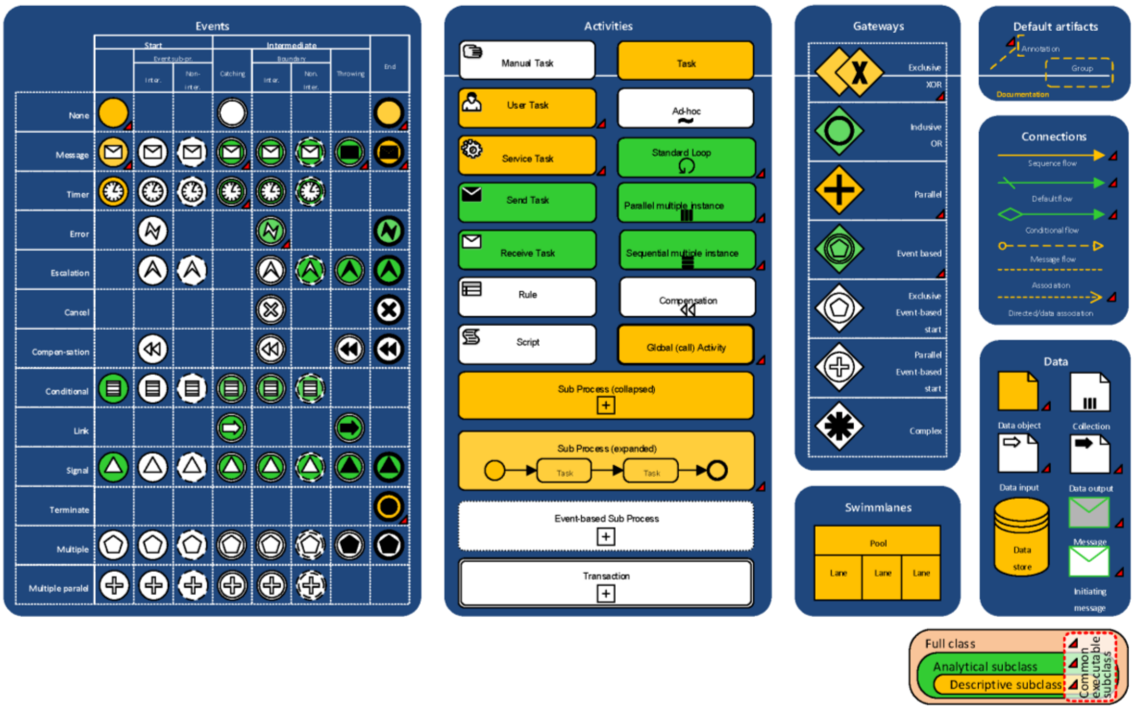

Figure 6.8: BPMN 2.0 Subclasses (adapted from Polančič (2014))

6.3.3 Flexibility of CMMN

As mentioned in Section 2.1.2, case management provides the ability to manage cases and gives the knowledge workers more freedom and flexibility. The workers usually drive case management, which is characterised by a "minimal predefined encoding of the work to be performed" (OMG, 2014, p.5). This means that only limited parts of the work to be performed is modelled during design time. Nevertheless, planning during runtime is an important aspect of case management.

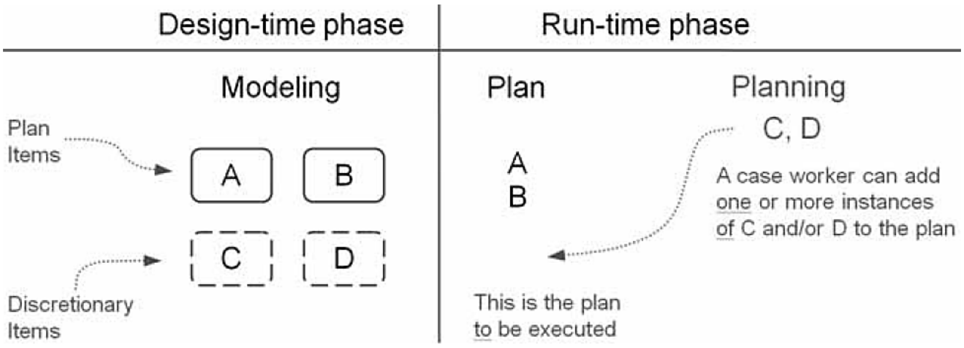

Figure 6.9: CMMN 1.0 - Design Time Phase Modelling and Runtime Phase Planning (OMG, 2014)

CMMN, a model and notation for case management, is currently released as version 1.0 (OMG, 2014) and as beta 1.1 (OMG, 2016) from the Object Management Group (OMG). As shown in Figure 6.9, case and knowledge workers can select existing tasks, reorder the sequence and collaborate with other case workers.

Nevertheless, the flexibility is limited in CMMN to only so-called discretionary items, which are predefined during design time and can be added to the case plan. In Section 6.4 a comparison between BPMN and CMMN in presented. Based on this, a new case model is presented in Section 6.4 using a suggested modelling language called BPFM (Cognini, Corradini, Polini and Re, 2015b; a; Cognini, Hinkelmann and Martin, 2016).

6.4 Case Model of Ontology-based CBR and Process Execution

This section introduces the case model of the ICEBERG-PE approach in consideration of the previous research concerning complexity, potential case content modelling languages and case description. Therefore, this section addresses research question 2 as a whole, including its sub-questions:

RQ 2: What should the case description for knowledge-intensive work consist of?

> *RQ 2.1: How can functional and process knowledge be included in a case description that is cognitively adequate to humans?*
>
> *RQ 2.2: How can the process execution context be integrated into the case description?*

In the following section, the objectives for the case model are introduced, followed by a conceptual description of the case content and characterisation. The implementation of the ICEBERG-PE approach, including the implementation of the case model, is provided in Chapter 8 and showcased using the admission process application scenario in the evaluation Chapter 9.

6.4.1 Objectives for Case Model

In this section, the objectives for the case model are introduced to answer the following research question:

RQ 2: What should the case description for knowledge-intensive work consist of?

The research objectives (RO) are derived from the admission process application scenario and its case data with the help of the stakeholders during two interviews, one on 31st May 2015 and a second one on 4th April 2015.

Objectives for case content: The case content should contain a representation of certain process fragments, which can be manually executed and adapted. The objectives for the case content are:

- Process fragment[16] modelling: The cases should consist of *process fragments (RO-A)*, which can be modelled by the knowledge workers.
- Representation: The case content should be presented using a *graphical representation (RO-B)*.
- Maintaining: The case content should have certain *update functionality (RO-C)* to evolve the process fragments.
- Modelling of information resources: The case content should contain *information resources (RO-D)* such as documents; data objects are system resources.
- Modelling of variants: Despite the objective of the fragment modelling, the application scenario stakeholders requested to model certain *variants (RO-E)* of potential activities and flows within one case, which can be concrete or generalised.
    - Concrete process fragment case: It should be possible to model process fragments as new *concrete cases (RO-F)*.
    - Generalizable process fragment case: Additionally, the stakeholders are requested to provide a possibility to manually model certain process fragments as *generalised cases (RO-G)*, which can be updated.

Objectives for case characterisation: The case characterisation enriches the case description with additional information which serves as a processable basis for a similarity measure. The following objectives for the case characterisation have been derived:

[16] Process fragment: The term process fragment has been introduced by the author of this study during the requirement analysis. The stakeholders referred this concept as loose and unspecific fragmental element containing specifically performed activities, sub-process or case data – the term is introduced in Section 6.4.2.1.

- *Structural*: The cases need to be described in a *structured (RO-H)* way.
- *Ontology-based*: The case characterisation should be described with a reusable vocabulary which can be provided in an enterprise *ontology (RO-I)*.
- *The inclusion of process execution information*: The case characterisation should include *process information (RO-J)* such as variables or roles to assign tasks to appropriate performers.

6.4.2 Case Content containing Process Knowledge[17]

Based on the derived requirements, the following case content case been elaborated. This section starts with a short investigation concerning process fragments followed by an introduction to a modelling language for the process fragments. Finally, the case content model will be introduced.

6.4.2.1 Process Fragments

As mentioned in the objectives section above, knowledge workers requested a certain process fragment possibility. They referred to this concept (see the conceptual sketch in Figure 6.10) as a loose and unspecific fragmental element containing specifically performed activities, sub-processes or case data. Knowledge workers should be able to make decisions based on process fragments, which can only be made by the knowledge workers themselves and which can only be executed by humans (see human tasks of case management model and notation). "Process fragments are reflecting the partial and intermittent knowledge one modeller [or a knowledge worker] has at a certain time about a specific situation" (Eberle, Unger and Leymann, 2009, p.399). For knowledge workers, it seems useful to take approaches that allow them to

[17] Some verbatim passages presented in this following section, which have been achieved related to this thesis project, are have been published in the following co-authored publications (Cognini, Hinkelmann and Martin, 2016; Martin et al., 2016)

structure the business process in part as process fragments since there are no fully defined models

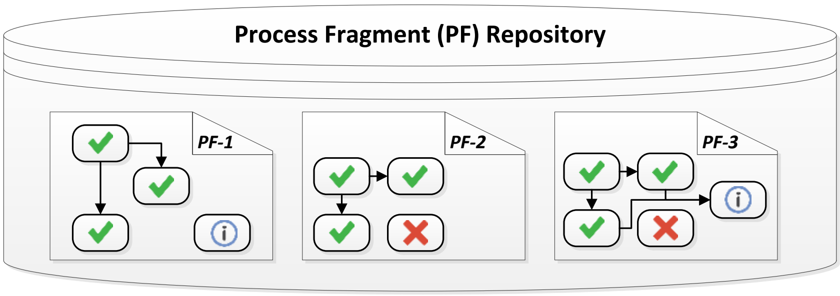

Figure 6.10: Process Fragment Repository

they can easily adapt or modify at runtime (Reichert and Weber, 2012). Therefore, the followed case model focuses on the man-made modelling of process fragments without a reduced granularity depending on the variation of the knowledge worker's situation.

6.4.2.2 Modelling Languages for Process Fragments

According to Swenson (2013), BPMN is suitable for representing cases in a case management system as case-based reasoning. Further, Swenson (2013) explains why BPMN should not be used for modelling in the adaptive case management (ACM) context, which is also valid for case-based reasoning approaches. End users of a case management system, including CBR systems, do not have enough knowledge and skills to model or update a BPMN model. Knowledge workers can describe the performed activities. Eventually they can describe alternative activities as well, but they may struggle with describing the temporal order of the alternative activities since the focus is on the current situation. Additionally, modifying an existing BPMN diagram, which is modelled by someone else, can produce similar difficulties as modifying software source code from someone else. Finally, imperative languages have been designed to be complete, ensuring the whole business process is modelled and not just parts.

To deal with the mentioned difficulties of BPMN, the OMG has introduced CMMN, which is a declarative language. CMMN can be used to model partially structured and not necessarily repeatable business processes, which are not necessarily predefined in advance. As shown in Table 6.2, with CMMN, it is not possible to model complex constraints. For instance, it is not possible to specify that at least one activity in a model must be executed. Such constraints are needed to create cases in the CBR system, which are evolving over time and gaining a higher quality and completeness. Additionally, the data representation is only limited in CMMN. This limitation of data representation leads to limited information about the type of data or document being available to the knowledge workers as part of the modelled case content.

Table 6.2: Comparison of Modelling Languages (adapted from Cognini, Hinkelmann and Martin (2016))

	BPMN	*CMMN*	*BPFM*
For BP Modelling	Yes	Yes	Yes
Language Type	Imperative	Declarative	Declarative
Defined Activities Flow	Full	In Part	In Part
Complex Constraints	Yes	No	Yes
Data Representation	Yes	In Part	Yes
Variants Representation	No	No	Yes

To deal with these issues, Cognini, Hinkelmann and Martin (2016) propose the use of the Business Process Feature Model notation (BPFM) (Cognini et al., 2015b; a) as a modelling language for the case content. BPFM notation permits defining business activities without specifying an execution order, and with considering complex constraints and different types of data objects. Moreover, BPFM can be regarded as a configurable process model containing more than one variant (variant representation) of process fragments. BPFM notation is explained in the next subsection.

6.4.2.3 Business Process Feature Model (BPFM) Notation

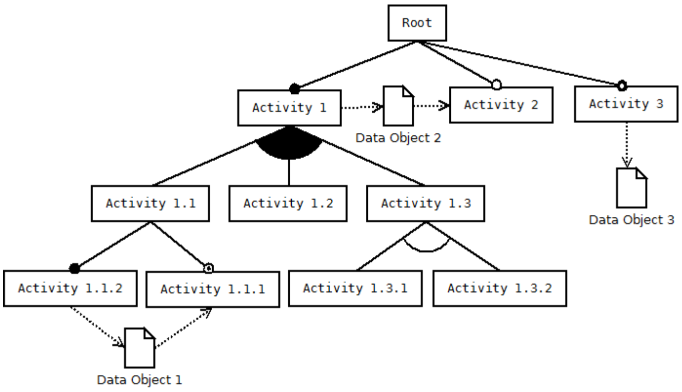

Figure 6.11: Exemplary Business Process Feature Model (adapted by Cognini from Cognini, Hinkelmann and Martin (2016))

The business process feature model (BPFM) consists of a tree of related activities (Cognini et al., 2015a). As depicted in Figure 6.11, the root identifies the main services, and each internal (non-leaf) activity denotes a sub-process that can be further refined. The external (leaf) activity represents an atomic task. BPFM allows for using the same meaning and graphical representation as BPMN 2.0, and for defining constraints between activities.

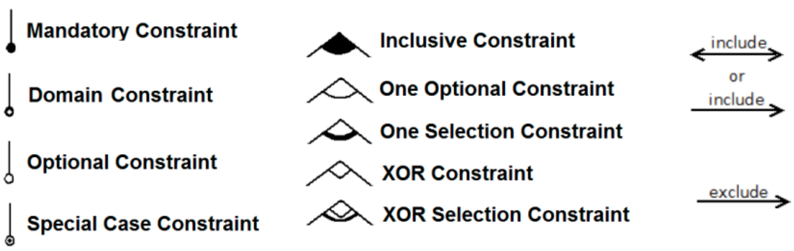

Figure 6.12: Business Process Feature Model Constraints (adapted by Cognini from Cognini, Hinkelmann and Martin (2016))

Constraints (see Figure 6.12) are used to express whether child activities can or have to be selected in the configuration to be included in the BP variant, and whether they can or have to be included in each execution path of the BP variant. BPFM, as presented in Cognini, Hinkelmann and Martin (2016), consists of nine different constraints. BPFM manages all types of BPMN 2.0 data objects, including data object states, with the same modelling notation. As shown in Figure 6.13, BPFM uses the data objects to represent the "flow" of a business process or process fragment.

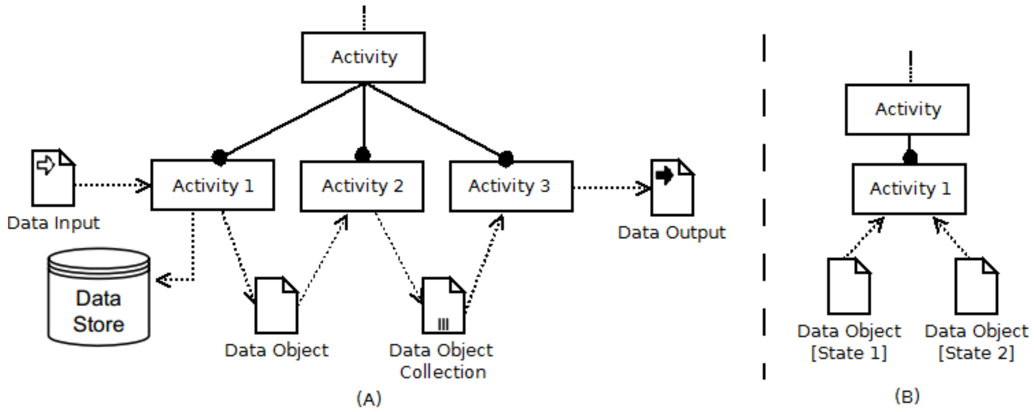

Figure 6.13: Data Object in BPFM (Cognini, Hinkelmann and Martin, 2016)

The final question of whether BPFM qualifies as a cognitively adequate case content language can partially be answered. Based on an evaluation of Cognini (2015), it is possible to conclude that BPFM is regarded as equal to BPMN regarding the required abilities of the end users. With the usage of BPFM, however, it is possible to model partial process fragments. BPFM-based process fragments can be modelled with a subset of modelling elements compared to BPMN. Therefore, it is possible to conclude that BPFM-based process fragments lead to a lower complexity.

6.4.2.4 Case Content Model

The case content model is formalised in the ICEBERG-PE ontology, which contains elements for describing the case, the case content description and further case items.

The case content description of the ICEBERG-PE approach consists of at least one BPFM-based process fragment that contains experienced knowledge about the work that has been done in a previous or current case. The knowledge worker can model the current case freely without having strong restrictions or limitations. In addition to the present case, the knowledge worker can include certain options describing alternative activities or flows. Figure 6.14 shows on the left side of the ICEBERG-PE system dialogue an example of a BPFM-modelled case content description.

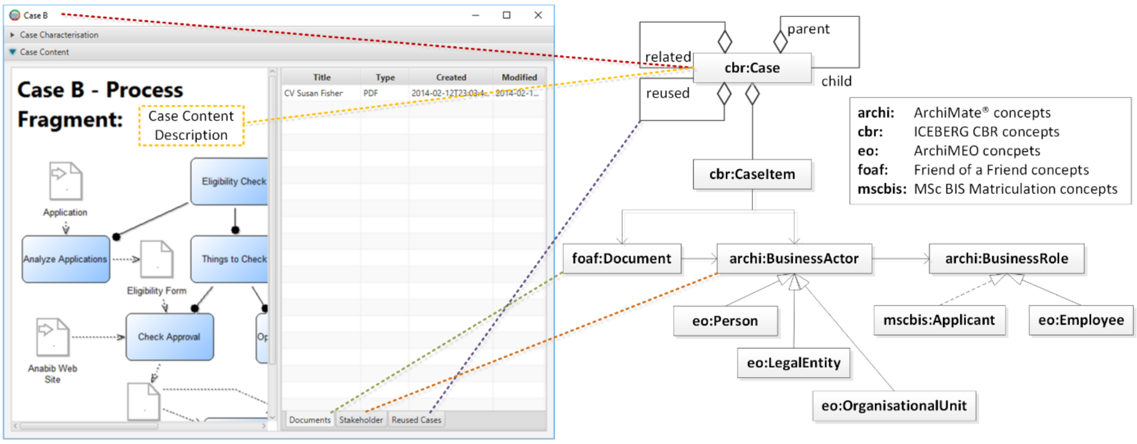

Figure 6.14: Case Content Model Elements

The BPFM modelling language provides the expressiveness to tackle the objectives for case content. In addition to the BPFM elements, the following case content elements are used to describe case interrelationships:

- Parent: The parent task/case element is used to express a possible sub-task/case relationship.
- Child: The child task/case element is the inverse of the parent element.
- Related: This element is used to express that there are related tasks/cases.
- Reused: The reused task/case element is used to list tasks, which have been reused in the adaptation phase of the CBR cycle.

Based on the original notion of CBR, some researchers investigated the generalisation and abstraction of cases (Maximini, Maximini and Bergmann, 2003). The generalisation can reduce the complexity of the cases, increase the flexibility and minimise the size of the case base to enhance the retrieval efficiency (Bergmann and Wilke, 1996). Abstraction differs from generalisation. According to Müller and Bergmann (2015, p.396), "[…] abstraction […] would require reducing the overall granularity of workflows (e.g. less tasks and data items) […]". This differentiation is particularly important when implementing an automatic algorithm. In contrast, the manual generation and refinement of generalised cases is aimed at this approach.

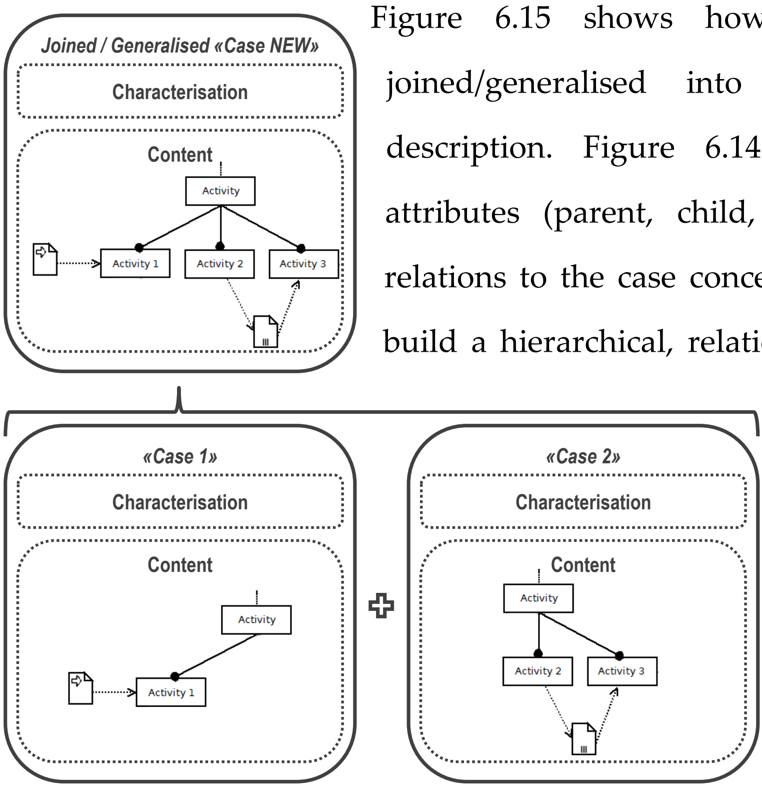

Figure 6.15: Cases Joined/Generalised in a Single BPFM Model

Figure 6.15 shows how two cases could be joined/generalised into a new case content description. Figure 6.14 shows the mentioned attributes (parent, child, related and reused) as relations to the case concept, which can be used to build a hierarchical, relational and generalised case structure. Additionally, the reused cases (see example in Listing 6.1) can be kept as a reference to the current case `(reusedCases)`. If during the execution of the case content model the model needs to be adapted, the dependency can be made explicit. In contrast, standard CBR revises the case model and retains it as a new case. This new case is independent of all the other cases and the information that it is a variant of an already existing case is lost. Instead of storing a new case, the user can adapt the current case content model by adding the case property: parent, child, related or reused. The mentioned elements can also be used knowledge to realise the generalisation and abstraction of cases.

Listing 6.1 shows the case content description `(cbrCaseHasSolutionDescription)` property, which is implemented as datatype property containing a markup language. This markup-based description can be edited by the knowledge worker using an editor and also links the modelled BPFM fragment.

Listing 6.1: Exemplary Case Content – Reused Case and Solution Description

```
mscbis:MScBISAdmissionCase_B
  rdf:type mscbis:MScBISAdmissionCase ;
  cbr:caseItem mscbis:case_item_MScBISAdmissionCase_B_Applicant ;
  cbr:caseItem mscbis:case_item__CV_Susan_Fisher ;
  cbr:cbrCaseHasSolutionDescription "<html><body>...</body></html>" ;
  cbr:reusedCases mscbis:MScBISAdmissionCase_A ;
.
```

Figure 6.14 and Listing 6.2 shows the case's stakeholders or collaborators. The stakeholders are linked to the ArchiMEO concept `Person`, which is inherited from `BusinessActor`, with `Applicant`, which is inherited from `BusinessRole`.

Listing 6.2: Exemplary Case Content – Stakeholder

```
mscbis:case_item_MScBISAdmissionCase_B_Applicant
  rdf:type cbr:CaseItem ;
  cbr:caseItemRepresentedBy mscbis:Susan_Fisher ;
.
mscbis:Susan_Fisher
  rdf:type eo:Person ;
  eo:personPerformsBusinessRole mscbis:Applicant ;
  foaf:personHasFamilyName "Fisher" ;
  foaf:personHasFirstName "Susan" ;
.
eo:personPerformsBusinessRole
  rdf:type owl:ObjectProperty ;
  rdfs:domain eo:Person ;
  rdfs:range archi:BusinessRole ;
.
```

As shown in Figure 6.14, each case can be described by case items. A case item can be any element that has been attached to the case to describe the case content. Based on the admission process application scenario, documents and case stakeholders are proposed here as case items. The case item document is a concept from the FOAF vocabulary, as shown in Listing 6.3.

Listing 6.3: Exemplary Case Content – Document

```
mscbis:case_item__CV_Susan_Fisher
  rdf:type cbr:CaseItem ;
  cbr:caseItemRepresentedBy mscbis:case_document__CV_Susan_Fisher ;
.
mscbis:case_document__CV_Susan_Fisher
  rdf:type foaf:Document ;
  eo:documentHasSubjectBusinessRole mscbis:Applicant ;
  cbr:documentHasFilePath "file_repository/" ;
  elements:documentHasFormat eo:pdf ;
  elements:documentHasTitle "CV Susan Fisher" ;
  elements:documentHasType eo:PDF ;
.
mscbis:Applicant
  rdf:type cbr:Role ;
  rdfs:comment "A new MSc BIS applicant" ;
  rdfs:label "Applicant" ;
.
```

This case content modelling approach allows a flexible adaptation of cases. The content model is linked to the ArchiMEO ontology and ensures that the enterprise-specific concepts are included. The implementation of the case content model is described in Chapter 8 and evaluated using the application scenario in Chapter 9. Table 6.3 summarises the realisation of the stated objectives (see Section 6.4.1) for the case content model.

Table 6.3: Case Content Model Objectives and Realisation

Objectives	***Realisation***
Process fragments (RO-A)	The process fragments are included using the BPFM modelling language as part of the case content description.
Graphical representation (RO-B)	A graphical representation is provided by the inclusion of the BPFM modelling notation. It is not restricted to BPFM; any modelling language can be used.
Update functionality (RO-C)	The update functionality is given by the possibility of creating and updating the case content during the CBR cycle. It is even possible to update content from already-learned cases through a special feature of the prototype, although this is highly controversial.
Information resources (RO-D)	The information resources are part of the case content and represented as document resources. Further information elements can be added using the case content description or in the context of the BPFM model.

Variants (RO-E)	The variants can be modelled in a BPFM-based content description using the BPFM constraints.
Concrete cases (RO-F)	The concrete cases are the usual way to describe the current work or the work that has been done in the case content as BPFM-modelled process fragments.
Generalised cases (RO-G)	If required, the knowledge worker can create generalised cases based on previous cases using the ontology properties and the BPFM-content modelling language as presented.

6.4.3 Case Characterisation describing Process Knowledge[18]

This section introduces the case characterisation that describes the process knowledge. As mentioned in Chapter 5, this ICEBERG-PE approach relies on ontology-based CBR and uses an underlying ontology (see the ontology structure in Chapter 5). The ICEBERG-PE case characterisation is used to describe (characterise) the case itself and to assess the similarity between cases using a retrieval mechanism and method. The configuration of the retrieval mechanism, the similarity model (see Section 7.2), using the case viewpoint model (see Section 5.4), defines the case characterisation from an implementation/instantiation perspective and is described in Chapter 7.

Section 5.5.4 describes how a CBR approach can be combined with process execution. If the CBR system is used as a stand-alone system (see option 1 in Section 5.5.4) certain workflow relevant data is needed to replicate a task management system. Therefore, the web service human task specification has been analysed, and a human task ontology has been extracted. Web services human task (WS-HumanTask) is a web service specification (OASIS, 2012) for delegating and assigning tasks to human actors. The specification has been

[18] Some verbatim passages presented in this following section, which have been achieved related to this thesis project, are have been published in the following co-authored publications (Cognini, Hinkelmann and Martin, 2016; Martin et al., 2016)

published as a draft by the OASIS BPEL4People technical committee as an extension to the web service business process execution language (WS-BPEL) (OASIS, 2007). Figure 6.16 shows the extracted and re-modelled human task ontology, which is included in the process execution ontology (see Section 5.4). The presented elements are usually used by a workflow engine and the corresponding task/work list. This is particularly the case if the CBR system is used as an invocable system. As mentioned in Section 5.5.4, this approach is demonstrated using a workflow management system. Therefore, only a limited set of the human task ontology elements will be used in this instantiation, as presented in Chapter 8.

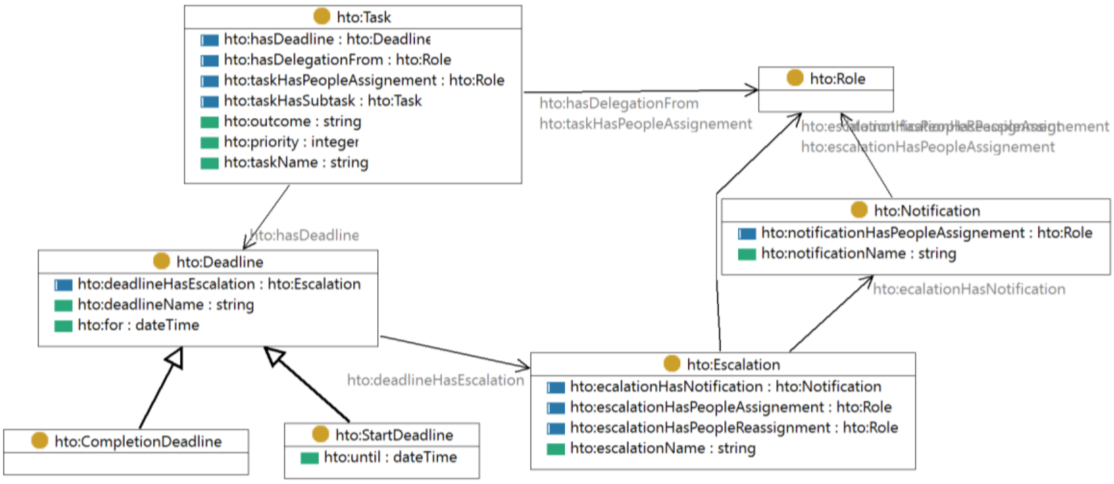

Figure 6.16: Human Task Ontology

Figure 6.17 shows a partition of an exemplary case characterisation and configuration including elements of the process and domain knowledge using the ICEBERG-PE ontology structure. Thus, it is possible to enrich the case-based reasoning system with domain knowledge, which improves the reflectivity of the system and reduces the effort to acquire the vocabulary (Recio-Garía and Díaz-Agudo, 2007; Gao and Deng, 2010). With the inclusion of an ontology structure in a CBR system, it is possible to "[…] take advantage of this domain knowledge and obtain more accurate results" (Recio-García, Díaz-Agudo and González-Calero, 2008, p.54).

The ICEBERG-PE approach (as an extension of the ICEBERG approach; see Chapter 7 and Martin et al. (2016) for details on implementation) provides a wide range of similarity functions for retrieval and adaptation. As shown in Figure 6.17, the case characterisation will be defined at the time the similarity configuration of the retrieval mechanism – the similarity model (see Section 7.2), using the case viewpoint model (see Section 5.4) – is created. The vocabulary for describing the cases is domain specific and therefore different from one application scenario to another.

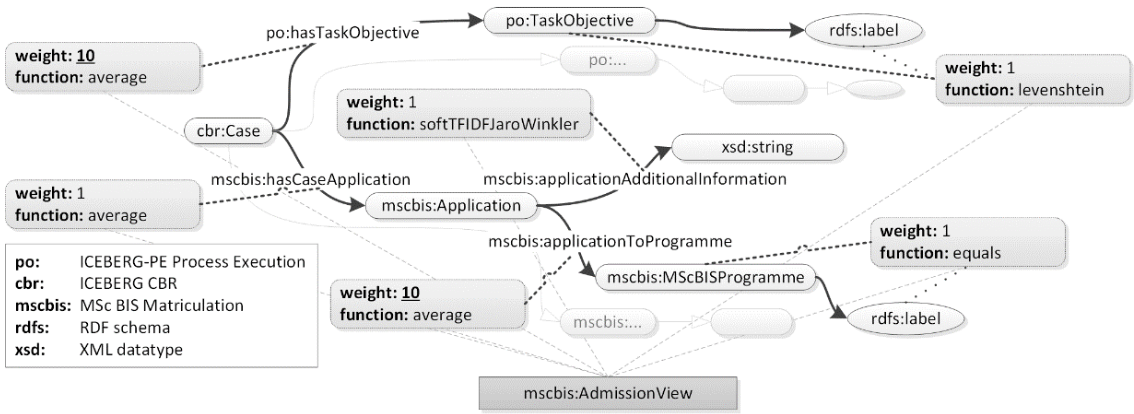

Figure 6.17: Exemplary Configuration of a Case Characterisation including View[19]

From the application scenario, the case characterisation objectives and the results of the task management system KISSmir introduced by Martin et. al. (2010) and Brander et. al. (2011a), the following process execution related elements have been derived and are required:

- Task objective: The task objective element describes the goal of the task itself. This element has some similarities to the name and/or description of a BPMN activity.
- Task role: The task role element is used to describe the role of the person involved in the task. Through the inclusion of an enterprise or domain

[19] The ontology structure has been simplified, and the corresponding element denotation has been abbreviated to fit the figure.

ontology, it is possible to reuse an existing enterprise-specific role/organisational model.

- Task user: The task user elements are used to indicate the person who described the case.

Listing 6.4: Exemplary Case Characterisation – Task Objective

```
mscbis:MScBISAdmissionCase_B
  rdf:type mscbis:MScBISAdmissionCase ;
  mscbis:mscBISAdmissionCaseIsCharacterizedByTaskObjective
     mscbis:TaskObjective_Eligibility_Check ;
.
mscbis:TaskObjective_Eligibility_Check
  rdf:type po:TaskObjective ;
  rdfs:label "Eligibility Check"@en ;
.
po:TaskObjective
  rdf:type owl:Class ;
  rdfs:subClassOf po:ProcessExecutionElements ;
.
```

Listing 6.4 shows an excerpt of an exemplary case characterisation including elements from the ICEBERG-PE ontology structure. The listing shows how a case (`MScBIS-AdmissionCase`) gets characterised by a task objective (`TaskObjective`).

The following Table 6.4 summarises the realisation of the stated objectives (see Section 6.4.1) for the case content model.

Table 6.4: Case Characterisation Model Objectives and Realisation

Objectives	***Realisation***
Structural (RO-H)	The structural case characterisation objective has been covered by the inclusion of structural CBR.
Ontology-based (RO-I)	The case characterisation can be described using elements, concepts and relationships from the enterprise ontology ArchiMEO and other ontologies.
Process execution information (RO-J).	Process execution information is reflected by concepts of the process execution and human task ontologies.

6.5 Conclusion

This chapter introduces the case model, which has been derived based on related work in conjunction with the admission process application scenario. Additionally, this chapter provides a conceptual answer to the following research questions, which are showcased by the implementation of the CBR services in the next chapter, implementation of the ICEBERG prototype and evaluation of the triangulated data sources:

- The research of the process execution context (RQ 2.2) laid the basis for the case characterisation. The processes execution context, which is enterprise context, can be integrated using information derived from an enterprise ontology and with the usage of a vocabulary that is linked to the same enterprise ontology.
- The investigation of complexity and cognitive adequacy builds the foundation for answering research question RQ 2.1. Based on the complexity investigation, potential case content modelling languages are presented. As a result of this investigation, the general procedure model (see Section 5.6.1) has been extended as a procedure model for process execution (see Section 5.6.2) in a next incremental design science research cycle.
- Finally, based on the objectives (RQ 2) for a case model, a case content model containing process knowledge is introduced (RQ 2.2). The case content model consists of elements from the ICEBERG-PE ontology structure and a potential case content modelling language BPFM with an acceptable level of complexity (RQ 2.1).

The next chapter describes the investigation of the CBR services and conceptually answers research questions 3, 3.1 and 3.2.

7 Case-based Reasoning Services

The case-based reasoning (CBR) services execute in the figurative sense the CBR cycle as introduced in Section 2.2. They are the interface between the conceptual and model-based investigation of the previous chapters, the ICEBERG-PE approach and the case model. The CBR services are presented in this section as a conceptual description, the similarity mechanism, and as a technical description.

This chapter gives an answer to the following research questions stated at the beginning of the thesis:

RQ 3: How can case-based reasoning services support process execution?

- *RQ 3.1: How can the similarity between cases for knowledge work be calculated?*
- *RQ 3.2: How can domain knowledge and contextual information be used for retrieval of cases and suggestion or adaptation of case items?*

The chapter starts with an introduction that gives an overview of the services and how they are used. Section 7.2 gives answers to research questions 3.1 and 3.2. Sections 7.3 and 7.4 close the CBR cycle and give a further answer to research question 3.2.

7.1 Introduction

Figure 7.1 shows the ICEBERG-PE CBR cycle including the functions and abilities of the CBR services. The ICEBERG-PE CBR approach, as presented here in this thesis, provides the following main CBR cycle functions:

1. The similarity mechanism uses a new query case to compare the characterisations of the cases in the case repository. This comparison or similarity evaluation is done automatically. As a result, a similarity value is assigned to each potential previous case.

2. The knowledge worker can adapt the knowledge from the retrieved cases to the current case by either adding the information manually or running an adaptation mechanism. This adaptation mechanism executes the default mechanism of transferring the case content of the retrieved case to the current one. This default mechanism can be controlled and modified by predefined semantic rules. This combination of manual and rule-based adaptation can be regarded as a semi-automatic adaptation. As a result, the knowledge worker receives a potential solution, which later can be determined to be a solved case.

3. The knowledge worker can revise the solved case based on the work that has been done or the gained insights and finally create a revised case.

4. The revised case is then ready for retention. The knowledge worker can trigger the retention mechanism, and the case will then be added to the case repository as a learned case which is available for the next cycle.

In the following section, the retrieval service is amalgamated with the CBR similarity, which is the main contribution of the ICEBERG approach (Martin et al., 2016) to the ontology-based CBR, the CBR adaptation and finally CBR learning.

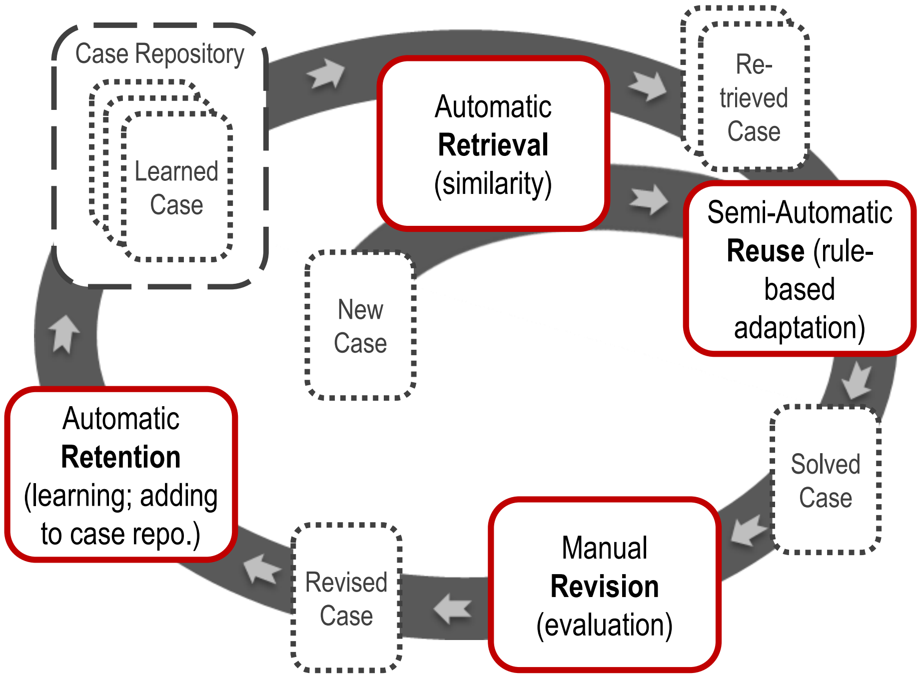

Figure 7.1: ICEBERG Cycle including Service Functions

7.2 Case Similarity[20]

As mentioned previously, retrieval is usually the starting point of the CBR cycle. Figure 7.2 shows the cycle of the ICEBERG-PE approach where a new case will be stipulated and transferred to a query case. This query case will then be used for retrieval (similarity computation) and later in the reuse phase (adaptation). The retrieval is based on the case characterisation exclusively and not on the case content. Therefore, no process model graph similarity analysis is applied.

As mentioned in the previous chapter, the characterisations of cases are expressed by a defined part of the enterprise and domain ontology. This explicit knowledge is used for the comparison of a query case with learned cases in the CBR retrieval phase, as shown in Figure 7.3. In this retrieval phase, the applied similarity measures use this explicit knowledge structure to compute the similarity among learned cases and query case.

[20] Some verbatim passages presented in this following section, which have been achieved in this thesis project, have been published in the following own authored publication (Martin et al., 2016; Martin, Emmenegger and Wilke, 2013).

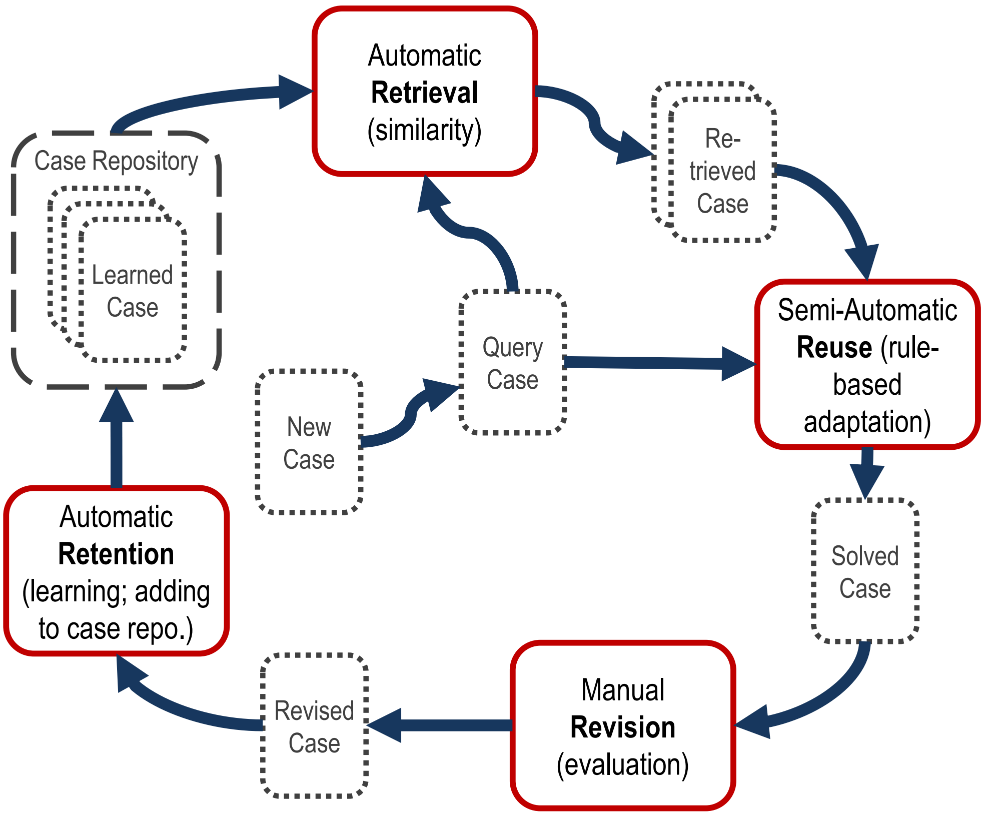

Figure 7.2: ICEBERG Cycle including Query Case

To define the concepts and properties that belong to the case characterisation, including the corresponding similarity functions and weights of the concepts and properties, the ICEBERG similarity ontology is used, which is based on OWL annotations. The ICEBERG similarity ontology can be referred as the similarity vocabulary for configuration as known from structural CBR systems. In this way, it is possible to configure the similarity model and ultimately the CBR system entirely within an ontology and with concepts and relations from an existing ontology. Such an ontology-based similarity approach has the advantage that explicit enterprise and domain knowledge can be modelled together with similarity measures and that a standard ontology development environment can be used to model the case characterisations.

The ontology language OWL provides a scheme containing different properties, which are used to determine the similarity model configuration. With the ICEBERG similarity ontology, it is possible to specify and select similarity measures, functions and weights for annotation, datatype and object properties, as well as for classes of the case characterisation.

As shown in Figure 7.3, either the knowledge worker or the invoking process execution system creates a case query based on the configured case characterisation structure and elements for the retrieval and ranking of learned cases. The query case is compared with all characterisations of the learned cases in the case repository. As a result, the similarity service returns a ranked list of learned cases assigned with a similarity value between 0 and 1.

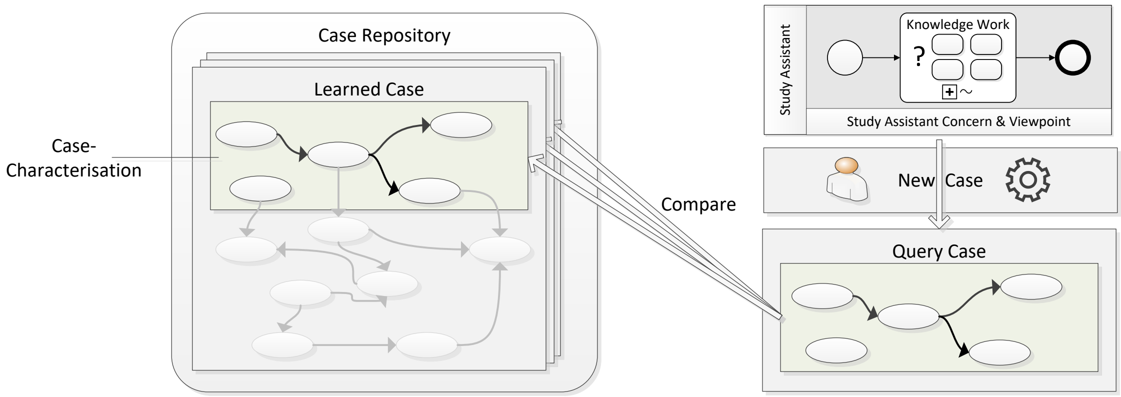

Figure 7.3: Query Case Characterisation used for Comparison (adapted from Martin et al. (2016))

7.2.1 Similarity Computation

As mentioned previously, the computation of the similarity of cases is the main task when applying the CBR method. In the structural ICEBERG approach, the instances and relations representing the case characterisations of the learned cases and the query case are compared. The similarity measure of the query case with each learned case is computed. The similarity measure is a value between 0 and 1 (see 7.1):

$$sim(c_q, c_i): S^2 \rightarrow [0,1] \qquad \forall c_q, c_i \in S \tag{7.1}$$

The computation of the similarity in the ICEBERG approach is based on the global-local principle (Bergmann, 2002). The global similarity measures are defined on class level, and the local similarity measures are defined at the attribute level. In essence, the primitive attributes of the class instances in the same level are compared by individually defined similarity functions. Then, the global similarity measure aggregates all local similarity values into one value. As an example, the global similarity measure can be computed by aggregating the local similarity measures according to a defined aggregation function, as shown in 7.2:

$$sim(I_q, I_i) = A\left(sim(p_{q1}, p_{i1}), \ldots, sim(p_{qn}, p_{in})\right) \qquad (7.2)$$

where A is the aggregation function, I_q the query case instance, I_i the case instance in the case repository where the query case instance is compared and p the properties of the instances that are compared.

All attributes are individually weighted, and primitive attributes are represented by annotation and datatype properties with value types such as Integer or String. Object properties represent relational attributes. Since relational attributes might lead to a comparison of multiple referenced instances in the query case as well as in the learned cases, the global similarity function must be used as a set function, as proposed in Hefke and Abecker (2006a). The ICEBERG approach contains a cosine set function, as found in several frameworks (Recio-García, González-Calero and Díaz-Agudo, 2014; Hefke and Abecker, 2006a), and a more sophisticated similarity function, as commonly used in information retrieval (Cohen, Ravikumar and Fienberg, 2003; Witschel et al., 2015).

Staab (2011) identified two criteria based on which a similarity measurement should be chosen. The first criterion concerns the entity that should be compared. Staab (2011) mentioned *objects*, *concepts* or *ontologies* as possible entities. The second criterion is the goal of the assessment. It is the question whether it is a *numeric similarity* or a *preference ordering assessment*. During the configuration of the similarity model, as described in the next section, the mentioned criteria can be applied. The ICEBERG similarity framework is flexible enough to be enhanced with new global and local similarity functions. Besides the global-local similarity knowledge, the object-oriented case characterisation might contain taxonomic knowledge, which can be considered in a local similarity measure as well.

7.2.2 Similarity Configuration

The similarity model configuration in the ICEBERG approach is done in conjunction with the definition of concrete case characterisation dependent on the views and viewpoints (see Section 5.4). Figure 7.4 shows an exemplary similarity model configuration where the case viewpoints are omitted for simplification.

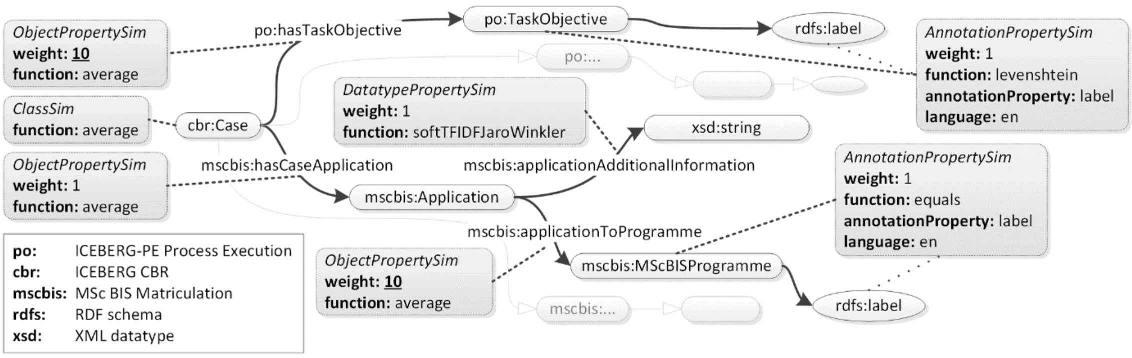

Figure 7.4: Exemplary Similarity Model Configuration[21]

As mentioned previously, the similarity model configuration is done in an ontology using elements from the similarity ontology. The ICEBERG approach and its instantiation provide a basic set of global-local functions. The following local similarity functions are currently available:

- Levenshtein: This function is a recursive field matching algorithm using Levenshtein edit-distance (Cunningham, 2009), which is the amount of edit operations required when changing one string into another.
- Equals: Equals is a strict equality of the values comparison.

[21] The ontology structure has been simplified, and the corresponding element denotation has been abbreviated to fit the figure.

- Jaro-Winkler: The Jaro-Winkler token-matching function, which is not based on edit-distance, performs well for short strings (Cohen, Ravikumar and Fienberg, 2003).
- SoftTFIDF: This is a TFIDF-based distance metric, which is extended with "soft" token-matching (Cohen, Ravikumar and Fienberg, 2003).

Listing 7.1 shows the mentioned possible local similarity functions (see `rdfs:range` element of the `owl:ObjectProperty` `sim:local-SimilarityFunction`). This local similarity function with the corresponding weight can be "attached" to classes as annotations to define what is to be considered for calculation by `sim:AnnotationPropertySimilarity`. If the to-be-considered property is a datatype property, the local similarity function including the weight can be "attached" to properties using `sim:DatatypePropertySimilarity`.

Listing 7.1: Local Similarity Functions of the ICEBERG Approach

```
sim:localSimilarityFunction
  rdf:type owl:ObjectProperty ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          sim:AnnotationPropertySimilarity
          sim:DatatypePropertySimilarity
        ) ;
    ] ;
  rdfs:range [
      rdf:type owl:Class ;
      owl:oneOf (
          sim:equals
          sim:levenshtein
          sim:jaroWinkler
          sim:softTFIDFJaroWinkler
        ) ;
    ] ;
  rdfs:subPropertyOf sim:similarityFunction ;
.
```

The following global similarity functions are currently available in the ICEBERG approach:

- Average: The average global similarity function is defined based on the weighted arithmetic mean, as shown in 7.3:

$$\bar{s} = \frac{\sum_{i=1}^{n}(w_i \times s_i)}{\sum_{i=1}^{n} w_i} \quad (7.3)$$

where n is the number of characterisation concepts (attributes) considered, w_i is the weight of attribute i and s_i is the value of the calculated local similarity that ranges between 0 and 1.

- Cosine: The cosine global similarity function is defined as proposed by Hefke et al. (2006).
- Probabilistic: The probabilistic global similarity function by Witschel, Martin, Emmenegger and Lutz (2015), as defined in 7.4, aggregates local similarities in a more conjunctive way than other functions.

$$P(i_1|i_2) = \prod_{k=1}^{n} P(J_{k1}|J_{k2})^{\alpha_k} \quad (7.4)$$

where J_{k1} are the set of instances that are linked to instance i_1 and the weights α_k reflect the relative influence of relationship (or attribute) r_k for determining the overall similarity.

Listing 7.2 shows the possible global similarity functions (see `rdfs:range` of the `owl:ObjectProperty sim:globalSimilarityFunction`). This global similarity function with the corresponding weight can be "attached" to properties using `sim:ObjectPropertySimilarity`. This encapsulates the global similarity function for object properties and allows for applying different similarity functions for the aggregation of multiple instances and local similarity measures. The root similarity annotation, the `sim:RootCaseClass-Similarity`, marks the start case class of the characterisation tree and "attaches" the overall global similarity function definition.

Table 7.1 shows an exemplary similarity computation based on the exemplary similarity model configuration as shown in Figure 7.4 using the admission process application scenario (without real case data for simplification). The example contains two learned cases (1 and 2) is a simplified manner and a query case. The computation starts, using the corresponding weights and functions, from the leaves and ends at the root with an overall similarity value for each of the learned cases.

Listing 7.2: Global Similarity Functions of the ICEBERG Approach

```
sim:globalSimilarityFunction
  rdf:type owl:ObjectProperty ;
  rdfs:domain [
      rdf:type owl:Class ;
      owl:unionOf (
          sim:ObjectPropertySimilarity
          sim:RootCaseClassSimilarity
        ) ;
    ] ;
  rdfs:range [
      rdf:type owl:Class ;
      owl:oneOf (
          sim:average
          sim:probabilistic
          sim:cosine
        ) ;
    ] ;
  rdfs:subPropertyOf sim:similarityFunction ;
.
```

Table 7.1: Exemplary Similarity Computation

Class	Instance	Property	Value/Reference	Weight	Function	Sim #1		Sim #2		Sim #3	
Case	$_queryCase	taskObjective									
		caseApplication									
	$case1	taskObjective	$case1Task	10,00	average					**0,14**	**0,21**
		caseApplication	$case1App	1,00	average					**0,96**	
	$case2	taskObjective	$case2Task	10,00	average					**1,00**	**0,92**
		caseApplication	$case2App	1,00	average					**0,08**	
Task	$_queryTask	label	"Eligibility Check"								
	$case1Task	label	"Eligibility History"	1,00	levenshtein	**0,14**	**0,14**				
	$case2Task	label	"Eligibility Check"	1,00	levenshtein	**1,00**	**1,00**				
Application	$_queryApp	string	"...FHNW..."								
		toProgramme	$_queryPro								
	$case1App	string	"...Text FH..."	1,00	softTFIDFJW			**0,57**	**0,96**		
		toProgramme	$case1Pro	10,00	average			**1,00**			
	$case2App	string	"...FHSG..."	1,00	softTFIDFJW			**0,84**	**0,08**		
		toProgramme	$case2Pro	10,00	average			**0,00**			
Programme	$_queryPro	label	"MSc BIS"								
	$case1Pro	label	"MSc BIS"	1,00	equals	**1,00**	**1,00**				
	$case2Pro	label	"MSc IM"	1,00	equals	**0,00**	**0,00**				

7.3 Case Adaptation

The case adaptation is usually the second step of the CBR cycle. Figure 7.2 shows the cycle of the ICEBERG-PE approach, where a new case is stipulated and transferred to a query case. This query case is used, after the retrieval (similarity computation) step, in the reuse phase (adaptation).

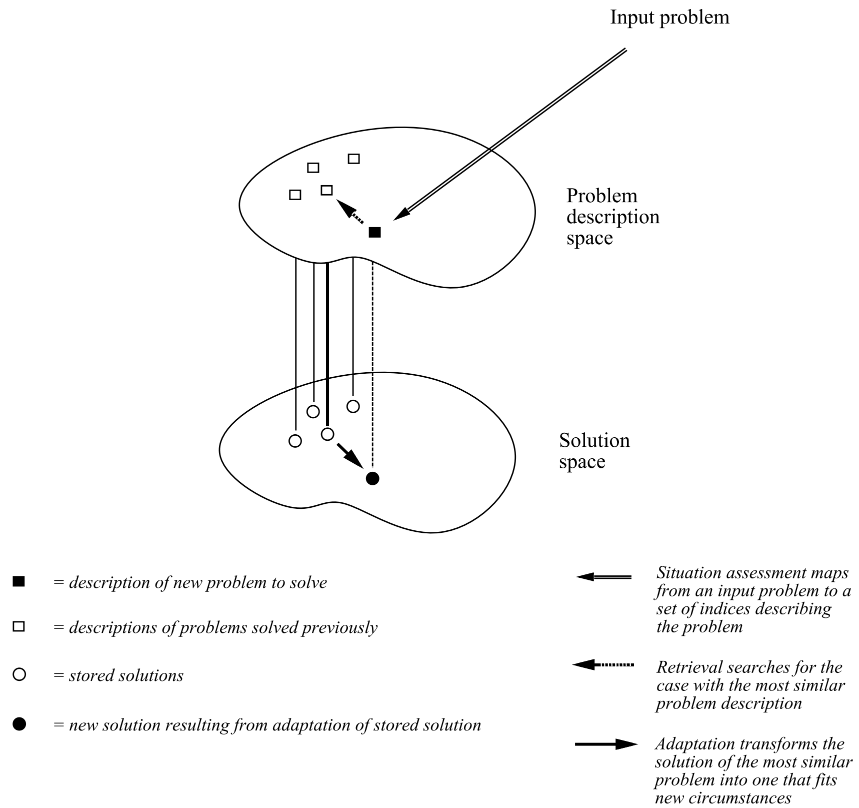

Figure 7.5: Adaptation Process in CBR (Leake, 1996)

Figure 7.5 shows the adaptation process in CBR. This adaptation transforms the case content (solution) to the query case (current problem). Using the ICEBERG approach, the knowledge worker can adapt the knowledge from the retrieved cases to the current case by either adding the information manually or running the adaptation mechanism. The default behaviour of the ICEBERG approach can be regarded as substitution adaptation (Definition 2.10). The adaptation mechanism can be extended in any direction, since the ICEBERG approach executes any pre-defined semantic adaptation rule of the current domain and application ontology. Even the default behaviour is a pre-defined rule, as shown in Listing 7.3, which can be extended in any direction.

Listing 7.3: Default ICEBERG Adaptation Rule

```
# LEVEL 2 Rule: Copy all case file items from selected source case,
but exclude replaced or removed items based on a previous rule
CONSTRUCT {
    ?newCase cbr:caseItem ?caseFileItem .
    ?newCase cbr:reusedCases ?case .
}
WHERE {
    ?newCase cbr:_automaticAdaptFromCase ?case .
    ?case cbr:caseItem ?caseFileItem .
    OPTIONAL {
        ?newCase cbr:caseItem ?newCaseFileItem .
        ?newCaseFileItem cbr:replacesCaseItemThroughAdaptionRule
            ?caseFileItem .
    } .
    FILTER (!bound(?newCaseFileItem)) .
}
```

Listing 7.3 shows a default rule behaviour where the rule copies all case file items from the retrieved and selected case (defined by the property `cbr:_automaticAdaptFromCase`) to the current query case, which becomes the solved case. The ICEBERG rule adaptation approach uses rule chaining as shown in Listing 7.3, where certain case content items are replaced or removed based on a previously executed rule, as shown as an example in Listing 7.4.

Listing 7.4 shows an exemplary adaptation rule based on the admission process application scenario. Although this rule concerns stakeholders as case file items, it is not restricted to stakeholders only. Any item of the case content can be modified, extended, added or deleted. It is even possible to change the case characterisation as well since the case characterisation and content is ontology-based and can be accessed by a semantic rule, as shown in Listing 7.4. The exemplary rule assesses if the previous case is an MSc IM case, if a specific stakeholder was responsible and if the current case is an MSc BIS case. Then the rule replaces the stakeholder with the responsible person of the MSc BIS application process. When applying an adaptation rule, it is important that the knowledge worker obtains a reasonable explanation for why something has been changed automatically. This explanation can be provided using the `cbr:adaptionRuleBasedComment` property, which is presented in the ICEBERG user interface to the knowledge worker.

Listing 7.4: Exemplary Adaptation Rule

```
# LEVEL 1 Rule: The previous case is an MSc IM case, Sarah was
responsible, and the current case is an MSc BIS case. -> Replace Sarah
with Neyyer, because he is responsible for MSc BIS.
CONSTRUCT {
    ?newCase cbr:caseItem ?newCaseFileItem .
    ?newCaseFileItem cbr:caseItemRepresentedBy mscbis:Neyyer_Admin .
    ?newCaseFileItem cbr:replacesCaseItemThroughAdaptionRule
        ?caseFileItem.
    ?newCaseFileItem cbr:adaptionRuleBasedComment "The previous case
        is an MSc IM case. However, the current case is an MSc BIS
        case. Therefore, Neyyer is responsible instead of Sarah." .
}
WHERE {
    ?newCase cbr:_automaticAdaptFromCase ?case .
    ?case mscbis:mscBISAdmissionCaseIsCharacterizedByApplication
        ?application .
    ?application mscbis:applicationToProgramme mscbis:MScIMProgramme .
    ?newCase mscbis:mscBISAdmissionCaseIsCharacterizedByApplication
        ?queryApplication .
    ?queryApplication mscbis:applicationToProgramme
        mscbis:MScBISProgramme.
    ?case cbr:caseItem ?caseFileItem .
    ?caseFileItem cbr:caseItemRepresentedBy mscbis:Sarah_Admin .
    BIND (URI(CONCAT("http://ikm-
        group.ch/MScBISOntology#case_item_infered_Neyyer_Admin")) AS
        ?newCaseFileItem) .
}
```

7.4 Case Evaluation and Learning[22]

The case evaluation and learning are usually the final two steps of the CBR cycle and are triggered by the knowledge worker itself. The outcome of the previous adaptation step is a potentially solved case, as shown in Figure 7.6. This solved case will be evaluated and revised by the knowledge worker, leading to a revised case. This revised case then can be passed on to the case repository using the automatic retention. This retention, by adding the case to the case repository, can

[22] The concepts and implementation presented in this section have not been published previously. The results in this section have been achieved by S. Emmenegger ([sic!] project researcher) and A. Martin (author of this thesis).

be regarded as case learning. Finally, the case is ready for the next CBR cycle as a learned case.

Figure 7.6 shows an extended CBR cycle. This ICEBERG cycle extension enables the knowledge worker to control the case editing process. Any solved, revised and even learned case can be transferred to an editable query case. This case transfer means that even a learned case can be edited at a later stage. This CBR cycle extension enables extended editing possibilities, as might be used during creation and extension of a generalised case (see Section 6.4.2.4).

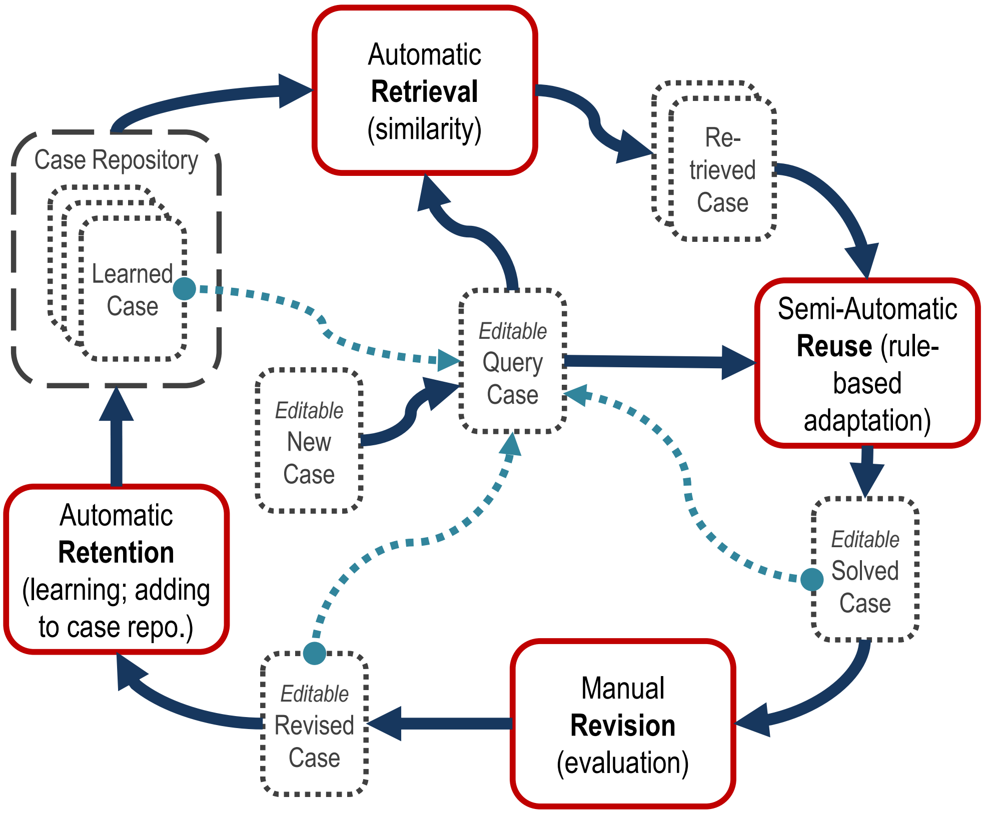

Figure 7.6: ICEBERG Cycle Extension

Listing 7.5 shows the possible case states, which are assigned to the cases: adaptation, revision and learned. Once a case receives the learned state `cbr:Learned_CaseState`, the case is available for retrieval. Otherwise, the case is just stored in the case repository but not available for retrieval.

Listing 7.5: ICEBERG Case States

```
cbr:CaseState
  rdf:type owl:Class ;
  rdfs:label "Case state"@en ;
  rdfs:subClassOf cbr:Element ;
.
cbr:Adaptation_CaseState
  rdf:type cbr:CaseState ;
  rdfs:label "Case in adaptation state"@en ;
.
cbr:Revision_CaseState
  rdf:type cbr:CaseState ;
  rdfs:label "Case in revision state"@en ;
.
cbr:Learned_CaseState
  rdf:type cbr:CaseState ;
  rdfs:label "Learned case"@en ;
.
```

7.5 Conclusion

This chapter introduces the CBR services, which are used to run the ICEBERG-PE approach. Additionally, this chapter provides a conceptual answer to the following research questions, which are showcased by the implementation of the ICEBERG prototype and evaluated by the triangulated data sources:

- The case similarity section provides an answer to how the similarity of knowledge work can be calculated (RQ 3.1). The introduced similarity mechanism uses case characterisation, which characterises the knowledge work description.
- The case adaptation sections explain how domain knowledge and contextual information are used for retrieval and adaptation (RQ 3.2). Through the inclusion of enterprise ontology, the ontology-based similarity model and the semantic adaptation rules, domain knowledge and contextual information can be seamlessly integrated.
- Finally, this chapter closes with an innovative approach that allows the knowledge worker to control the CBR cycle and enables related or generalised case creation.

The next chapter describes the prototypical implementation of the ICEBERG-PE approach and answers research questions 1 and 3.

8 Ontology-based CBR and Process Execution Prototype

This chapter provides an answer to the following research questions from a profound technical perspective:

RQ 1: How can case-based reasoning be integrated with process execution?

RQ 3: How can case-based reasoning services support process execution?

The chapter starts with a delineation of the prototype environment, which consists of third-party software components and tools. Finally, the chapter shows the prototypical implementation of the ICEBERG-PE approach as described in the previous chapters.

8.1 Prototype Environment

The 'prototype environment' is composed of existing and third party software components, which are used to realize the approach and to develop and to embed the demonstrative artefact.

The selection of software components is sustained and grounded on certain constraints. This selection of environmental components affects the entire software architecture of the instantiation. To ensure the quality of a software architecture Zörner (2012) and Hruschka (2012) suggest relying on certain constraints, which are made explicit, based on the arc42 guidelines (Starke and

Hruschka, 2011) and template (Starke and Hruschka, 2012) for software architecture. These constraints will justify the selection of environmental components and frame the "freedom of design decisions or the development process" (Starke and Hruschka, 2012). In the flowing the general (required and optional) constraints for the selection of environmental components are listed in Table 8.1:

Table 8.1: Constraints for Environmental Software Components (SC)

Constraint	***Explanation***
SC-1: Source code access (recommended)	To ensure debugging deep capability, component vendors should provide full access to source code (preferably open source).
SC-2: License	The instantiation of the thesis work must be available for future work. Therefore, it is advised that all software components are based on a public license (preferably open source).

In addition to the general constraints for environmental components, the following constraints frame software components, which will be coupled/integrated into the environment by library linking or source code compiling:

Table 8.2: Constraints for Environmental and Coupled Software Components (CSC)

Constraint	***Explanation***
CSC-1: JVM based (recommended)	Software components should run on a Java Virtual Machine (JVM).
CSC-2: Java API	Software components must provide a Java API to in co-operating with the CBR component.
CSC-3: Programming language (recommended)	To ensure maintainability, the source code should be available in the Java programming language

Besides the general constraints for all environmental software components, additional contestants for a workflow management system are listed in the following:

Table 8.3: Constraints for Workflow Management System (WMS)

Constraint	***Explanation***
WMS-1: BPMN compatibility	A potential workflow engine, as part of a workflow management system, needs to provide BPMN compatibility and native support to reflect the application scenario.
WMS-2: Execution engine	It is advisable that a workflow engine, as part of a workflow management system, can execute BPMN process diagrams without a manual or semi-automatic conversion into an executable representation.
WMS-3: Task list	A workflow management system needs to provide an extendable task list to demonstrate the possible user involvement when consulting a CBR approach.
WMS-4: User and groups	A user and group management are required to demonstrate a different user involvement.
WMS-5: Instance management (recommended)	To enhance the development process, debugging possibility and runtime inspection, a workflow management system should provide the inspection of process instances.
WMS-6: Runtime container	A server-based implementation is required to run a workflow management system in a multi-user environment. Additionally, it is advised that a workflow engine, as part of workflow management system, is accessible using an HTTP interface, and it provides an HTML based interface. Therefore, a web server (servlet container) is required.
WMS-7: Java EE integration (recommended)	Tt is recommended that a workflow engine runs in a Java EE environment to reduce the implementation effort.

8.1.1 Workflow Management System: Camunda Engine, Tasklist, Cockpit and Admin

As mentioned before, the decision for choosing a workflow management system has been made by qualitative criteria (certain (boundary) constraints) from the given application scenario and the technical requirements for implementing the introduced approach.

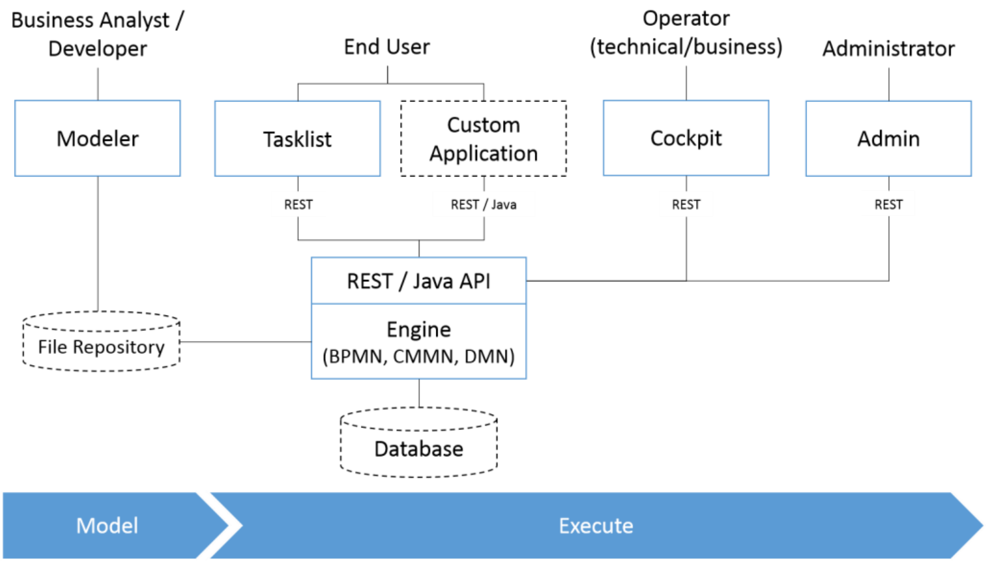

Figure 8.1: Architecture of Camunda BPM[23]

Camunda BPM is an open source workflow management system and a fork of the BPM project Activiti. It is mainly driven by a German consulting and software company called Camunda were the name roots. Camunda BPM reflects the constraints for a workflow management system, since it is a BPMN-based (WMS-1) workflow engine (complies WMS-2) and contains a task list (complies WMS-3), an administration component for the user and group management (complies WMS-4) and a cockpit for runtime process instance inspection (complies WMS-5). Apart from that, the Camunda engine runs in a servlet based container (complies WMS-6) and in a Java EE environment (complies WMS-7). Finally, Camunda BPM is JVM based (complies CSC-1), written in Java (complies CSC-3)

[23] Retrieved from [16-03-2016]: https://camunda.org/features/

and provides and Java based API (complies CSC-2). The source code is hosted and available (complies SC-1) from GitHub[24] based on the Apache license (complies SC-2). Figure 8.1 shows the architecture of the whole Camunda BPM suite containing the components mentioned before including a modelling environment (see Section 8.1.2).

8.1.2 Workflow Modelling: Camunda Modeler

The Camunda Modeler (see Figure 8.1) is a modelling tool for executable BPMN process models. It provides certain workflow specific extensions for the Camunda engine. Beside the BPMN modelling the tool also can be used for DMN modelling and is licensed under an open source MIT license (complies SC-2) and a JavaScript based modelling engine. The code itself is hosted on GitHub[25] (complies SC-1).

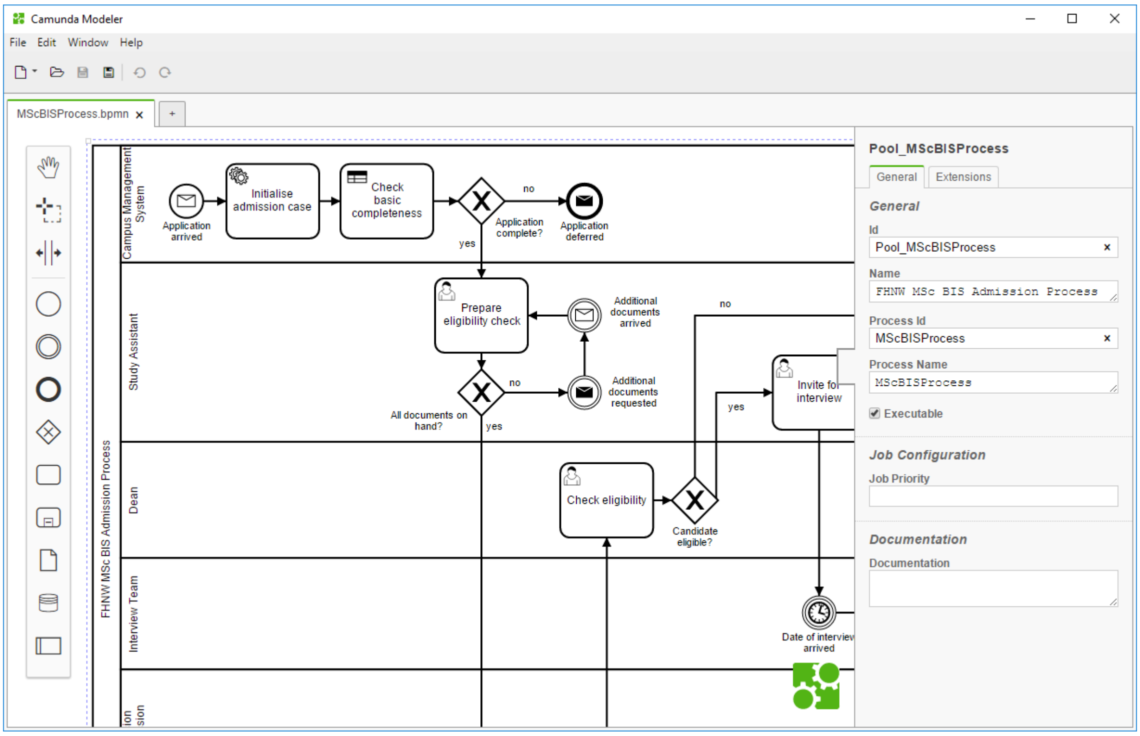

Figure 8.2: Screenshot of the Camunda BPMN Modeler

24 https://github.com/camunda/camunda-bpm-platform

25 https://github.com/camunda/camunda-modeler

8.1.3 Semantic Framework: Jena

Jena (McBride, 2001) is a semantic web framework developed in Java, which is open source, originally developed by the HP Labs and now maintained by the Apache Software Foundation. The framework provides an API for creating an abstract model from RDF graphs, which can be queried by SPARQL 1.1 queries. Additionally, Jena provides partial (not OWL full) OWL reasoning support and can be enhanced with external reasoners. The source code is hosted on GitHub[26] (complies SC-1) and released under an Apache License (complies SC-2).

8.1.4 Ontology Modelling: TopBraid Composer

The TopBraid Composer is a modelling environment for ontologies and a commercial industry-level product of the TopQuadrant company. The composer is closed source and available in three versions (free, standard and maestro). Since the modeller is fully W3C compliant and uses Jena (see 8.1.3) for maintaining ontology models internally, source code access is not needed – the modeller can be exchanged with any other W3C compliant ontology modelling environment. The modelling environment is used for the ontology engineering, CBR configuration, and data integration.

8.1.5 Ontology Inferencing: TopBraid SPIN API

The TopBraid SPIN API is a Java-based software component for executing SPIN rules. SPIN is an SPARQL-based rule and constraint notation for W3C Semantic Web ontologies, and it is a W3C member submission (Knublauch, Hendler and Idehen, 2011). The source code of the SPIN API is distributed under an Apache License (complies SC-2) and available (complies SC-1) from a private repository[27] of TopQuadrant. The SPIN API is used by the ICEBERG (see 8.2.1) component in the retrieval computation and semi-automatic adaptation tasks.

---

[26] https://github.com/apache/jena

[27] http://topquadrant.com/repository/spin/

8.1.6 Process Fragment Modelling: OMiLAB bpFM Modelling Toolkit

In Section 6.4.2.3 BPFM is introduced as a possible way to model process fragments, which can be used as case content. When modelling process fragments, the bpFM modelling toolkit[28] (see Figure 8.3) can be used. This bpFM modelling toolkit is an OMiLAB project and affiliated with the University of Camerino, and provides the knowledge worker with a compliant way to model BPFM based process fragments.

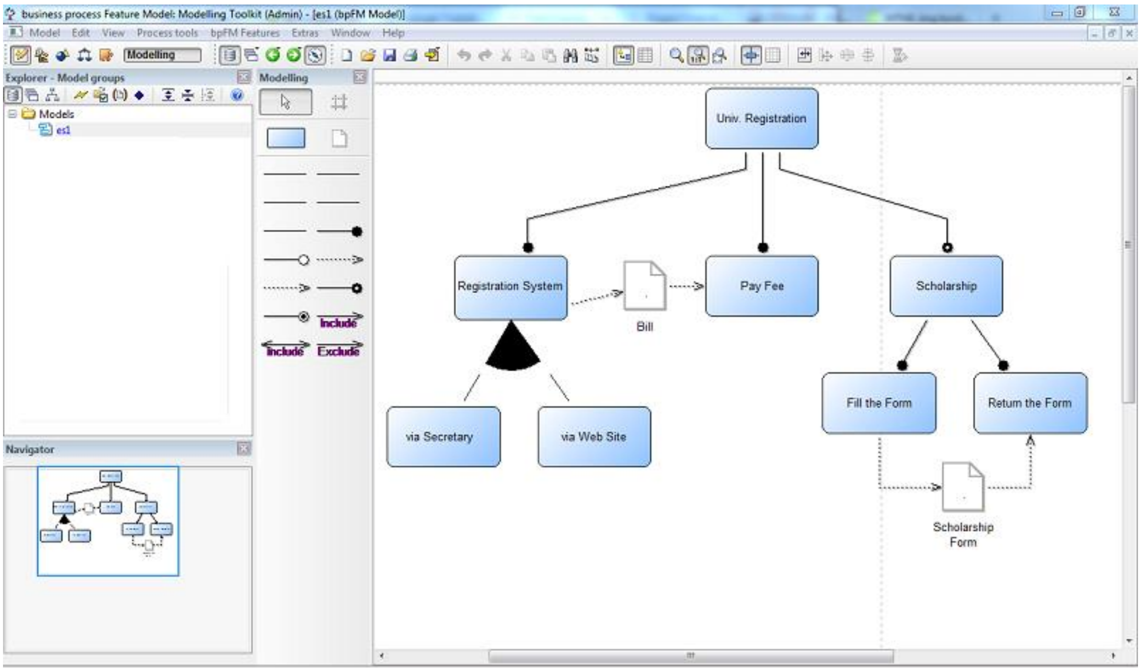

Figure 8.3: Screenshot of the bpFM Modelling Toolkit[29]

8.2 ICEBERG-PE Approach Instantiation

This section describes the prototypical instantiation of the ICEBERG-PE approach of this thesis as a toolkit.

The integrated case-based reasoning (ICEBERG) toolkit is the core CBR instantiation of the described approach in this thesis. It has been developed by the author of this thesis and researchers of the applied research project [sic!]

[28] http://austria.omilab.org/psm/content/bpfm

[29] http://austria.omilab.org/psm/content/bpfm/info?view=details

based on the conceptual foundations of this thesis. The demonstrator has been named as ICEBERG toolkit, branded using a logo as shown in Figure 8.4, and part of the source code has been made publicly available[30] and licensed under the Apache License Version 2.0.

Figure 8.4: ICEBERG Toolkit Logo

The system architecture in Figure 8.5 depicts the ICEBERG and related components of this instantiation. After the conceptual architectural description, two technical deployment descriptions, one deployment diagram of the ICEBERG toolkit and one deployment diagram of the process execution extension (ICEBERG-PE), are provided. The ICEBERG-PE toolkit is an extension to the ICEBERG toolkit for process execution.

The ICEBERG toolkit is structured on four layers: data, persistence, business and presentation. The data layer consists of the ICEBERG triple store and a logical representation of the file system.

- File repository: The file repository element is a logical representation of a server filesystem, which manages and maintains the document case items, the modelled process fragments and the process models of the overall workflow (process execution). The documents and the process fragments can be attached to a case using the ICEBERG GUI (graphical user interface) and finally persisted in the file repository. The process models can be created using a process modeller and stored directly in the file repository. These process models are available for process execution in the process engine.

[30] https://bitbucket.org/account/user/ikmgroup/projects/ICEBERG

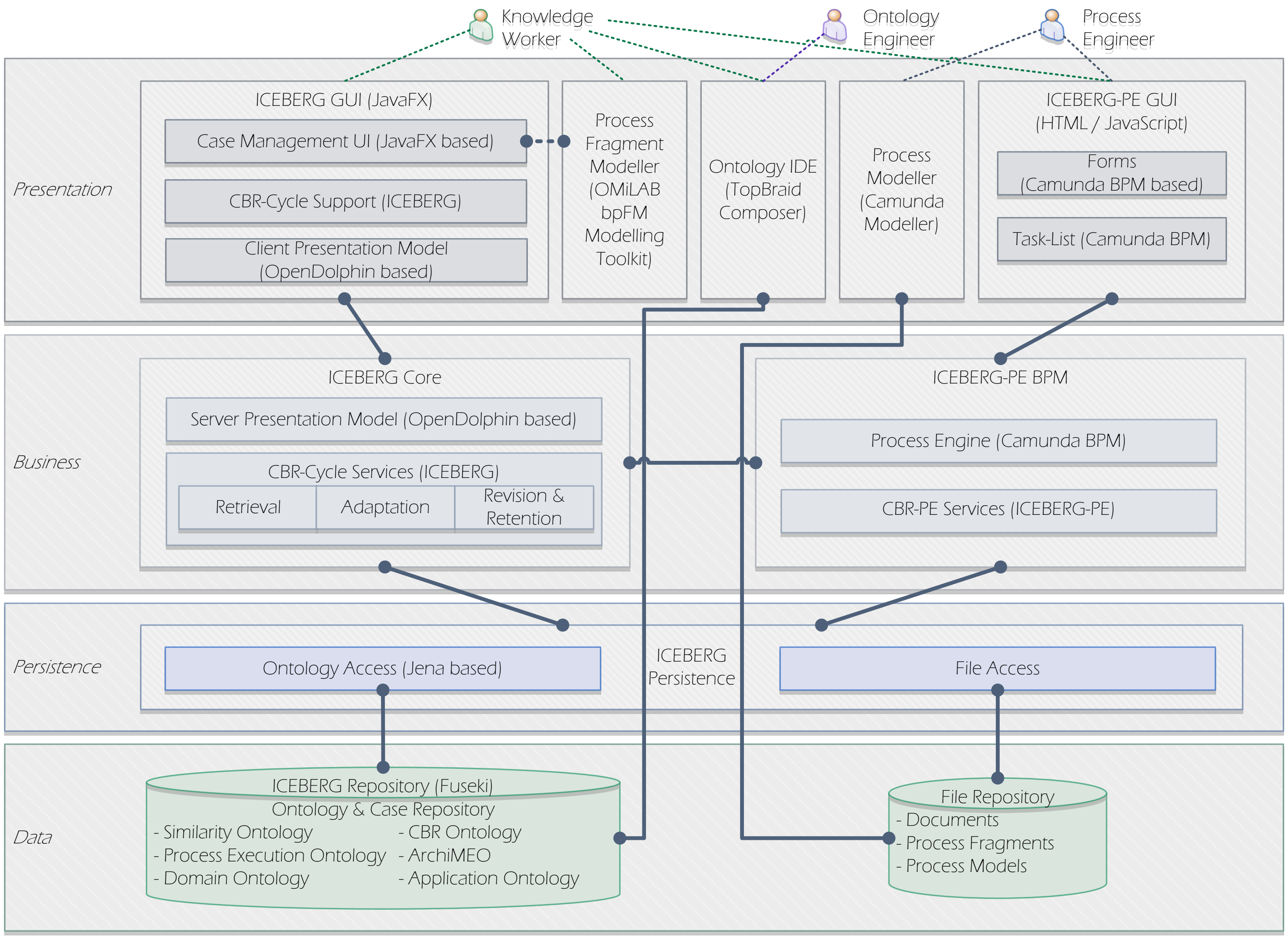

Figure 8.5: ICEBERG-PE System Architecture

- ICEBERG repository: The ICEBERG repository contains the ICEBERG ontology including the cases of the CBR system and is an instantiation of the Apache Jena Fuseki triplestore and SPARQL endpoint server. However, the ICEBERG repository is not restricted to the Fuseki[31] implementation. Any semantic graph database with an Apache Jena interface can be used instead. Since the case model of the ICEBERG approach is entirely ontology-based, all case data is stored in the triple store, except the document files and a graphical representation of the process fragments, which are stored in the file repository.

The persistence layer consists of a logical ICEBERG persistence component, which can be assessed from the ICEBERG core and ICEBERG-PE BPM components.

- ICEBERG Persistence: The ICEBERG persistence component ensures the ontology and file access. The ontology access is based on Apache Jena and used to access the ICEBERG repository. The file access element acts as a data access object component, which ensures the file access from the file repository. Moreover, the file access component is used to load and transfer the executable process models to the process engine.

The business layer consists of the ICEBERG core and the ICEBERG-PE BPM components. This business (logic) layer is an execution element of the CBR methodology using a programmatic implementation.

- ICEBERG Core: The ICEBERG core component encapsulates the core implementation of a complete CBR cycle. The CBR- cycle services build the heart of the ICEBERG core. These CBR-cycle services contain a retrieval component for the similarity computation of the cases. The adaptation component executes the semantic rules (SPIN rules) during

[31] https://jena.apache.org/documentation/fuseki2/

case adaptation. Finally, the revision and retention component supports the final life cycle steps of the cases. The server presentation model is the server part of a shared presentation model, which contains loaded and the to be transferred case data. This server presentation model is based on OpenDolphin[32], which is a remoting framework.

- ICEBERG-PE BPM: The ICEBERG-PE bpm component is the process execution element of this CBR approach. The CBR-PE services can access the CBR-Cycle services from the ICEBERG core component. This access is realised by interfaces between the CBR components retrieval and retention, and the process execution engine. The interface is implemented using contexts and dependency injection (CDI), which allows the injection of workflow data to a service that invokes the CBR-Cycle services. The process engine executes an executable process model. This process engine is based on Camunda BPM and executes BPMN, CMMN and DMN models. The process engine can invoke the CBR-PE services using CDI. The invocation can be described in an executable BPMN or CMMN model using services tasks or other elements (listener based).

The presentation layer consists of the ICEBERG GUI and ICEBERG-PE GUI elements as well as of third party software. These presentation layer components provide to the knowledge worker, the ontology engineer and the process engineer various interfaces for their work.

- ICEBERG GUI: The ICEBERG GUI is the user interface of this prototype, which is based on the JavaFX user interface framework. The knowledge worker can access the cases using the case management UI, which is also written using the JavaFX user interface framework. The knowledge worker can retrieve, reuse, revise and retain cases within the case

[32] http://open-dolphin.org

management UI in conjunction with the CBR-cycle support component. This CBR-cycle support component performs the retrieve, reuse, revise and retain phases from the user interface perspective. Finally, the client presentation model is the counterpart of the server presentation model mentioned before.

- ICEBERG-PE GUI: The ICEBERG-PE GUI is the user interface to the process execution engine for the knowledge worker. This user interface is HTML and JavaScript based and is maintained by the process engineer. The ICEBERG-PE GUI consists of web forms and a workflow task list, both based on the Camunda BPM. The web forms provide the knowledge worker an interface for assessing and manipulating workflow data. While the task list provides the knowledge worker with an overview over the tasks, which need to be done.
- Process Modeller: The process modeller is a third-party component from Camunda for creating BPMN, CMMN and DMN models by the process engineer. The created models will be stored directly in the file system.
- Ontology IDE: The ontology IDE is a development environment for maintaining and creating ontologies. The knowledge worker configures the case model and maintains the domain and application ontology with the help of the ontology engineer. The ontology IDE is also a third-party application, the TopBraid composer. The ontology IDE can access the ontologies from the ICEBERG repository.
- Process Fragment Modeller: The process fragment modeller is a third-party application, which is used for creating BPFM based process fragments in an integrated environment. This process fragment modeller is an instantiation of the OMiLAB bpFM modelling toolkit, which is used by the knowledge worker. The knowledge worker can freely create new

BPFM models and assign these models to a case as case content in the case management UI.

8.2.1 Integrated Case-based Reasoning (ICEBERG) Toolkit

In the following, the implementation and technical structure is depicted based the UML deployment diagram as shown in Figure 8.6. The ICEBERG deployment diagram consists of artefacts, execution environments and one component. The artefacts are named with a .war (Web Application Archive) or .jar (Java Archive) extension. These extensions are a technical definition expressing that such an element is an archive, which contains Java class libraries. For an adequate illustration and description, the artefact names might be slightly different to the ones used in the source code. Moreover, some artefacts are grouped into one artefact for an adequate illustration and description as well. The ICEBERG toolkit consists of the three execution environments: repository, server and GUI. The execution environment is a node element representing software on a device that executes software code of the corresponding artefacts.

The ICEBERG repository is running on an Apache Tomcat servlet and web container (broadly speaking a web server). This ICEBERG repository is the implementation of the ontology and case repository as introduced before containing the ontology schema and instances, and the learned cases including the cases, which are currently in process.

- ICEBERG Ontology Triplestore: The ICEBERG ontology triplestore artefact is a web archive containing the ICEBERG ontology structure and imports the Apache Jena Fuseki library (third-party). As mentioned before the ICEBERG toolkit can run using any triplestore that consists of a Jena interface. In this case, ICEBERG ontology triplestore artefact can be deployed on any servlet container. This triplestore artefact can be extended with the TopBraid SPIN API and further CBR mechanisms in case server load balancing is required.

The ICEBERG server is running on an Apache Tomcat servlet and web container as well. This ICEBERG server consists of the following own (not third-party imported) artefacts, and one emphasised component:

- CBR Core: The CBR core artefact is the main artefact of the whole toolkit and contains all the source code for execution the similarity computation, adaptation rule execution, case lifecycle and viewpoint management. Since this ICEBERG approach is entirely ontology-based, the CBR core element requires an ontology manipulation framework, which is, in this case, the Apache Jena (imported third-party artefact) framework for building Semantic Web applications. TopBraid SPIN API is an additional imported third-party artefact, which enables the CBR core element to execute semantic rules written in the SPARQL Inferencing Notation (SPIN). These semantic rules are used during semi-automatic case adaptation and as a preparation for similarity computation. The emphasised ICEBERG ontology TTL I/O component is depicted as an alternative to the ICEBERG repository during ontology engineering. Through this component, it is possible to store and access the ontology schema and instances using TTL (Turtle: Terse RDF Triple Language) files. This TTL file access allows a simplified ontology engineering and a concurrent ontology version control.
- CBR Core Service: The CBR core service artefact encapsulates the CBR core artefact as a service and provides an accessible interface. Moreover, the CBR core service artefact provides serializable value objects (VO), sometimes called data transfer objects (DTO), representing the complex ICEBERG case data and schema structure.

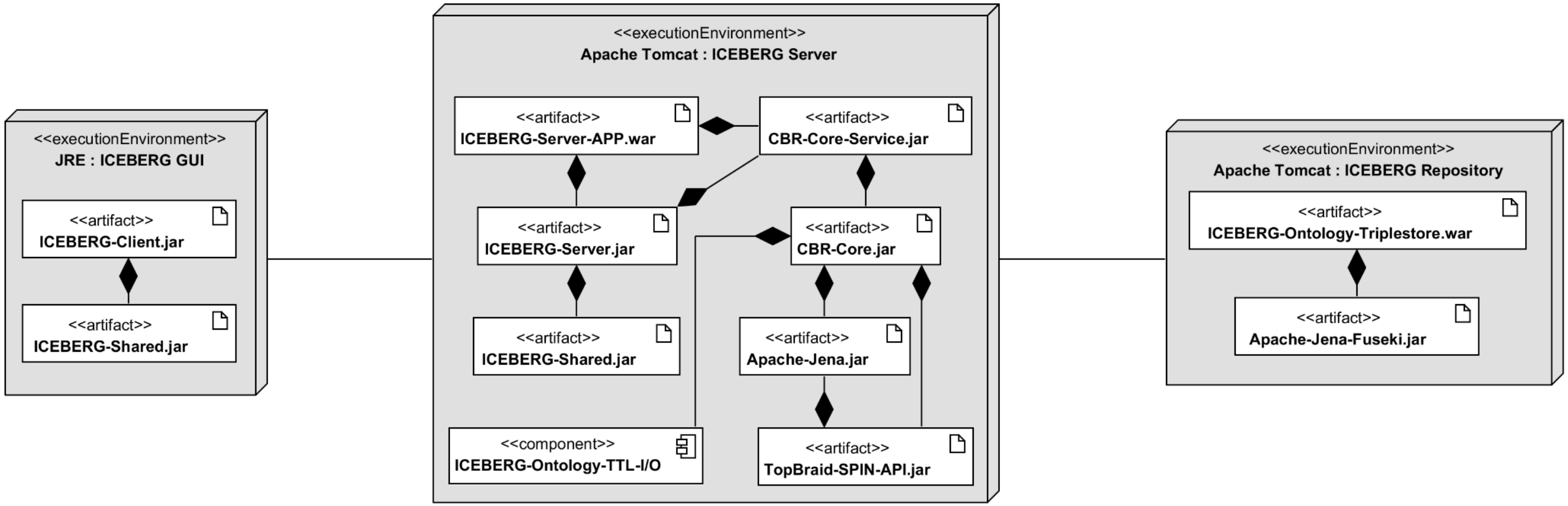

Figure 8.6: ICEBERG Deployment Diagram

- ICEBERG Server: The ICEBERG server artefact imports the CBR core service artefact and acts as server part for the corresponding ICEBERG GUI. Moreover, the ICEBERG server artefact contains server controller actions, which are triggered by commands mainly from the ICEBERG GUI. The server artefact implements the server presentation model (OpenDolphin based), which is a self-contained model that contains the data, states and actions of the user interface on the server side.
- ICEBERG Shared: The ICEBERG shared artefact exists on the server as well as the GUI instantiation. On the server side, it is imported by the ICEBERG server and provides classes and configuration files, which is used for sharing knowledge and definitions between client and server.
- ICEBERG Server APP: The ICEBERG server app exposes an application servlet from the ICEBERG server artefact using an Apache Weld servlet listener. The whole ICEBERG server app (war file package) including Apache Weld can then be deployed on an Apache Tomcat acting as the ICEBERG server. This ICEBERG server app can then be accessed from the ICEBERG GUI over HTTP(S).

The ICEBERG GUI runs on any device with a JRE (Java Standard Edition Runtime Environment) execution environment and consists of the client and the shared artefact.

- ICEBERG client: The ICEBERG client artefact contains the graphical user interface of the ICEBERG toolkit. This user interface provides to the knowledge worker the possibility to manage the cases and access all CBR cycle functionalities. The ICEBERG client artefact uses the ICEBERG shared artefact for knowledge exchange between client and server as mentioned before. Moreover, the ICEBERG client is developed using the JavaFX framework for rich internet applications (RIAs) and is built as a single-click executable fat JAR (a JAR artefact containing all required classes to execute).

8.2.2 ICEBERG for Process Execution (PE) Toolkit

The ICEBERG toolkit, as it is designed and conceptualised based on this thesis and related research work (see Section 4.1 and Section 9.4.1), can be applied and reused in different uses cases. The ICEBERG-PE toolkit is an extension to the ICEBERG toolkit for process execution. In the following, the implementation and technical structure of the ICEBERG-PE toolkit is described based on the UML deployment diagram as shown in Figure 8.7. This section contains a description of artefacts, which are new or modified compared to the ICEBERG toolkit (see Section 8.2.1 for a description of the equivalent artefacts).

The ICEBERG-PE toolkit consists of the four execution environments: repository, core, BPM and GUI. The ICEBERG-PE server core accesses the ICEBERG repository to create and modify cases in the case repository and accesses the ICEBERG ontology including schema and instances. The ICEBERG-PE server core is running on an Apache WildFly application server (a web server where the Java EE application framework and further server management tools are provided). Unlike the ICEBERG-Toolkit, the ICEBERG-PE server core execution environment consists of the ICEBERG-PE core artefact.

- ICEBERG-PE Core: The ICEBERG-PE core artefact exposes an application servlet from the ICEBERG server artefact. Moreover, the ICEBERG-PE core artefact imports the CBR core service artefact and exposes the service interfaces specifically for the process execution. This separated structure allows fine-grained services and autonomous modules running in their own context, without losing the stateful, stateless or singleton EJB capability and transaction management. The invocation can be done based on JNDI API lookups, using a Web service implementation or any other application server specific remoting implementation on the client side.

The ICEBERG-PE server BPM is running on an Apache WildFly application server with a bootstrapped Camunda BPM instantiation. This bootstrapping is depicted using the abstract Camunda BPM artefact. In fact, the Apache WildFly Camunda BPM bootstrapping requires several configuration steps but enables the Camunda tool stack like a task-list, the underlying process execution engine or further process execution management tools (see Section 8.1.1). As mentioned, the ICEBERG-PE approach is not restricted to Camunda BPM - it is just a possible implementation of a process execution environment. Nevertheless, the ICEBERG-PE BPM artefact of the ICEBERG-PE server BPM is specific to Camunda BPM.

- ICEBERG-PE BPM: The ICEBERG-PE BPM artefact uses the CBR core service artefact (relation not shown in Figure 8.7) and the Camunda API (third-party) as an import for providing the following features. The ICEBERG-PE BPM artefact provides HTML/JavaScript-based user task forms. Further, ICEBERG-PE BPM artefact provides an EJB client implementation to access the ICEBERG-PE server core with the CBR functionality. Moreover, ICEBERG-PE BPM artefact provides CDI managed services, which can be invoked from BPMN service tasks or any Camunda listener defined in a BPMN or CMMN model. The task forms and the CDI managed services are specific to an application scenario and the corresponding business process.

The knowledge workers can interact with the ICEBERG-PE toolkit using a web browser showing an HTML/JavaScript-based interface to a process execution instantiation. The task forms contain workflow specific data, process execution and case data that are provided from the ICEBERG repository. The knowledge worker can use the ICEBERG GUI to access the same case data but with the extension of all the ICEBERG CBR-specific functionalities.

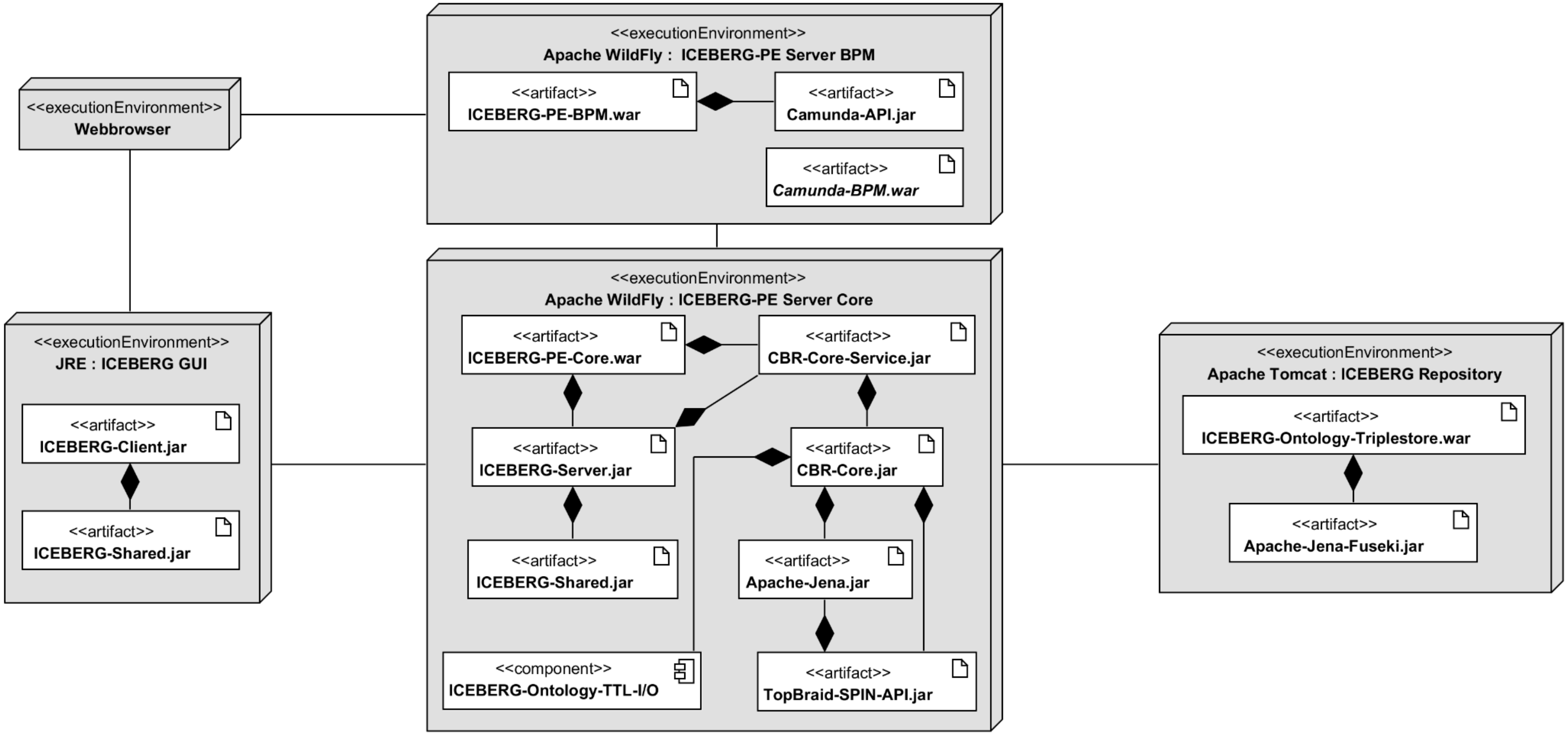

Figure 8.7: ICEBERG-PE Deployment Diagram

8.3 Conclusion

This chapter introduces the ICEBERG and ICEBERG-PR toolkit, which act as an instantiation and demonstrator of the ICEBERG-PE approach. Moreover, this chapter provides a profound technical description of the prototype. Besides, this chapter provides an answer from a technical perspective to the following research questions, which are showcased and evaluated by the triangulated data sources in the evaluation chapter next.

- The CBR-Cycle and the CBR-PE services show how CBR can support the process execution (RQ 3) from a technical perspective, although this is further evaluated and showcased in the evaluation chapter next.
- Finally, it is possible to conclude that the ICEBERG-PE server and the GUI provide an integration of CBR and process execution (RQ 1).

The next chapter describes the overall evaluation of the ICEBERG-PE approach and the prototype as described in this chapter.

9 Evaluation

This chapter concerns the evaluation of the introduced ICEBERG-PE approach and the corresponding instantiation. The chapter starts with a brief description of the data sources for this evaluation and is followed by the summative evaluation based on the ICEBERG-PE procedure model and a showcased example using the ICEBERG-PE instantiation. Finally, the evaluation chapter contains a confirmatory evaluation using two related contexts. Based on this evaluation, the answers (suggestion) to the research questions of this thesis is confirmed and concluded in the Chapter 10.

9.1 Data Source for Evaluation

To ensure the validity of the evaluation, evaluation data source triangulation has been applied. The data source triangulation of the ICEBERG-PE approach, instantiation and procedure model, as shown in Figure 9.1, is ensured by methods (method triangulation; demonstration and qualitative interviews), data sources (source triangulation)

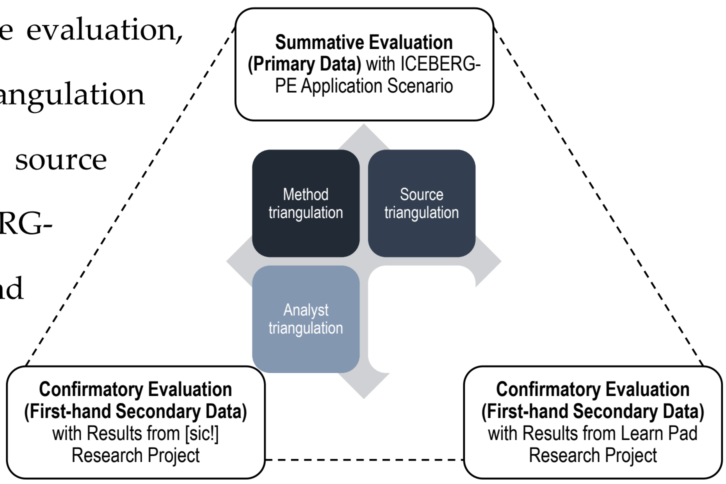

Figure 9.1: Evaluation Data Triangulation

and analysis (analyst triangulation; apart from myself, other research project members supported the analysis).

The first and most comprehensive evaluation data source is the summative evaluation based on the ICEBERG-PE procedure model and a showcased example using the ICEBERG-PE instantiation, followed by two confirmatory first-hand secondary evaluation data sources. “First-hand” means that the results are gathered by the author of this thesis within a project team. Yet this data is considered to be secondary data because it was gathered in a different research project with a similar application scenario. The confirmatory evaluation results are coming from the [sic!] research project and the EU research project Learn PAd. This confirmatory evaluation provides additional evidence and confirmation of the validity of the ICEBERG approach itself, as well as confirms that the ICEBERG approach is generalised and transferable to other contexts.

9.2 Summative Evaluation of Procedure Model

This summative evaluation section is based on the ICEBERG-PE procedure model and is showcased using the admission process application scenario. This procedure model based evaluation and demonstration starts with step 8, the domain/application ontology, of the ICEBERG-PE procedure model. The previous steps are described in the application scenario and the previous chapters.

9.2.1 Domain / Application Ontology

The first step in this evaluation setting is the creation of the domain and application ontology. This ontology has been derived from the application scenario, the admission process. This domain/application ontology creation is done with stakeholder involvement, as suggested in the procedure model. Figure 9.3 shows a visualisation of the domain/application ontology of the admission process setting called "MSc BIS Ontology". The visualisation only contains elements, concepts and properties from the "mscbis" namespace, for readability purposes. The "MSc BIS Ontology" imports elements from the ArchiMEO ontology as well, such as elements from the "eo" (Enterprise Ontology), "top" (Top Level Ontology) and "archi" (ArchiMate) namespaces. The appendix Listing-Appx 1 contains the complete "MSc BIS Ontology", including its relations to the ArchiMEO ontology. The "MSc BIS Ontology" is interlinked with the ArchiMEO ontology and contains elements for describing the admission process scenario, such as the application and the applicant, which may have an academic qualification and further personal data. Further, the academic qualification may be documented based on a certain bachelor degree, which has been awarded by a university with a certain rating. Such an application ontology can exist independent from an ICEBERG-based setting. The application ontology and the ArchiMEO elements represent the enterprise ontology of admission process entity (the MSc BIS programme) and serve as the basis for the case-based reasoning characterisation vocabulary. The enterprise ontology does not define the case characterisation, content and model itself. This case model definition will be done by annotating selected elements (classes and properties) of the underlying enterprise ontology as described in the next section.

9.2.2 Similarity and Adaptation Model

The characterisation is defined by configuring the similarity model. This configuration can be done by annotating the properties and classes as described in Section 7.2.2. This configuration can be done using the similarity ontology of the ICEBERG approach in conjunction with an ontology engineering editor, as shown in Figure 9.2. The configuration reflects the viewpoint model as described in Section 5.4, which means that each similarity configuration belongs to a certain case view, concern and, finally, role. Figure 9.4 shows a visualisation of the case model containing the initial case characterisation elements (depicted by the "case is characterised by" properties rooting from the admission case) and imported elements from the MSc BIS ontology. Listing-Appx 2 represents the case model itself, which only contains a few elements, such as the case definition itself (line 24 to 31), the case file items (line 40 to 44), the concerns (line 87 to 96), the initial case characterisation elements (line 45 to 62) and finally the process execution elements (line 32 to 39). As mentioned previously, the configuration is then done using the viewpoint model. Figure 9.5 visualises a fragment of the MSc BIS case similarity model configuration, as completely represented in Listing-Appx 3. This exemplary similarity model configuration reflects two viewpoints (eligibility and admission process view). The eligibility view is framing the view for the interview team and its task for verifying the eligibility of a prospective student. The admission process view considers the need of the study assistant to have an overview of the whole admission process since he or she is involved in the entire process. Figure 9.5 shows how the configuration can be done by annotating similarity values, such as functions and weights, to properties or classes.

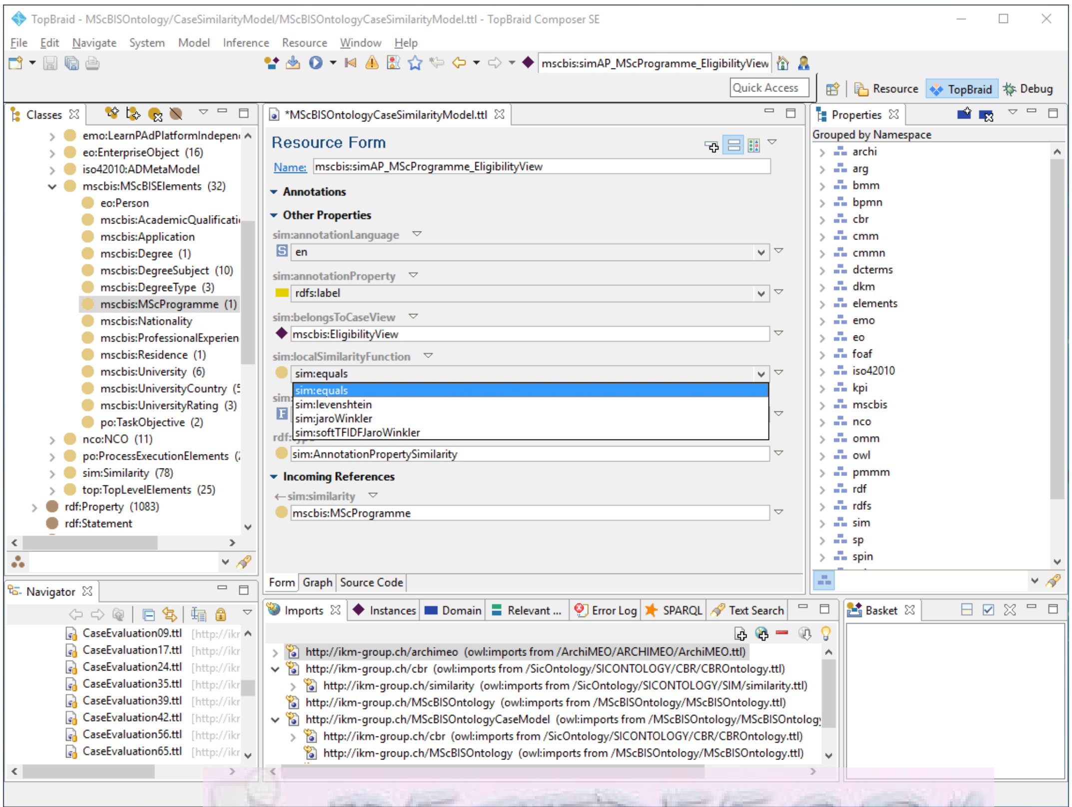

Figure 9.2: Similarity Model Configuration using an Ontology Engineering Editor (such as the TopBraid Composer)

9.2.3 CBR and Process Execution Configuration

The ICEBERG toolkit (see Section 8.2) is highly generic and can be entirely configured by the similarity model, as described in the previous section. This similarity model-based configuration means that no additional configuration step is required for the ICEBERG toolkit itself.

To run the ICEBERG-PE prototype, an executable process model, such as a BPMN-based workflow including workflow variables, is required. Figure 9.6 shows a visualisation of the BPMN-based workflow implementation of the admission process (see admission process description in Section 4.2). The "prepare-eligibility-check" and "validate-eligibility" activities are integrated with ICEBERG CBR services. Listing-Appx 4 shows an exemplary implementation of an ICEBERG-PE BPM service, which consumes the CBR core service. Such an ICEBERG-PE BPM service can be attached as a task listener to a manual activity, which then is a realisation of the invocable CBR system setting, as described in Section 5.5.4. Finally, after defining further workflow variables and implementing workflow forms, the ICEBERG-PE instantiation can be executed.

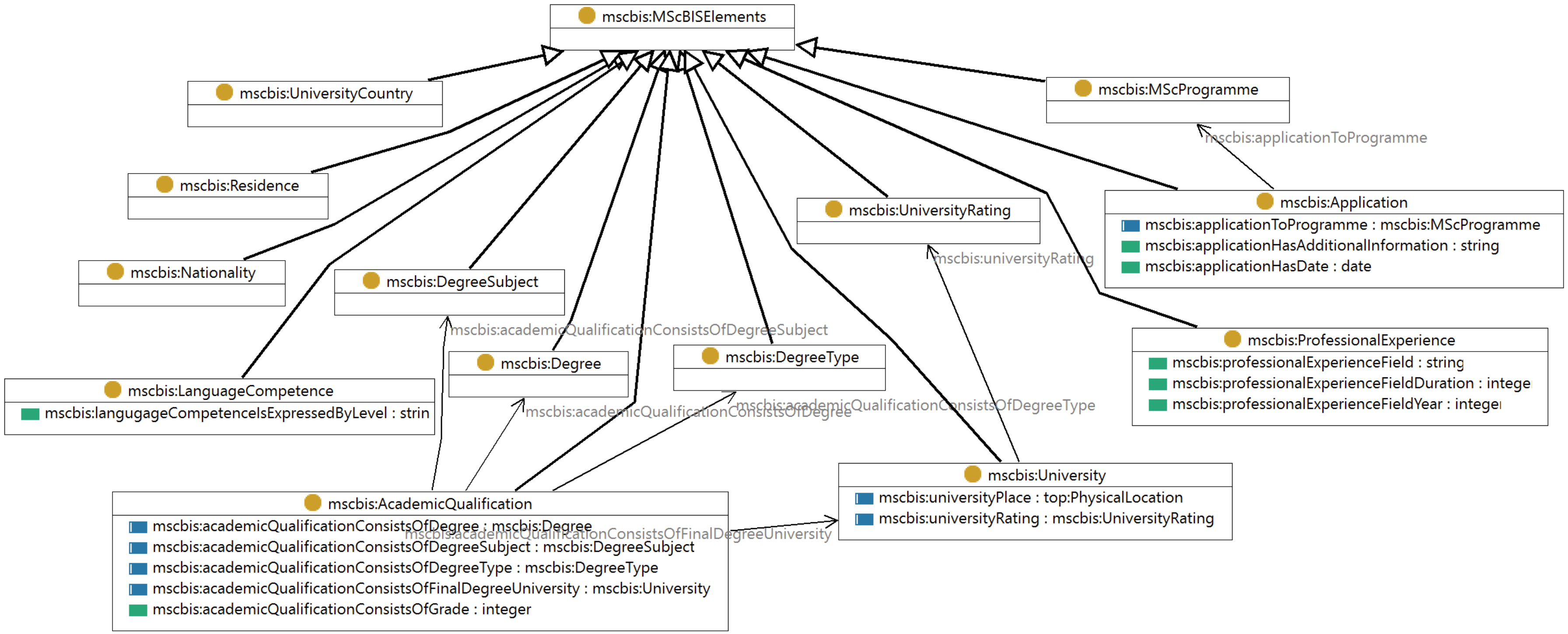

Figure 9.3: MSc BIS Ontology

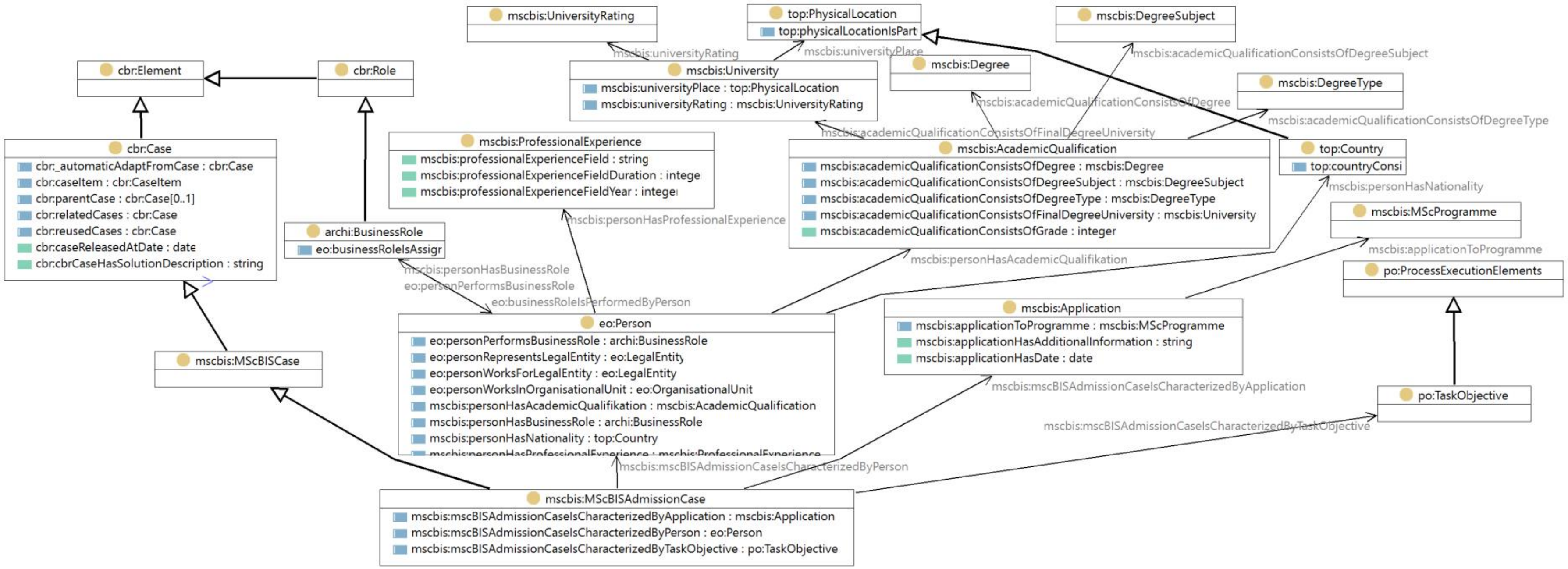

Figure 9.4: MSc BIS Case Model Ontology

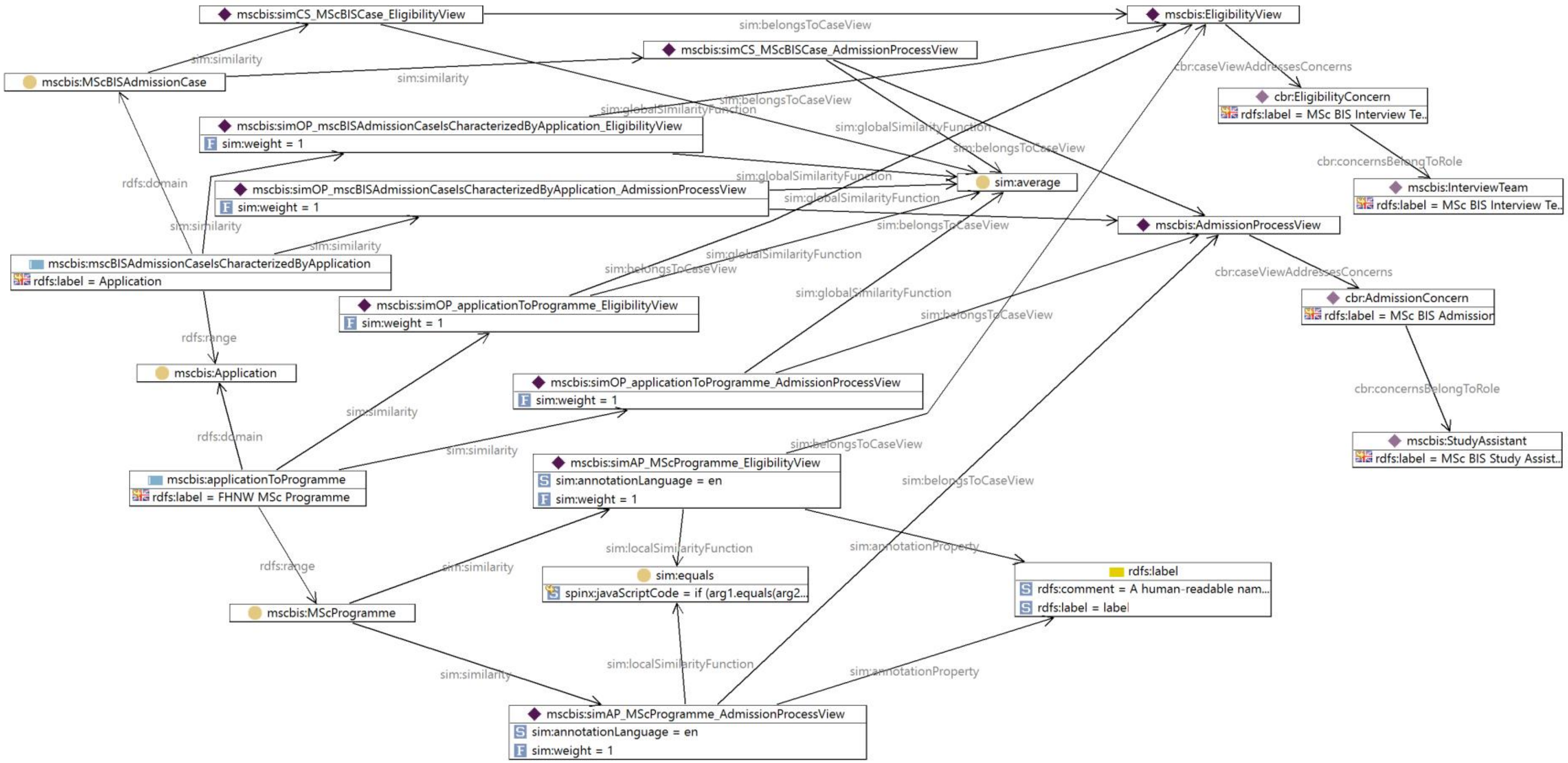

Figure 9.5: Exemplary Similarity Model Configuration

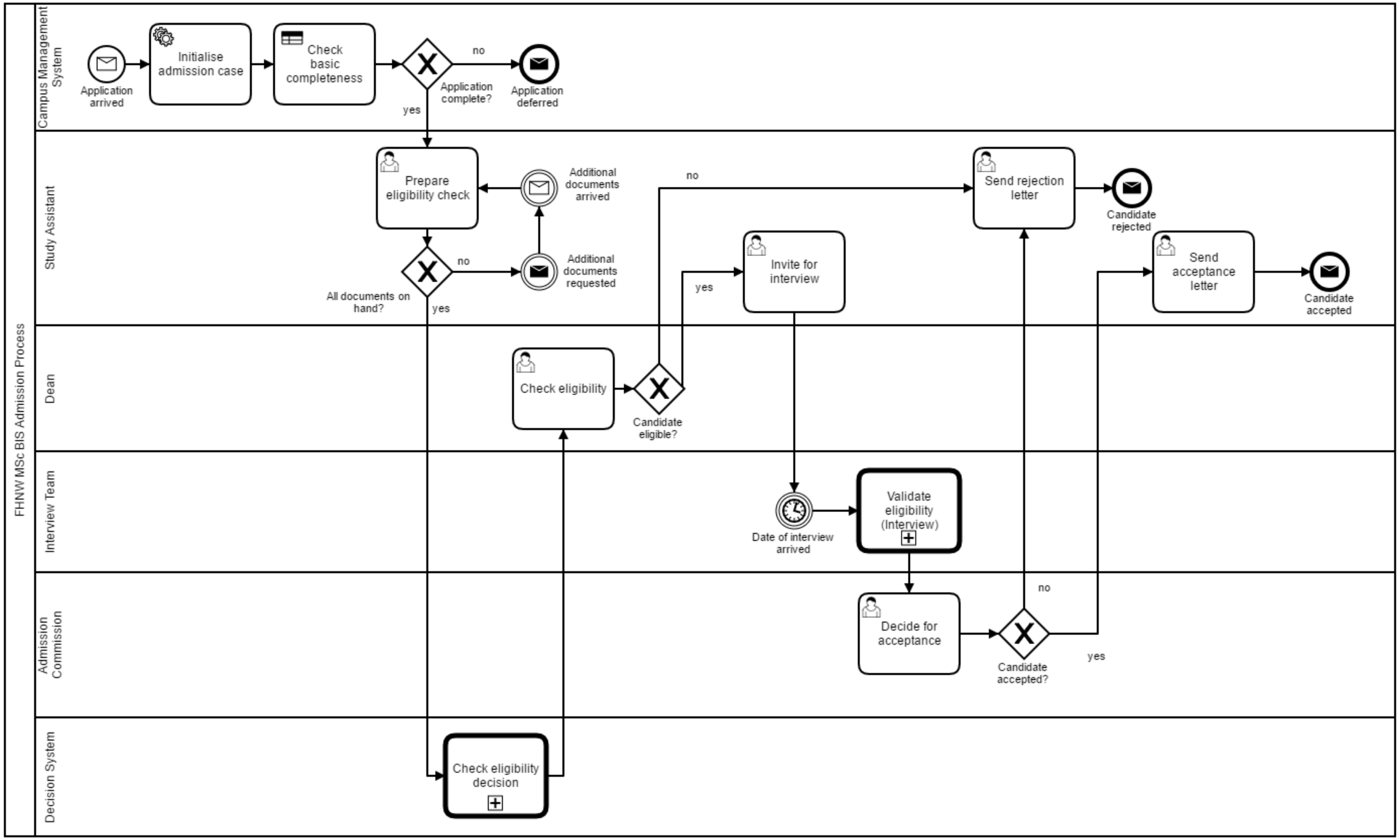

Figure 9.6: Admission Process implemented as BPMN Workflow

9.3 Summative Evaluation of Approach and Prototype

This section describes the summative evaluation of the ICEBERG-PE approach and prototype. This summative evaluation consists of two parts, which are a CBR process execution integration evaluation (Section 9.3.1) and similarity model evaluation (Section 9.3.2). Both parts use the admission process as described in the application scenario (Section 4.2) and the previous summative evaluation of the procedure model (Section 9.2).

9.3.1 Case-based Reasoning and Process Execution Integration Evaluation

This CBR process execution integration evaluation is an evaluation by demonstration and consists of a walk-through and execution of admission process implementation, as depicted in Figure 9.6. For this demonstration, an anonymized admission case (query case Q2; see Figure-Appx 6) is used, and the case repository consists of the cases shown in Appendix-B: Case Data as Graph Visualisation. Figure 9.7 shows the instantiation of the admission process by entering data from the applicant. The start of the process can be manually done, as depicted in Figure 9.7, or the process can be instantiated by receiving an email containing structured data from the web-based application system of the university. After the process has been instantiated, the CBR case will be initialized and created using the initial workflow data.

Figure 9.7: ICEBERG-PE - Start of the Admission Process

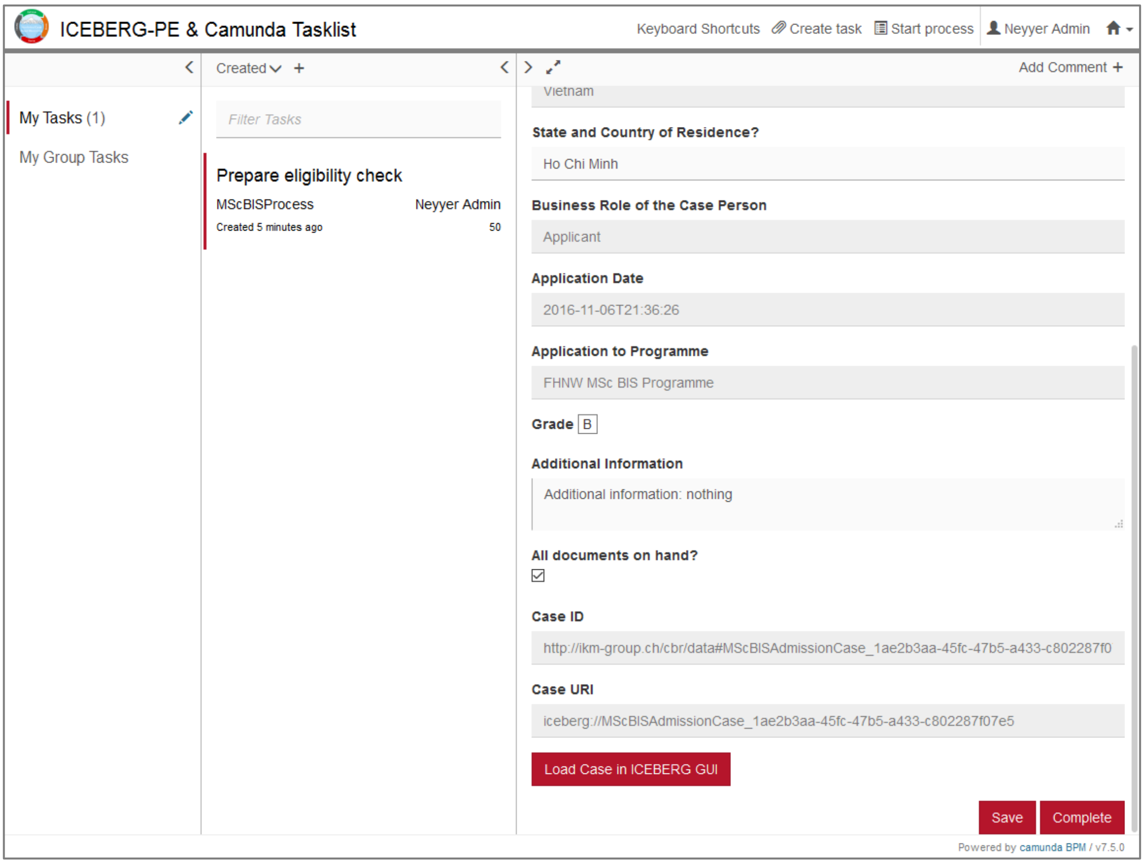

Figure 9.8: ICEBERG-PE - Prepare Eligibility Check by Study Assistant

Figure 9.8 shows how the case will be assigned as a task to the task list of the workflow system. The study assistant then sees all workflow relevant data to fulfil the current task of preparing the eligibility check. He or she then may want to consult the CBR system to retrieve similar cases from the past. However, the study assistant adds certain case data such as a certificate, transcripts of records or additional procedural knowledge to the current case. Since the CBR system is tightly linked to the workflow, a case is available just from the instantiation of the process, and the study assistant can load the current case containing the workflow data using the ICEBERG toolkit app. The current case is identified using a unique case ID and a URI, and can be seamlessly loaded in the CBR system.

Figure 9.9 shows how the current case can be loaded inside the ICEBERG Toolkit App. In the beginning, the current case just contains the basic workflow and case data. On the bottom of Figure 9.9, a graph-based representation of the current case characterisation is presented. On the upper half of the ICEBERG Toolkit App, the knowledge worker can describe the current case (describe the problem) by characterising the current case. This characterisation can be completed using a tree-like user interface, which has been generically generated based on the current viewpoint and concern of the study assistant. In other words, the similarity model as described in Section 9.2.2 defines the stakeholder specific characterisation elements here.

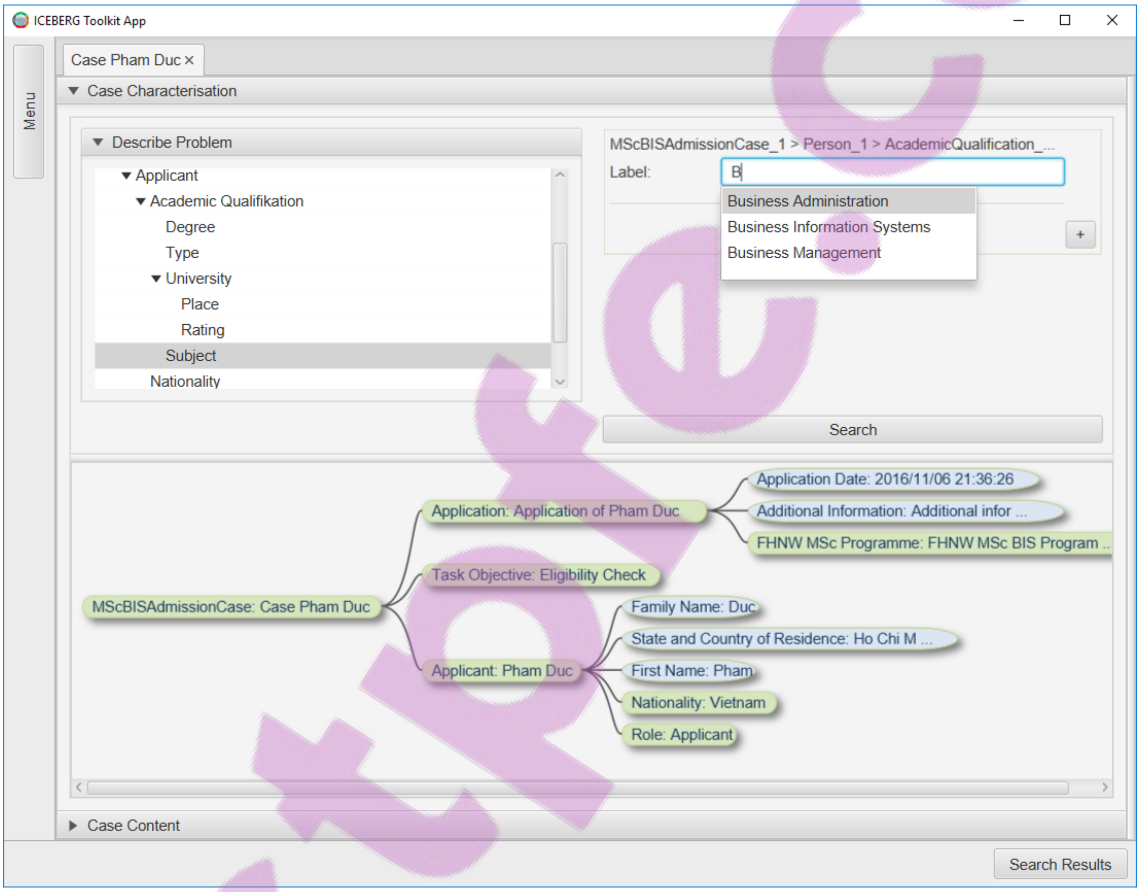

Figure 9.9: ICEBERG Toolkit App - Retrieve Similar Cases

The case characterisation and the content elements are entirely stored in a triple store using the ICEBERG and ArchiMEO ontologies as schema. The case data and potential characterisation values are always available by using value suggest or competition. At this stage, the knowledge worker can already add case content to the case.

Figure 9.10 shows the similarity computation after pressing the search button. In this scenario, the case repository contains five learned cases. The cases are ranked according to the computed similarity values. The similarities are calculated using the similarity configuration model, which is viewpoint and concern specific. The knowledge worker can preview the suggested cases using the show icon and then select the most relevant cases for adaptation. The knowledge worker is not restricted to select the top-ranked cases only; he or she is free to select any cases which might contain relevant knowledge items for adaptation.

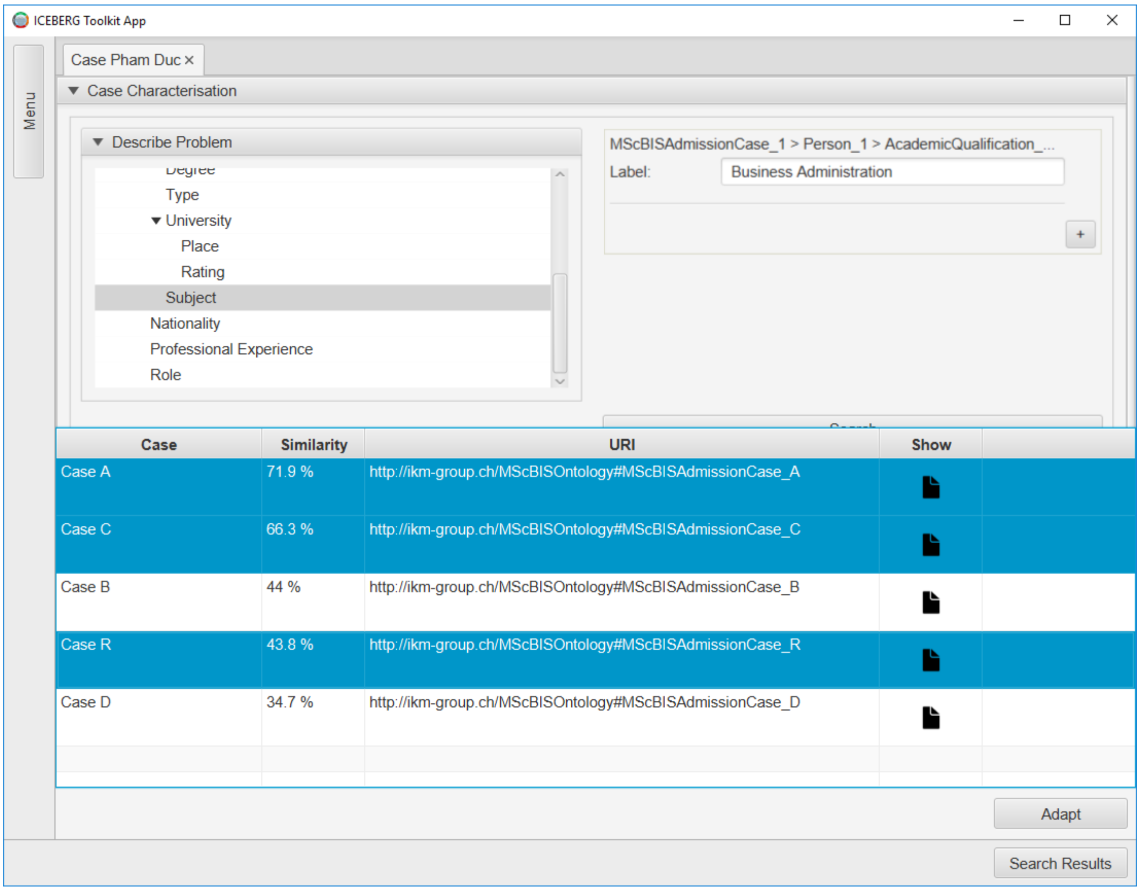

Figure 9.10: ICEBERG Toolkit App - Select Similar Cases

After selecting the relevant cases for adaptation, the knowledge worker can start the adaptation phase by pressing the adapt button. Figure 9.11 shows how the ICEBERG App Toolkit then presents the selected cases in the adaptation view. The knowledge worker is then exposed to a split screen view, which displays the current case on the left and the selected cases for adaptation on the right. Figure 9.11 shows the case characterisation of the current as well as the selected cases, including a graph-based representation and a tree-based editing possibility. At this stage, it is possible for the knowledge worker to run a retrieval again using the current case and even using a selected case as well. The adaptation itself can be done by switching to the case content since it is a principle of CBR that only content (the solution) can be adapted to another case (another problem). Alternatively, in other words, a learned solution might be a solution to a similar problem.

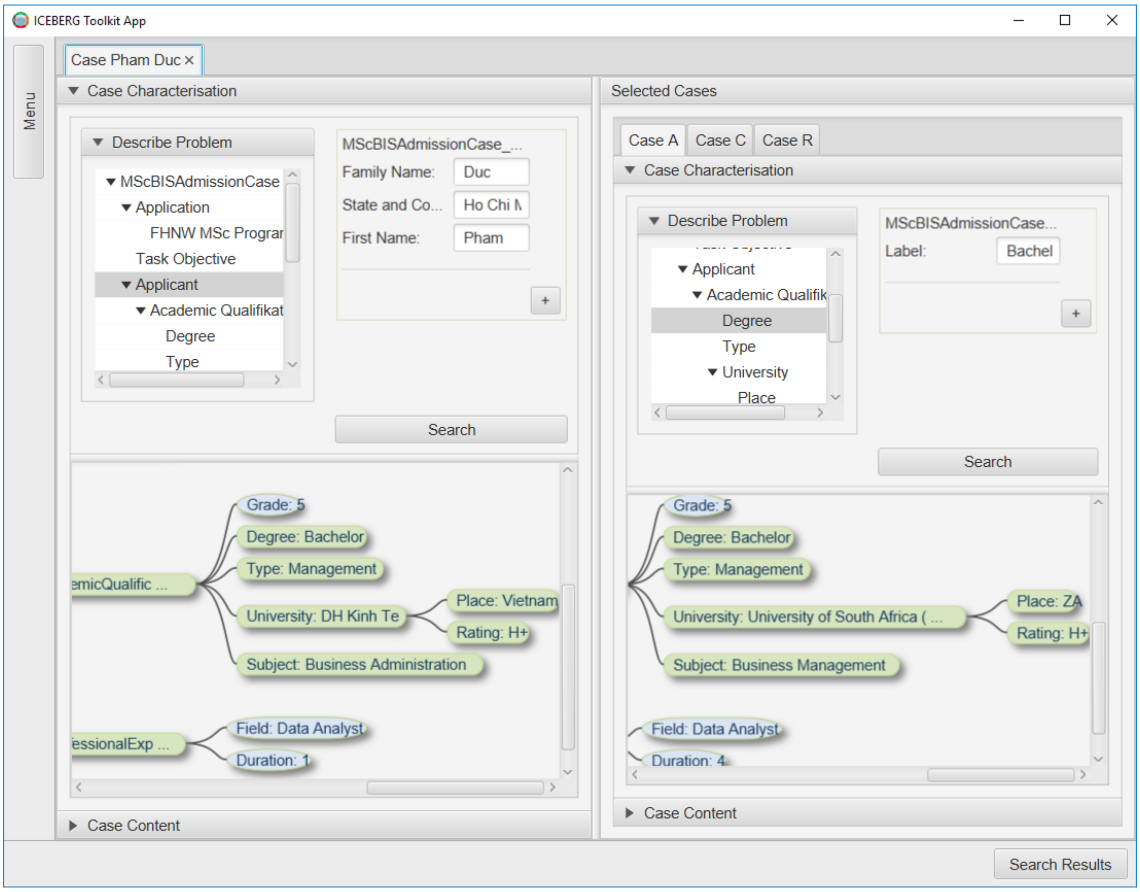

Figure 9.11: ICEBERG Toolkit App - Adapt Selected Cases

Figure 9.12 shows the case content view containing the content of the current case and the content of the selected cases. The case content view consists of structured elements such as documents, stakeholders and reused cases. On the left side, the content view provides an editor for embedding process fragments (as described in Section 6.4.2) and rich text. During adaptation, knowledge items can be moved from an old solution to the current case via drag-and-drop. Figure 9.12 shows how the adaptation has been supported by adaptation rules, as described in

Section 7.3. In this case, as shown in Figure 9.12, the rule described in Listing 7.3 has been violated, and the defined solution has been executed.

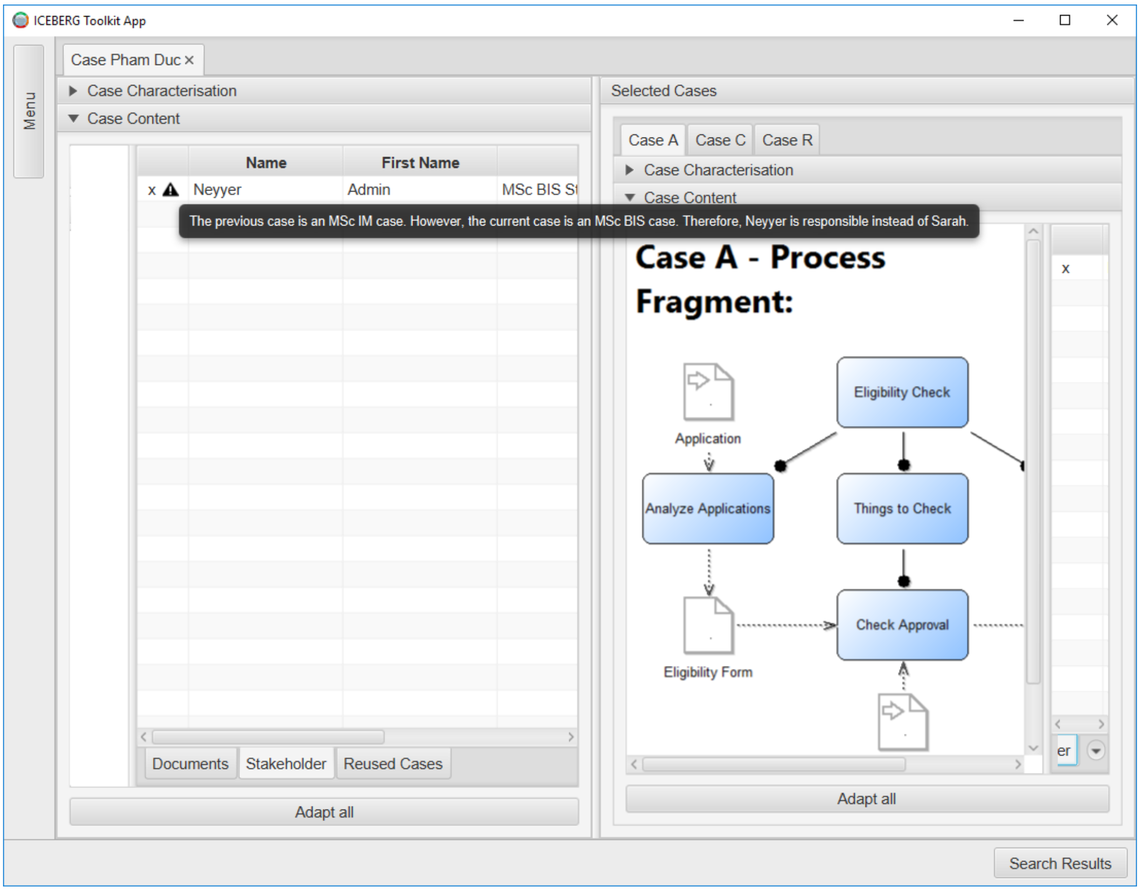

Figure 9.12: ICEBERG Toolkit App - Process Fragment Case Content and Adaptation Rule

If the case has been adapted and the knowledge worker has finished the editing, the knowledge worker then releases the case for revision, as shown in Figure 9.13. Later the knowledge worker can define the case as a learned case, which is then available for retrieval.

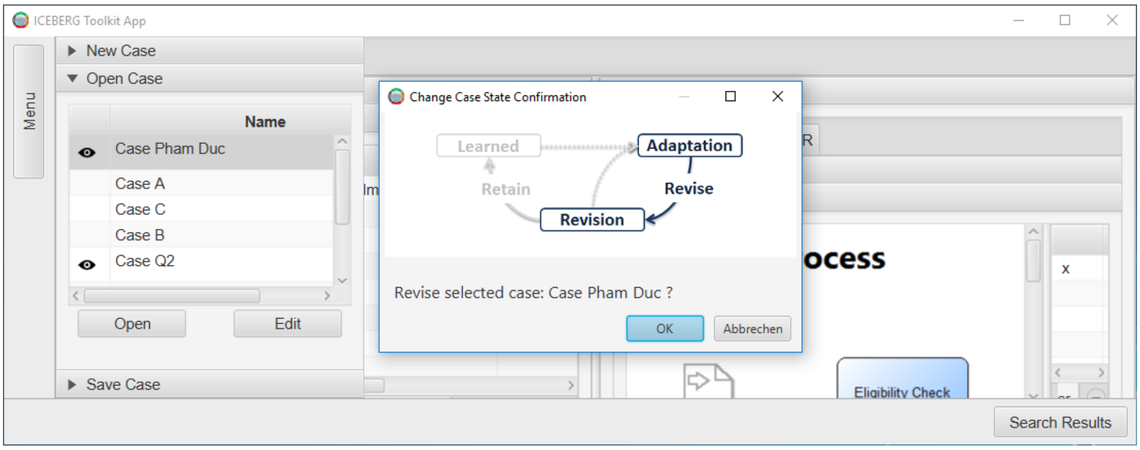

Figure 9.13: ICEBERG Toolkit App - Case State

The corresponding CBR cycle extension is described in Section 7.4. By the implementation of this CBR extension, it is possible for the knowledge worker to change an already learned case of required by changing the case state to revision. With this cycle extension, it is possible to realise the creation of generalised cases by the knowledge workers, as described in Section 6.4.2.4.

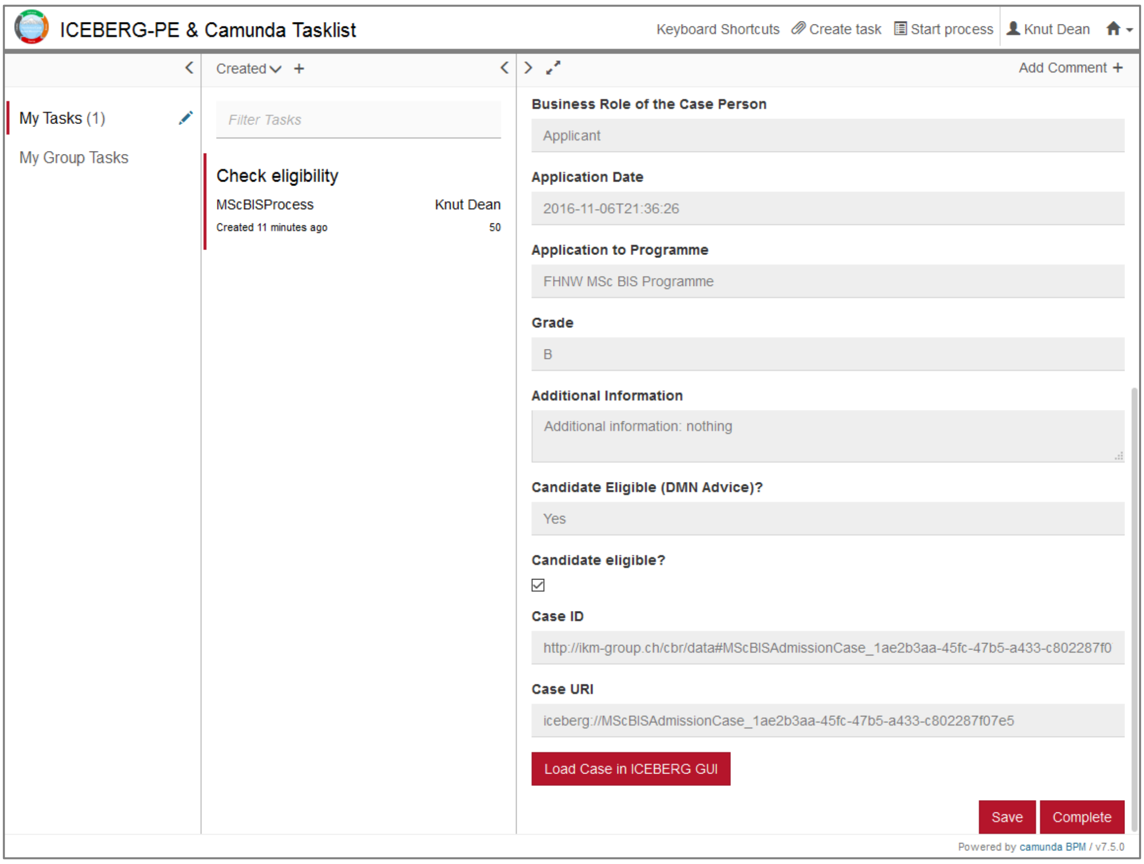

Figure 9.14: ICEBERG-PE - Check Eligibility by the Dean

If the study assistant has completed the check completeness activity, further workflow actors can continue working on the current case in a different activity. As shown in Figure 9.14, the Dean has to verify the eligibility check and has the same available case as the study assistant. The Dean can load the current case in the ICEBERG Toolkit App and eventually run a retrieval for finding similar cases. Nevertheless, the Dean now has a different view and concern in this different task, which may result in a different similarity result.

This demonstrative evaluation shows that case-based reasoning can be integrated with process execution. The workflow context and the case data are seamlessly integrated and available as cases. Furthermore, this demonstrative evaluation shows that the requirements, which have been derived from the research question and the application scenario, have been fulfilled. The next section shows the performance of the CBR retrieval with anonymized cases, which have been derived from real cases.

9.3.2 Similarity Model Evaluation

This section describes an evaluation of the similarity performance of the ICEBERG-PE approach. This evaluation has been done with two stakeholders of the admission process. Expert A and B are interviewers with the interview team performing the "validate eligibility" interviews, as depicted in Figure 9.6.

The experts have been exposed to two evaluation tasks (one and two) on two evaluation workshop dates: 06th September 2016 (expert A) and 08th September 2016 (expert B). The evaluation workshops have been conducted at the University of Applied Sciences and Arts Northwestern Switzerland FHNW and comply with the ethical clearance restrictions (see Section 3.4.6).

Procedure: The procedure of the evaluation was as follows. First, the experts were asked to create a similarity configuration. Then they had to rank the cases based on two query cases in task 1. Finally, they had to rank randomly selected cases according to two randomly selected query cases in task 2.

Setting: The experts are interviewers with the interview team performing the "validate eligibility" sub-task as depicted in Figure 9.6 and Figure 9.15. Figure 9.15 is a CMMN-based representation of the mentioned sub-task and contains two tasks. The validate eligibility task is the interview itself, where a candidate will be interviewed to verify if he or she is eligible for studying in the MSc BIS programme. Before, after and even during the interview, the interviewers can consult the ICEBERG-PE case base for retrieving similar cases to come to a more precise and well-balanced decision concerning the eligibility of a candidate by consulting previous decisions and cases.

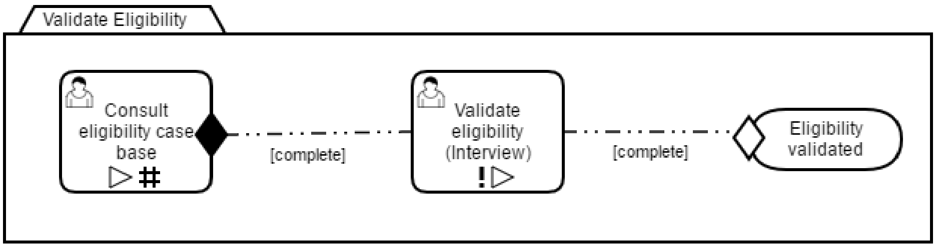

Figure 9.15: Validate Eligibility Sub-Task depicted as CMMN Model

Similarity configuration: During the preparation phase of the workshop, the experts were asked to create a similarity configuration by assigning values to an empty case characterisation graph. The experts could choose the similarity values (weights) freely (ad libitum), although both were using values between 0 and 1. The assignment of the weights was done without considering the annotation types, as described in Section 7.2.2. Additionally, no similarity function was defined by the experts. Both similarity configuration models, as depicted in Figure-Appx 10 and Figure-Appx 11, were transferred to an ICEBERG similarity model (as shown in Figure 9.5) configuration by an ontology engineer.

- ***Findings of similarity configuration***: Based on the similarity configuration evaluation it was possible to identify the following two findings:
    1. The global and local assignment of weights were appropriate. Nevertheless, it was not possible to differentiate between the annotation types as described in Section 7.2.2. Additionally, the

similarity functions have to be defined by an ontology engineer or a case-based reasoning expert.

2. The similarity configuration showed that it is not always possible to create an entirely agreed-upon similarity configuration which is shared among all stakeholders. One possibility would be to create a reasonable compromise as the similarity configuration. Alternatively, it may require that separated viewpoints be implemented, leading to significant effort. It would be worthwhile to investigate in future work if the knowledge worker could adjust a shared similarity configuration prior to a retrieval.

Task 1: In task one, the experts were asked to rank four cases, visualised as characterisation graphs (see Appendix-B: Case Data as Graph Visualisation), according to the similarity of two query cases. The experts were asked to assign numbers between 1 (highest similarity) and 4 (lowest similarity) to the cases. Later the ICEBERG-PE prototypes were used to run the retrieval with the same cases in the repository and the corresponding query cases. The results are represented as similarity percentages.

Table 9.1: Task 1 Results of Expert A

*Evaluation **Task 1** with Similarity Model of **Expert A***

	Query Case Q1		**Query Case Q2**	
Repository	Expectation	Result	Expectation	Result
Case A	4	41%	1	67%
Case B	1	55%	3	44%
Case C	3	40%	2	60%
Case D	1	55%	4	35%

Table 9.1 shows the results of expert A concerning task 1. The expected ranking of query case 1 was perfectly matched for top-ranked cases, and the lower-ranked cases showed an accurate result. The ranking of query case 2 perfectly matched for the whole repository, as expected by the expert.

Table 9.2: Task 1 Results of Expert B

Evaluation **Task 1** with Similarity Model of **Expert B**	Query Case Q1		Query Case Q2	
Repository	Expectation	Result	Expectation	Result
Case A	3	44%	1	72%
Case B	2	61%	3	44%
Case C	4	53%	2	66%
Case D	1	61%	4	35%

Table 9.2 shows the results of expert B concerning task 1. The expected ranking of query case 1 were almost perfect, since there is a separation between the top-ranked and the lower-ranked cases, although the lower-ranked cases have a slightly different ranking expectation. Similar to the results of expert A, the resulted ranking concerning query case 2 with the similarity model of expert B was also precise.

- *Findings of Task 1*: The results of task 1 were almost perfect, as expected. This expectation was stated by both experts during the workshop discussions. They reported that the comparison of four cases is manageable but nearly unmanageable. For comparing more cases, a system such as ICEBERG would improve the retrieval significantly, according to the experts.

Task 2: In task two, the experts were exposed to a more challenging task. They had to randomly select 6 repository cases and 2 query cases each out of 66 anonymized cases. Comparable to task 1, the experts were then asked to assign numbers (expectation) between 1 (highest similarity) and 6 (lowest similarity) to the cases. Later the ICEBERG-PE prototypes were used to run the retrieval with the same cases in the repository and the corresponding query cases. The results are represented as similarity percentages.

Table 9.3: Task 2 Results of Expert A

Evaluation ***Task 2*** *with Similarity Model of* ***Expert A***

	Query Case 42		Query Case 17	
Repository	Expectation	Result	Expectation	Result
Case 9	5	63%	2	54%
Case 24	6	54%	1	73%
Case 35	1	72%	3	62%
Case 39	2	74%	5	57%
Case 56	3	65%	6	53%
Case 65	4	64%	4	60%

Table 9.3 shows the results of expert A concerning task 2. The results of query case 42 were almost accurate, except for the slight difference of the expectation of the two top-ranked cases. Comparable were the results of query case 17, where the results were again almost accurate except for the outlier of case 9.

Table 9.4: Task 2 Results of Expert B

Evaluation ***Task 2*** *with Similarity Model of* ***Expert B***

	Query Case 59		Query Case 64	
Repository	Expectation	Result	Expectation	Result
Case 12	5	53%	6	52%
Case 54	2	43%	3	54%
Case 7	3	45%	2	49%
Case 32	1	47%	1	65%
Case 15	6	55%	4	62%
Case 41	4	44%	5	50%

Table 9.4 shows the task 2 results of expert B. The expectation and the resulting similarity values of query case 64 were satisfactory, although case 7 was ranked with the lowest similarity. Unfortunately, the result of query case 59 was not as expected, except for cases 7 and 41.

- *Findings of task* 2: In task 2 the results of expert A were almost accurate, and one result of expert B was almost satisfactory. Nevertheless, a final discussion with the experts revealed some difficulties and suggestions for future research. Both experts stated that it is extremely demanding and difficult to compare more than four cases. Neither expert excluded the possibility of errors due to a large number of repository cases. Although the results may not have been perfect due to the number of repository cases, the experts attested and considered the ICEBERG-PE approach to be a significant facilitation of the admission process and the corresponding knowledge work. Expert B came up with a possible explanation of the lower results of query case 59. According to expert B, it would be worthwhile to investigate in future research the possibility of adjusting the similarity configuration model just before retrieval. Expert B realised that some aspects of the case characterisation should have been weighted differently, specifically for query case 59. In other words, for specific and unusual cases, it would make sense to weight certain characterisation elements differently. To be able to adjust the weights, the experts suggested that the similarity configuration should be visible inside ICEBERG.

In general, the experts attested and considered the ICEBERG-PE approach to be a significant facilitation of the admission process. The approach supports the knowledge work of the admission process significantly.

9.4 Confirmatory Evaluation

In this section, the confirmatory evaluation is described, which provides additional evidence and confirmation of the validity of the ICEBERG approach itself and confirms that the ICEBERG approach generalises and is transferable to other contexts. These contexts are sales and project management in the private sector of the economy, as well as business processes in publication administrations. The confirmatory evaluation results come from the [sic!] research project and the EU research project Learn PAd.

9.4.1 Results from the [sic!] Research Project[33]

This section describes an evaluation that was conducted using two application scenarios derived from the analysis of the business of ELO Digital Office CH AG, the business partner of the applied [sic!] research project. This evaluation provides additional evidence and confirmation of the validity of the ICEBERG approach.

The ICEBERG approach was evaluated based on these scenarios, which verified the utility with a prototype (instantiation):

1. Offering: During the sales and offer phases, ELO experts have to analyse a large set of requirements and answer questionnaires provided by their customers. The offer is made based on this analysis.
2. Project management: Another service provided by ELO is the management of projects with the target of integrating its standard software in its customers' IT environment.

The two application scenarios have been used and implemented in practice to verify if a CBR system can provide knowledge for the given cases and tasks to be performed.

Following the ICEBERG procedure model, a case characterisation has been defined with annotations of the similarity ontology applied to the enterprise and domain ontology. The enterprise ontology consists of concepts of the business partner domain ontology extended concepts of ArchiMEO and the business motivation model:

1. Offering: During the offer phase, a technical consultant answers questions about requirements specified by potential customers. Since the technical consultant knows the ELO software modules, the questions about the

[33] Some verbatim passages presented in this following section, which have been achieved in this thesis project, are published in the following own authored publication: Martin et al. (2016)

integration of legacy systems are more critical for him/her. Therefore, the technical consultant is looking for experiences made with integrated systems in previous projects. These concerns lead to a specific viewpoint on previous cases and are reflected in a view of the case characterisation. Weights and similarity functions defined in this view are applied when a query concerning their characterisations which is based on this view is compared with the previous cases. The background knowledge of the defined scenarios and the stakeholder's concerns regarding the (legacy) systems are made explicit.

2. Project management: In the project management scenario, a project manager is staffing his project team. The manager knows that some adaptations have to be programmed for specific modules to fit the target environment. Therefore, he/she is looking to see if previous cases can be retrieved in which equal modules have been integrated and programmers have been assigned. This specific view is reflected in the additional view, which consists of the same case characterisation but with different weights (the similarity of modules is now of higher interest) than in the offering viewpoint.

Table 9.5 shows a shortened and exemplary similarity computation based on the above-mentioned project management viewpoint and scenario.In the [sic!] research project, the ICEBERG approach and procedure model has been applied. As a result, the overall experience management of the business partner could be enhanced with the applied ICEBERG approach. Through the inclusion of contextual information, based on the enterprise (ArchiMEO based), domain and application ontology:

1. the stakeholder could use the enterprise-specific vocabulary, which has been stated as positive from the stakeholder;
2. the CBR vocabulary acquisition effort has been decreased; and

3. the CBR configuration (similarity and adaptation models) was effortless and more precise.

Table 9.5: Exemplary Similarity Computation (Martin, Emmenegger and Wilke, 2013)

Class	Instance	Property	Weight	Function	Sim #1		Sim #2		Sim #3	
Case	"_queryCase"	hasSystem "_querySys"								
		hasModule "_queryMod"								
	"case1"	hasSystem "case1Sys"	1	average					0.0	**0.017**
		hasModule "case1Mod"	5	average					0.1	
	"case2"	hasSystem "case2Sys"	1	average					1.0	**0.417**
		hasModule "case2Mod"	5	average					0.3	
System	"_querySys"	name "MySQL"								
		version ""								
	"case1Sys	name "Oracle"	2	levenshtein	0.0	0.0				
		version "11g"	1	version						
	"case2Sys	name "MySQL"	2	levenshtein	1.0	1.0				
		version "5.1"	1	version						
Module	"_queryMod"	label "Backup"								
		hasExpert "_queryExp"								
	"case1Mod"	label "Barcode"	3	equals			0.0	0.1		
		hasExpert "case1Exp"	2	average			0.25			
	"case2Mod"	label "Backup"	3	equals			1.0	0.3		
		hasExpert "case2MExp"	2	average			0.75			
Employee	"_queryExp"	role "Programmer"								
		level "Expert"								
	"case1Exp"	role "TechConsultant"	3	levenshtein	0.0	0.25				
		level "Expert"	1	equals	1.0					
	"case2Exp"	role "Programmer"	3	levenshtein	1.0	0.75				
		level "Beginner"	1	equals	0.0					

Overall the [sic!] research project showed the validity and usefulness of the ICEBERG approach. Further information can be accessed in the own-own authored publications of Martin, Emmenegger and Wilke (2013) and Martin et al. (2016).

9.4.2 Results of the Learn PAd Research Project[34]

This section describes an evaluation that was conducted using an application scenario derived from public administrations, which were project partners of the EU research project Learn PAd. The objective of Learn PAd was to provide learning solutions for the public administrations focussing on their business processes (BPs) and context.

[34] Some verbatim passages presented in this section have already been published in the co-authored publications (Thönssen et al., 2016; Emmenegger et al., 2017).

In the Learn PAd project, the ICEBERG approach and method was applied in a process execution context, where the cases correspond to instances of (parts of) business processes. The goal is to support civil servants in the reuse of experience from past process instances. This goal is achieved using the characteristics of the process instances, which are relevant for finding learned cases. The case characterisation and content model have been developed for the "Titolo Unico" process. This business process is executed when a public administration provides permissions to citizens' activity requests (e.g. starting a business, restructuring or extending a commercial location) and is specified for public administrations of the Marche, Italy region. The case characterisation is based on Learn PAd specific metamodels and ontologies, including ArchiMEO. The ICEBERG approach is used to assess the similarity between a new case characterisation and learned ones, and at the end of the day, to retrieve learned cases from which civil servants can re-use experience.

The evaluation was done with two representatives of the Marche, Italy region and on the achieved quality of recommended cases, as explained next. Starting with 12 difficult former cases which had been selected, an expert extracted certain aspects to create a fictitious new query case. Then the expert identified the three most similar cases to the query case and determined their ranking. Next, the expert compared and assessed their own selection with the suggestion of the ICEBERG system according to ranking and relevancy (higher ranked cases, which are not part of their own selection, may be more relevant than expected).

Table 9.6 shows the retrieval results from two runs. In the first run, the initial case characterisation configuration was used. The result of the second run was achieved after the weights were optimised to achieve a better rank for the cases identified as relevant by the expert (bold case title in Table 9.6).

Table 9.6: Results of Two Retrieval Runs with the Learn PAd CBR System (Thönssen et al., 2016)

Rank	Run 1	Similarity	Run 2	Similarity
1	655.2015 Realization – Installation radioelectric antenna for WiFi data transmission in protected area	.046	**829.2015: Restructuring of a chalet and adjustment of the beach area**	0.44
2	**829.2015: Restructuring of a chalet and adjustment of the beach area**	0.42	**1118.2015: Realization of masonry walls on hotel business – Senigallia.**	0.36
3	195.2015 Realization of a petrol station and a building crafts	0.40	**431.2014: Restructuring of a civil building to allocate as a B&B**	0.36
4	**1118.2015: Realization of masonry walls on hotel business – Senigallia.**	0.40	515.2015 Restructuring of a chalet for the realization of an internal bar	0.34
5	**431.2014: Restructuring of a civil building to allocate as a B&B**	0.39	655.2015 Realization – Installation radioelectric antenna for WiFi data transmission in protected area	0.33
6	515.2015 Restructuring of a chalet for the realization of an internal bar	0.33	195.2015 Realization of a petrol station and a building crafts	0.31
7	22294.2013 Realization of a petrol station	0.30	1267.2015 Transformation – Replacing of windows fixtures in a hotel business activity	0.26
8	64682.2014 Realization of a recovery/waste disposal plant	0.30	889.2015 Expansion – Installation of removable covers for outdoor dining in a restaurant business activity	0.26

These results show that the ICEBERG approach works well with an initial case characterisation. Nevertheless, this result can be enhanced with further examples of potential cases with the corresponding relevancy suggested in the ICEBERG procedure model (step 4 case content; see Section 5.6.2).

Overall the results from the Learn PAd case show the validity and usefulness of the ICEBERG approach. Further information can be accessed in the co-authored publication of Emmenegger et al. (2017) and research project deliverable of Thönssen et al. (2016).

9.5 Conclusion

This chapter evaluates the artefacts of this theses, both those implemented in the prototype and those described as the procedure model. The evaluation of the procedure model verified the usefulness and applicability of the procedure model.

The summative evaluation of the approach and demonstrator showed that case-based reasoning could be integrated with process execution. The workflow context and the case data are seamlessly integrated and available as cases. Furthermore, this demonstrative evaluation showed that the requirements, which have been derived from the research question and the application scenario, had been fulfilled.

The similarity model evaluation, together with end users, showed almost perfect results, provided profound findings and uncovered potential elements for future research.

Finally, the confirmatory evaluation is described, which provides additional evidence and confirmation of the validity of the ICEBERG approach itself and confirms that the ICEBERG approach generalises and is transferable to other contexts.

10 Conclusion

It is the nature of research that researchers (dwarfs) are standing on the shoulders of giants. The metaphor of dwarfs standing on the shoulders of giants is attributed to Bernhard von Chartres in 1120 (of Salisbury, 2009; Merton, 1965). This metaphor reflects the idea of "[…] discovering truth by building on previous discoveries" (Kling, Manrodt, Vitasek and Keith, 2015, p.1).

This research builds on the shoulders of giants since it depends on the results of previous research and combines different research directions of information systems research to introduce a new approach. This research combines the research directions of case-based reasoning, ontologies and enterprise ontologies, enterprise architecture, business process management and execution, and case management. This thesis uses this underlying principle of combining existing work and new elements, which has led to the following new approaches, models and instantiations:

- A combined case-based reasoning and process execution approach.
- An ontology-based case-based reasoning approach.
- A new CBR configuration using a configuration ontology.
- A new procedure model for implementing a new instantiation of the ICEBERG-PE approach in practice.
- A case content model for procedural knowledge and classical case content.
- A case characterisation model for describing cases using an enterprise ontology and process execution context.

- A prototypical implementation and initiation of the new approach.

In the following section, the contributions to the body of knowledge and practice are listed, the results are discussed based on the research questions and the potential for future work is described.

10.1 Contribution

The main outcome (artefact) of this research work is an approach for knowledge-intensive work which combines case-based reasoning and process execution. The approach is divided into sub-artefacts, which are guided by the research questions. These sub-artefacts are an ontology, a procure model, a case model and case-based reasoning services. The approach is implemented in a prototype system, which has been evaluated using real-world data.

A scientific contribution is a new approach supporting the execution of knowledge-intensive business processes by adopting the case-based reasoning. The approach has been implemented as a reusable open-source prototype for running experiments and has been tested for a real-world application scenario including test data.

This research work considers the real-world context just from the beginning. This real-world context focus ensures that the results contribute to the business practice. The results have been transferred to similar application areas as presented in the evaluation section.

10.1.1 Artefact Contribution

As mentioned before, it is a basic principle of design science research to produce artefacts, which then can be used to gain knowledge and understanding about a certain research problem. “The fundamental principle of design science research is that knowledge and understanding of a design problem and its solution are acquired in the building and application of an artefact” (Hevner and Chatterjee, 2010, p.5). Table 10.1 lists the artefacts of this thesis and points out the acquired knowledge and understanding while building, describing and implementing them.

Table 10.1: Artefact Contribution and Understanding

Artefact	*Description*	*Understanding*	*Reference*
Approach	The conceptual framework of the approach shows conceptually how case-based reasoning can be integrated with process execution, which allows learning from the execution of non-structured process parts.	• Conceptual understanding of the integration, the sub-artefacts and the corresponding elements.	Chapter 5
Ontology	The ontology framework provides a conceptual answer on how domain knowledge and contextual information can be used for the retrieval of cases and suggestion or adaptation of case items. The case viewpoint model as part of the ontology framework delineates how the process execution context can be integrated into the case description.	• General conceptualisation and structuring • Enterprise knowledge integration • Integration of different viewpoints and concerns • Integration of domain and contextual knowledge	Section 5.5.1 and 9.2.1
Procedure Model	The procedure model (methodology) describes how the combination of CBR and process execution can be implemented in a new application scenario in practice.	• Guideline and reference for a project realisation • Requirements elicitation • Domain model construction	Section 5.6

Case Model	The case model consists of a description of the case characterisation and the case content. The case model integrates the process execution context. The case content is extended with models of process fragments, which represent procedural knowledge.	• Case complexity • Cognitive adequacy • Case characterisation describing process knowledge • Process knowledge as content • Case content modelling	Chapter 6
CBR Services	The CBR services are used to run the approach and are implemented in the prototype. The CBR services contribute with a new similarity model configuration including the viewpoint model, a semantic rule-based adaptation and a new case state model.	• Similarity configuration • Case similarity and adaptation • Case evaluation and learning	Chapter 7
Prototype	The prototype embeds all the required components (artefacts) to run the case-based reasoning and process execution approach as a software application.	• Process execution and CBR integration • CBR architecture • CBR implementation	Chapter 8

The artefacts themselves and their construction is a contribution to the involved research projects and the application scenario of the thesis. However, it is a principle of a design science research projects that the artefacts and the acquired knowledge and insights will be contributed back to the body of knowledge and to practice, as described in the following two sections.

10.1.2 Contribution to Practice

As mentioned in the previous section, it is a principle of a design science research projects that the artefacts, acquired knowledge and understanding are communicated back to practice. In this thesis, the communication process took place as part of the project work and application scenario involvement.

- *Contribution to entrepreneurs*: From the beginning, it was possible to communicate initial results of the ICEBERG approach to a large number of entrepreneurs through a newspaper article (Martin, 2013). The article appeared in a Swiss newspaper called "UnternehmerZeitung" (UZ), whose translated name is "Entrepreneur's Newspaper". The translated title of the article is "Knowledge work is not routine work".
- *Contribution to the thesis project partner*: The artefacts have been developed based on the admission process scenario of the Master of Science (MSc) programmes at the FHNW University of Applied Sciences and Arts Northwestern Switzerland. The university stakeholders could use the provided prototype in practice.
- *Contribution to an applied research project partner*: As mentioned in Section 3.4.2.1 and 9.4.1, the ICEBERG approach and procedure model has been applied in the research project [sic!], in which a case-based reasoning system for the offer process and project management of a software company has been developed. As a result, the overall experience management of the business partner (ELO Digital Office AG) could be enhanced. Overall the [sic!] research project showed the validity and usefulness of the ICEBERG approach. The artefacts of this thesis have been transferred to the [sic!] application partner and are available to the company.

10.1.3 Contribution to the Body of Knowledge

The contribution to the body of knowledge is communicated within a research community primarily through journal articles and conference publications.

- Early results of the underlying approach of this thesis, the ICEBERG approach (see Chapter 5), were presented at the IEEE Enterprise Systems conference (Martin, Emmenegger and Wilke, 2013).
- Later a new retrieval function was elaborated specifically for the ICEBERG toolkit (see Section 7.2.2), which was presented at the International Workshop on Case-Based Reasoning CBR MD in Hamburg (Witschel et al., 2015).
- After several development iterations of the case model, a paper concerning the case modelling language as suggested in Sections 6.4.2.3 and 6.4.2.4 was published at the AdaptiveCM 2015, the 4th International Workshop on Adaptive Case Management and other non-workflow approaches (Cognini, Hinkelmann and Martin, 2016).
- In 2016 it was possible to publish the advanced and matured approach (see Chapter 5) with a focus on the enterprise ontology, the procedure model (see Section 5.6.1) and the viewpoint-model (see Section 5.4) in the Enterprise Information Systems Journal (Martin et al., 2016)
- Finally, a book section has been accepted for publication, which is currently in press, describing the implementation of the ICEBERG-PE approach (see Chapter 5), as well as selected components of the ICEBERG-PE prototype (see Chapter 8) and their application for workplace learning (Emmenegger et al., 2017 forthcoming).

10.2 Summary and Research Questions

This section summarises the results of this research work by providing the answers to the research questions, including the references to the relevant chapters and section in the body of this thesis.

10.2.1 Summary: Ontology-based CBR and Process Execution Approach and Prototype (Research Question 1)

Research question 1 guided the research concerning the overall ontology-based CBR and process execution approach.

RQ 1: How can case-based reasoning be integrated with process execution?

The research question is answered by introducing the new ICEBERG-PE approach and a methodology for implementing it. The ICEBERG-PE approach has been derived from existing work, literature and the admission process application scenario. To ensure reproducibility, the instantiation and the underlying concept of the approach have been iteratively developed in additional application scenarios as presented in Section 9.4.

A conceptual answer to research question 1 is provided in Chapter 5. This conceptual view consists of four elements:

- A case-repository is a central feature of the ICEBERG-PE approach; it contains retained and learned cases.
- The case-based reasoning services are providing automatic retrieval, semi-automatic re-use and adaptation of the previous cases, manual revision and automatic retention of cases.
- An ontology is used for the CBR configuration, storage of the enterprise and domain ontology.
- The process execution element is an instantiation of a workflow engine running a workflow definition.

Section 5.5.4 provides an answer to how the integration of case-based reasoning and process execution can be made. The answer reflects two possible settings: first a stand-alone CBR system and second an invocable CBR system. The ICEBERG-PE approach focuses on invocation since it incorporates with the workflow engine and pre-populates the case characterisation using workflow- and case-relevant data.

Section 5.6 introduces a methodology consisting of an ICEBERG-PE procedure model, which answers the research question by describing how a combined approach can be implemented in a new application scenario in practice.

The implementation of the ICEBERG-PE approach as presented in Chapter 8 answers research question 1 from an implementation perspective. Based on constraints for selecting the environmental software components, a framework for the prototype could be set up. The ICEBERG-PE architecture, as shown in Section 8.2, embeds all the required components and finally answers how the integration of CBR and process execution can be realised technically.

10.2.2 Summary: Case Description for Knowledge-intensive Work (Research Question 2)

Research question 2 consists of two further sub-research questions which guide the investigation of a case description of knowledge work:

RQ 2: What should the case description for knowledge-intensive work consist of?

RQ 2.1: How can functional and process knowledge be included in a case description that is cognitively adequate to humans?

RQ 2.2: How can the process execution context be integrated into the case description?

Research question 2.1 is answered first by an investigation about complexity and cognitive adequacy. The investigation of complexity and cognitive adequacy is described in Section 6.2. Based on the complexity investigation, potential case content modelling languages are presented in Section 6.4.2.2.

Research question 2.2 is answered by an investigation about context itself and specifically for process execution context in Sections 6.1.1 and 6.1.2. Section 6.1.3 then gives an answer to sub-research question 2.2 by describing the integration of process execution context into the ontology framework of the ICEBERG-PE approach. This research into the process execution context laid the basis for the case characterisation. The processes execution context, which is enterprise context, can be integrated using information derived from an enterprise ontology and with the usage of a vocabulary that is linked to the same enterprise ontology.

Finally, research question 2 could be answered based on the objectives for a case model (see Section 6.4.1), from which a case content model containing process knowledge is introduced in Section 6.4.2. The case content model consists of elements from the ICEBERG-PE ontology structure and a potential case content modelling language, BPFM, with an acceptable level of complexity (see Section 6.4.2.3).

10.2.3 Summary: Case-based Reasoning Services (Research Question 3)

Research question 3 guides the investigation into how process knowledge can be retrieved, adapted and learned, which has been made available in cases.

RQ 3: How can case-based reasoning services support process execution?

- *RQ 3.1: How can the similarity between cases for knowledge work be calculated?*
- *RQ 3.2: How can domain knowledge and contextual information be used for retrieval of cases and suggestion or adaptation of case items?*

Chapter 7 introduces the CBR services, which are used to run the ICEBERG-PE approach and ultimately answer research question 3.

Section 7.2 provides the answer to sub-question 3.1 about how the similarity of knowledge work can be calculated. The introduced similarity mechanism uses the case characterisation, which characterises the knowledge work description.

Section 7.3 provides the answer to how domain knowledge and contextual information can be used for retrieval and adaptation to answer sub-research questions 3.2. Through the inclusion of enterprise ontology, the ontology-based similarity model and the semantic adaptation rules, domain knowledge and contextual information can be seamlessly integrated.

The implementation of the ICEBERG-PE approach as presented in Chapter 8 then answers research question 3 from an implementation perspective. The ICEBERG-PE architecture, as shown in Section 8.2, embeds all the required components and finally answers how case-based reasoning services can support process execution technically.

10.3 Methodological Reflection

This section reflects the methodological decision in this research and assesses the appropriateness using the design research guidelines as presented by Hevner et al. (2004, p.83). These guidelines are used to evaluate this thesis to determine if it is in adherence to design-science research. The following Table 10.2 shows a design research guideline-based methodological reflection and the corresponding references to chapters or sections of this thesis:

Table 10.2: Design Research Guideline-based Reflection

Guideline	*Description*	*Reflection*	*Reference*
1. Design as an Artefact	Design-science research must produce a viable artefact in the form of a construct, a	The research produced the following artefacts: 1. Approach as a model. 2. Ontology as a model and instantiation.	1. Chapter 5 2. Sections 5.5.1 and 9.2.1

	model, a method or an instantiation.	3. Procedure model. 4. Case description and content as a case model. 5. Services as instantiation. 6. Prototype as implementation and instantiation.	3. Section 5.6 4. Chapter 6 5. Chapter 7 6. Chapter 8
2. Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.	The application scenario, including derived requirements, ensures that the approach is relevant to business problems.	Chapter 4 and Section 6.4.1
3. Design Evaluation	The utility, quality and efficacy of a design artefact must be rigorously demonstrated via well-executed evaluation methods	The artefact is evaluated based on a triangulated evaluation setting. The demonstrator is evaluated in three independent settings.	Chapter 9
4. Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artefact, design foundations and design methodologies.	The new integrated case-based reasoning and process execution approach has been presented in a way that is reproducible and published in several articles and papers.	Chapter 5, Sections 5.6 and 10.1
5. Research Rigour	Design-science research relies upon the	This research relies on design science research as	Chapter 3 and Section 3.4

	application of rigorous methods in both the construction and evaluation of the design artefact.	well as on case study research, requirements analysis, interviews and experiments.	
6. Design as a Search Process	The search for an effective artefact requires utilising available means to reach desired ends while satisfying laws in the problem environment.	The artefact is designed in iterative cycles. The artefact creation process has been accompanied by the evaluation partners.	Sections 3.4, 4.1 and 5.6
7. Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.	The results of this study are disseminated in peer-reviewed conference proceedings and a journal publication. The results have been presented to business partners.	Sections 10.1.2 and 10.1.3

10.4 Recommendations for Further Research

In the following section, two potential directions for future research are presented which have been derived from business (relevance) in later cycles specifically during the evaluation phase. Although these suggestions for future research were not in the scope of the research questions (limitation), it would be worthwhile to investigate them in future work.

10.4.1 Individual Similarity Configuration

As mentioned in Section 5.6.2, the procedure model for process execution guides a new instantiation of the ICEBERG-PE approach in practice. When implementing CBR, it is the ultimate goal to represent mental similarity and adaptation models as adequately as possible. Before such a configurable model can be implemented, the various individual mental models need to be elicited and consolidated from the stakeholders. The case viewpoint model (see Section 5.4) has been introduced for creating consolidated similarity models, which are dependent to the stakeholders and their concerns at the same time. The case viewpoint model allows a certain level of individual configuration, although it is on a higher abstraction level, such as the stakeholder's role, and not the stakeholder himself/herself.

The end user can select the individualisation at any stage of the CBR cycle. However, this grade of individuality is not sufficient, because it is not always possible to fully agree on a consolidated similarity configuration as mentioned in the evaluation Section 9.3.2.

Individual Similarity Pre-Configuration: A first possibility to realise an individual configuration would be to use the viewpoint model on an individual basis to create individual similarity configurations, as shown for evaluation purposes in Section 9.3.2. Moreover, individual similarity configurations would require an additional effort when instantiating such a CBR approach. Since it has been demanded by the end user during evaluation, it would be worthwhile to investigate how individual similarity configuration can be realised within a pre-configuration step.

Adjustable Similarity Configuration: Kowalski et al. (2012; 2013) introduced a CBR approach in the logistics domain where the similarity functions and weights can be adjusted before running the retrieval. Figure 10.1 shows an input form for setting the similarity functions, which contains project-related knowledge from the logistics domain. The end user can adjust the weights using slider controls and select a certain similarity function.

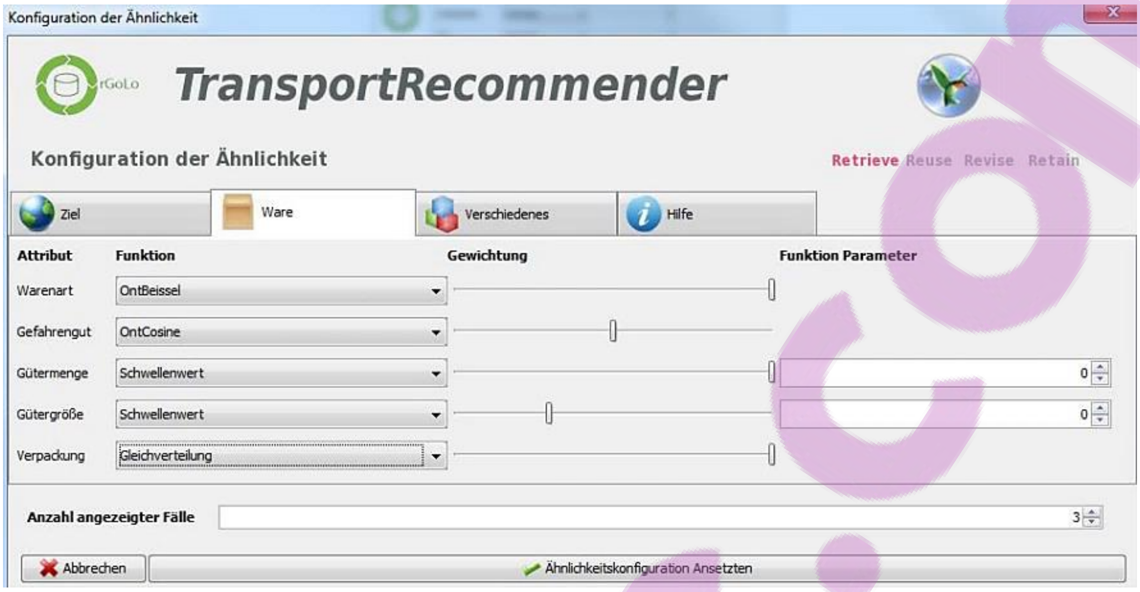

Figure 10.1: Input Mask (in German) for the Setting of the Similarity Functions During Retrieval Phase (Kowalski et al., 2013)

Figure 10.2 shows a similar input from where an individual configuration can be related to an existing profile similar to the viewpoint model of the approach introduced in this thesis. Kowalski implemented this approach for retrieving logistic cases.

Both examples of Kowalski et al. (2012) and Kowalski et al. (2013) are remarkable in terms of allowing and providing a possibility for end users to configure the CBR system individually. Unfortunately, both examples use an ontology that focuses on a hierarchical structure, which has more of a taxonomical characteristic. The approach in this thesis has more of the characteristic of a graph, which means that the ICEBERG ontology uses the global and local similarity principle to attach the similarity configurations to relations of a domain ontology.

The approach of Kowalski et al. (2012, 2013) represents a hierarchical structure as a flat list of attributes which can be adjusted. This characteristic makes it difficult for end users to recognise the effects of changing the weights. The feedback from the end users during the evaluation workshops of this thesis was that it would be difficult for them to understand the effect of a similarity function change. To select an appropriate similarity function requires profound knowledge about the mechanism itself and may require testing cycles.

Therefore, it would be worthwhile to investigate how the end users can make or adjust individual similarity configurations in a cognitively adequate manner.

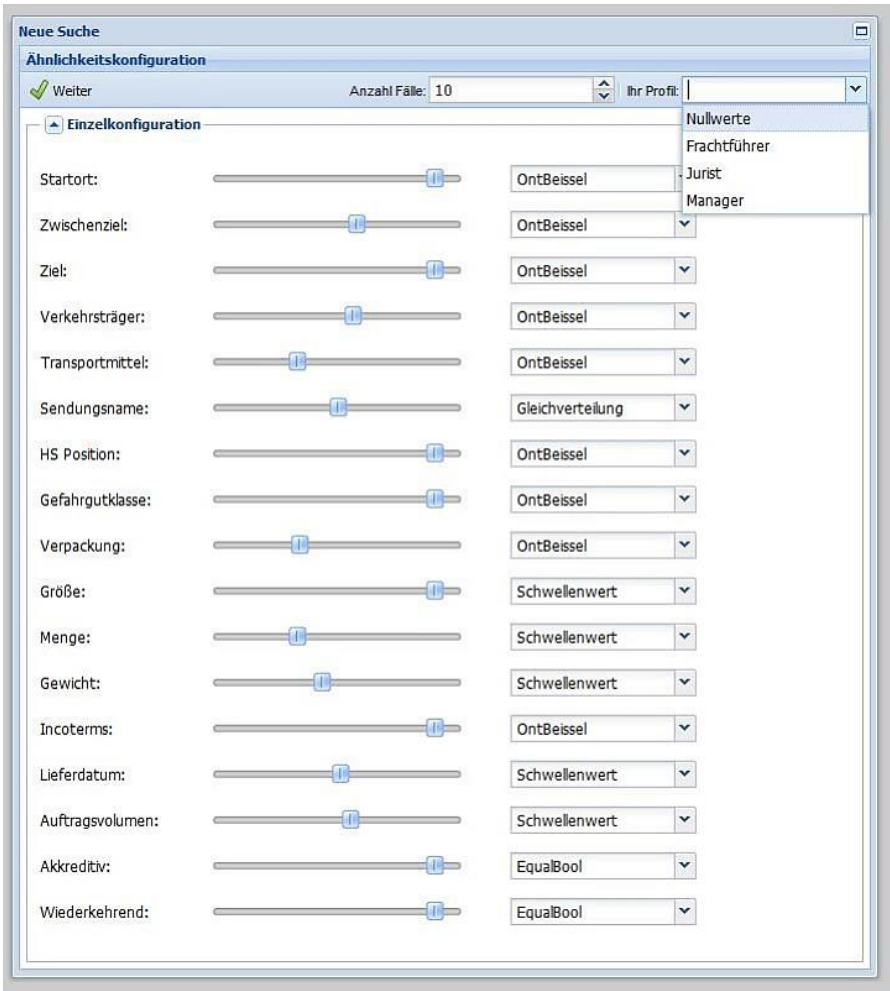

Figure 10.2: Input Mask (in German) for the Setting of the Similarity Functions and Assigning the Configuration to a Profile (Kowalski et al., 2012)

10.4.2 Visualisation of Similarity Configuration

Related to the configuration of the similarity introduced in the previous section, the visualisation of the similarity configuration and the characterisation should be further investigated as well. As shown in Figure 10.1 and Figure 10.2 in the previous section, the end user cannot recognise how a graph-oriented characterisation would be adjustable in a more or less flat structure. The ICEBERG-PE approach provides a graph-oriented visualisation of a specific case characterisation. In this case, the end user can recognise where a specific characterisation feature is located within the graph. However, the similarity configuration is hidden. During the evaluation of the approach, end users suggested that the similarity configuration should be visible (see Section 9.3.2).

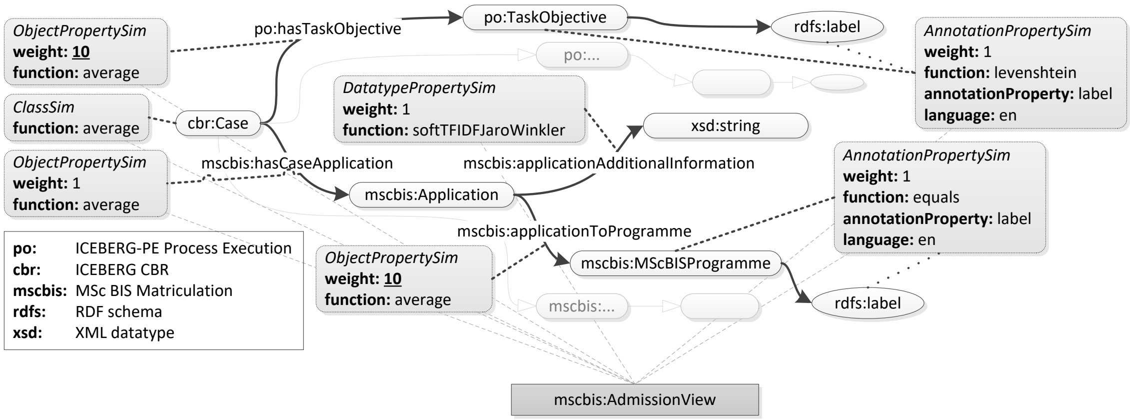

Figure 10.3: Example of a Complex Similarity Configuration.

Such a visualisation would require further research since the global and local principle is not trivial to visualise. Figure 10.3 shows an example of a complex similarity configuration derived from the admission process scenario including one viewpoint. Figure 10.3 shows all required elements for configuring the ICEBERG-PE similarity model for a part of the admission process similarity configuration. Although not all elements are required to be shown to the end user, some elements (such as an object, annotation or datatype property similarity annotations) should be transferred to a cognitively adequate visualisation. Nevertheless, it is assumed that a similarity visualisation would enhance the user acceptance of a CBR approach. Therefore, it would be worthwhile to investigate how the end users can recognise the similarity configuration in a cognitively adequate representation.

10.5 In Closing

A solid design science research project should balance relevance and rigour, business requirements and applicable knowledge as equally as possible.

This balance means that the research objectives and research questions should root both from a literature gap and business requirements derived from analysing a real-world application scenario. The research questions of this thesis have been selected based on a literature review for identifying a research problem and with the usage of the admission process application scenario. The research questions have been answered stepwise in an iterative development process.

In addition to having well-balanced research questions, a design since research project must produce certain artefacts which can contribute back to the knowledge base and environment. The outcomes of this thesis are an approach, ontology, procedure model, case content model, case characterisation model and prototypical implementation. Since this research is applied research, the artefacts should contribute to the (business) environment. Although it was not the intention to produce artefacts, which can be considered as products, the artefacts are available for all the business partners of the application scenarios (admission process scenario, project management and offering scenario) to be reproduced towards a productive instantiation. Furthermore, the artefacts have been generalised and described. The artefacts have contributed to the knowledge base (rigour) through writings, conference papers and a journal article (see 10.1.3).

Finally, research should point out further research directions, as is described in Section 10.4, because further truth builds on previous discoveries (Kling et al., 2015).

Bibliography

van der Aalst, W., Weske, M. and Grunbauer, D., 2005. Case handling: a new paradigm for business process support. *Data & Knowledge Engineering*, 53(2), pp.129–162.

Aamodt, A., 1991. *A Knowledge-Intensive, Integrated Approach to Problem Solving and Sustained Learning*. Universitetet I Trondheim.

Aamodt, A., 2004. Knowledge-Intensive Case-Based Reasoning in CREEK. In: P. Funk and P. González Calero, eds., *7th European Conference, ECCBR 2004, Madrid, Spain*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.1–15.

Aamodt, A. and Plaza, E., 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1), pp.39–59.

Abecker, A., Bernardi, A., Hinkelmann, K., Kuhn, O. and Sintek, M., 1998. Toward a technology for organizational memories. *IEEE Intelligent Systems and their Applications*, 13(3), pp.40–48.

Adams, M., Edmond, D. and ter Hofstede, A.H.M., 2003. The application of activity theory to dynamic workflow adaptation issues. In: *Proceedings of the 2003 Pacific Asia Conference on Information Systems (PACIS 2003)*. pp.1836–1852.

Aha, D. and Breslow, L., 1997. Refining conversational case libraries. In: D. Leake and E. Plaza, eds., *Case-Based Reasoning Research and Development SE - 26*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.267–278.

Aha, D., Maney, T. and Breslow, L., 1998. Supporting dialogue inferencing in conversational case-based reasoning. In: B. Smyth and P. Cunningham, eds., *Advances in Case-Based Reasoning SE - 23*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.262–273.

Aha, D.W., Breslow, L.A. and Muñoz-Avila, H., 2001. Conversational Case-Based Reasoning. *Applied Intelligence*, 14(1), pp.9–32.

Allemang, D. and Hendler, J.A., 2008. *Semantic Web for the Working Ontologist:*

Effective Modeling in RDFS and OWL. San Francisco: Morgan Kaufmann/Elsevier.

Althoff, K.-D., 1997. *Evaluating case-based reasoning systems: The Inreca case study*. *Postdoctoral thesis (Habilitationsschrift), University …*. University of Kaiserslautern.

Amailef, K. and Lu, J., 2013. Ontology-supported case-based reasoning approach for intelligent m-Government emergency response services. *Decision Support Systems*, 55(1), pp.79–97.

Anglia Ruskin University, 2012. *Guide to the Harvard Style of Referencing*.

Assali, A.A., Lenne, D. and Debray, B., 2010. Heterogeneity in Ontological CBR Systems. In: S. Montani and L. Jain, eds., *Successful Case-based Reasoning Applications - I SE - 5*, Studies in Computational Intelligence. Springer Berlin Heidelberg, pp.97–116.

Atkinson, R.C. and Shiffrin, R.M., 1968. Human Memory: A Proposed System and its Control Processes. pp.89–195.

Avison, D.E., Lau, F., Myers, M.D. and Nielsen, P.A., 1999. Action research. *Communications of the ACM*, 42(1).

Bach, K. and Althoff, K.-D., 2012. Developing Case-Based Reasoning Applications Using myCBR 3. In: B. Agudo and I. Watson, eds., *Case-Based Reasoning Research and Development SE - 4*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.17–31.

Bello-Tomás, J.J., González-Calero, P.A. and Díaz-Agudo, B., 2004. JColibri: An Object-Oriented Framework for Building CBR Systems. In: P. Funk and P. González Calero, eds., *Advances in Case-Based Reasoning SE - 4*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.32–46.

Bergmann, R., 1998. On the Use of Taxonomies for Representing Case Features and Local Similarity Measures. In: L. Gierl and M. Lenz, eds., *Proceedings of the Sixth German Workshop on CBR*. Rostock, Germany: Univ. of Rostock.

Bergmann, R., 2002. *Experience Management: Foundations, Development Methodology, and Internet-Based Applications*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg.

Bergmann, R., Althoff, K.-D., Breen, S., Göker, M., Manago, M., Traphöner, R. and Wess, S., 2003. *Developing Industrial Case-Based Reasoning Applications*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg.

Bergmann, R., Althoff, K.-D.K., Minor, M., Reichle, M. and Bach, K., 2009. Case-Based Reasoning - Introduction and Recent Developments. *Künstliche Intelligenz: Special Issue on Case-Based Reasoning*, 23(1), pp.5–11.

Bergmann, R., Freßmann, A., Maximini, K., Maximini, R. and Sauer, T., 2006. Case-Based Support for Collaborative Business. In: T. Roth-Berghofer, M. Göker and H.A. Güvenir, eds., *8th European Conference, ECCBR 2006 Fethiye, Turkey, September 4-7, 2006 Proceedings*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.519–533.

Bergmann, R. and Gil, Y., 2011. Retrieval of Semantic Workflows with Knowledge Intensive Similarity Measures. In: A. Ram and N. Wiratunga, eds., *19th International Conference on Case-Based Reasoning, ICCBR 2011*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.17–31.

Bergmann, R., Kolodner, J.L. and Plaza, E., 2005. Representation in case-based reasoning. *The Knowledge Engineering Review*, 20(3), pp.209–213.

Bergmann, R. and Schaaf, M., 2003. Structural Case-Based Reasoning and Ontology-Based Knowledge Management: A Perfect Match? *Journal of Universal Computer Science*, 9(7), pp.608–626.

Bergmann, R. and Wilke, W., 1996. On the role of abstraction in case-based reasoning. In: *Third European Workshop EWCBR-96 Lausanne, Switzerland, November 14–16, 1996 Proceedings*.

Bichindaritz, I., 2004. Mémoire: Case Based Reasoning Meets the Semantic Web in Biology and Medicine. In: P. Funk and P. González Calero, eds., *Advances in Case-Based Reasoning SE - 5*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.47–61.

Biggam, J., 2008. *Succeeding with Your Master's Dissertation - A step-by-step handbook*. Open University Press.

Bingham, W. van D. and Moore, B.V., 1941. *How to interview*. Third revi ed. New York: Harper & Brothers.

Bogaerts, S. and Leake, D., 2005. *IUCBRF: A framework for rapid and modular CBR system development*.

Bouhana, A., Zidi, A., Fekih, A., Chabchoub, H. and Abed, M., 2015. An ontology-based CBR approach for personalized itinerary search systems for sustainable urban freight transport. *Expert Systems with Applications*, 42(7), pp.3724–3741.

Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L. and Stuckenschmidt, H., 2004. Contextualizing ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(4), pp.325–343.

Brander, S., Hinkelmann, K., Hu, B., Martin, A., Riss, U., Thönssen, B. and Witschel, H., 2011a. Refining Process Models through the Analysis of Informal Work Practice.

Brander, S., Hinkelmann, K., Martin, A. and Thönssen, B., 2011b. Mining of Agile Business Processes. In: *2011 AAAI Spring Symposium Series*. pp.9–14.

Brun, R., 2010. *Linked Enterprise Models and Objects representing Content and Context of an Enterprise*. University of Applied Sciences Northwestern

Switzerland FHNW.

Bulu, H., Alpkocak, A. and Balci, P., 2012. Ontology-based mammography annotation and Case-based Retrieval of breast masses. *Expert Systems with Applications*, 39(12), pp.11194–11202.

Chandler, P. and Sweller, J., 1991. Cognitive Load Theory and the Format of Instruction. *Cognition and Instruction*, 8(4), pp.293–332.

Chen, H., Perich, F., Finin, T. and Joshi, A., 2004. SOUPA: standard ontology for ubiquitous and pervasive applications. In: *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004.* IEEE, pp.258–267.

Chen, H. and Wu, Z., 2003. On case-based knowledge sharing in semantic Web. In: *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*. Washington, DC, USA: IEEE Computer Society, pp.200–207.

Cheng, B.H.C. and Atlee, J.M., 2007. Research Directions in Requirements Engineering. In: *Future of Software Engineering (FOSE '07)*. IEEE, pp.285–303.

Claes, J., Gailly, F. and Poels, G., 2013. Cognitive Aspects of Structured Process Modeling. In: *CAiSE 2013 International Workshops, Valencia, Spain, June 17-21, 2013. Proceedings*. pp.168–173.

Claes, J., Vanderfeesten, I., Gailly, F., Grefen, P. and Poels, G., 2015. The Structured Process Modeling Theory (SPMT) a cognitive view on why and how modelers benefit from structuring the process of process modeling. *Information Systems Frontiers*, 17(6), pp.1401–1425.

Cognini, R., 2015. *A Features Based Approach to Model Variability of Business Processes*. SCHOOL OF ADVANCED STUDIES - Doctoral course in Information Science and Complex Systems (XXVII cycle).

Cognini, R., Corradini, F., Polini, A. and Re, B., 2015a. Extending Feature Models to Express Variability in Business Process Models. In: A. Persson and J. Stirna, eds., *Advanced Information Systems Engineering Workshops: CAiSE 2015 International Workshops, Stockholm, Sweden, June 8-9, 2015, Proceedings*. Cham: Springer International Publishing, pp.245–256.

Cognini, R., Corradini, F., Polini, A. and Re, B., 2015b. Using Data-Object Flow Relations to Derive Control Flow Variants in Configurable Business Processes. In: F. Fournier and J. Mendling, eds., *Business Process Management Workshops: BPM 2014 International Workshops, Eindhoven, The Netherlands, September 7-8, 2014, Revised Papers*. Cham: Springer International Publishing, pp.210–221.

Cognini, R., Hinkelmann, K. and Martin, A., 2016. A Case Modelling Language for Process Variant Management in Case-Based Reasoning. In: M. Reichert and A.H. Reijers, eds., *Business Process Management Workshops: BPM 2015, 13th International Workshops, AdaptiveCM 2015: 4th International Workshop on Adaptive Case Management and other non-workflow approaches to BPM, Innsbruck, Austria,* *August 31 -- September 3, 2015, Revised Pa*. Cham: Springer International Publishing, pp.30–42.

Cohen, D. and Crabtree, B., 2006. *Triangulation*. [online] Qualitative Research Guidelines Project, Robert Wood Johnson Foundation. Available at: <http://www.qualres.org/HomeTria-3692.html> [Accessed 26 May 2016].

Cohen, W.W., Ravikumar, P.D. and Fienberg, S.E., 2003. A Comparison of String Distance Metrics for Name-Matching Tasks. In: S. Kambhampati and C.A. Knoblock, eds., *Proceedings of IJCAI-03 Workshop on Information Integration on the Web*. Acapulco, Mexico, pp.73–78.

Cook, J., Schmidt, A., Bradley, C., Barnes, S.-A., Bimrose, J., Brander, S., Braun, S., Brown, A., Kump, B., Kunzmann, C., Mazarakis, A., Nelkner, T., Pearson, C. and Taylor, I., 2012. *D6.4 Summative Evaluation Report*.

Coyle, L., Doyle, D. and Cunningham, P., 2004. Representing Similarity for CBR in XML. In: *Advances in CaseBased Reasoning*. pp.119–127.

Coyle, L., Hayes, C. and Cunningham, P., 2002. *Representing Cases for CBR in XML*.

Creswell, J.W., 2008. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. 3rd ed. ed. Los Angeles: Sage Publications, Inc.

Cunningham, P., 2009. A Taxonomy of Similarity Mechanisms for Case-Based Reasoning. *IEEE Transactions on Knowledge and Data Engineering*, 21(11), pp.1532–1543.

D'Aquin, M., Badra, F., Lafrogne, S., Lieber, J., Napoli, A. and Szathmary, L., 2007. Case Base Mining for Adaptation Knowledge Acquisition. In: *Twentieth International Joint Conference on Artificial Intelligence - IJCAI'07*. pp.750–755.

d'Aquin, M., Lieber, J. and Napoli, A., 2005. Decentralized Case-Based Reasoning for the Semantic Web. In: Y. Gil, E. Motta, V.R. Benjamins and M. Musen, eds., *The Semantic Web – ISWC 2005 SE - 13*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.142–155.

d'Aquin, M., Lieber, J. and Napoli, A., 2006. Case-Based Reasoning Within Semantic Web Technologies. In: J. Euzenat and J. Domingue, eds., *Artificial Intelligence: Methodology, Systems, and Applications SE - 21*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.190–200.

Dadam, P., Reichert, M. and Rinderle-Ma, S., 2010. Prozessmanagementsysteme. *Informatik-Spektrum*, 34(4), pp.364–376.

Davenport, T.H., 2005a. Knowledge Work Processes. In: *Thinking for a Living: How to Get Better Performance and Results from Knowledge Workers*.

Davenport, T.H., 2005b. *Thinking for a Living: How to Get Better Performances And Results from Knowledge Workers*. Harvard Business Review Press.

Davenport, T.H., 2010. Process Management for Knowledge Work. In: J. vom Brocke and M. Rosemann, eds., *Handbook on Business Process Management 1*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.17–36.

Delir Haghighi, P., Burstein, F., Zaslavsky, A. and Arbon, P., 2013. Development and evaluation of ontology for intelligent decision support in medical emergency management for mass gatherings. *Decision Support Systems*, 54(2), pp.1192–1204.

Denzin, N.K., 2009. *The Research Act: A Theoretical Introduction to Sociological Methods*. Transaction Publishers.

Denzin, N.K. and Lincoln, Y.S., 2011. Introduction: Disciplining the Practice of Qualitative Research. In: N.K. Denzin and Y.S. Lincoln, eds., *The SAGE Handbook of Qualitative Research*, Fourth Edi. Sage Publications, Inc., pp.1–19.

Dey, A.K., 2001. Understanding and Using Context. *Personal and Ubiquitous Computing*, pp.4–7.

Dey, A.K. and Abowd, G.D., 1999. *Towards a better understanding of context and context-awareness*.

Díaz-Agudo, B. and González-Calero, P.A., 2000. An Architecture for Knowledge Intensive CBR Systems. In: E. Blanzieri and L. Portinale, eds., *Advances in Case-Based Reasoning*, Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin / Heidelberg, pp.37–48.

Dıaz-Agudo, B. and González-Calero, P., 2001. Knowledge Intensive CBR Made Affordable. In: R. Weber and C. Gresse von Wangenheim, eds., *Proceedings of the Workshop Program at the Fourth International Conference on Case-Based Reasoning*. Washington, DC, USA: Navy Center for Applied Research in Artificial Intelligence.

Dıaz-Agudo, B., González-Calero, P.A., Gómez-Martın, P.P. and Gómez-Martın, M.A., 2005. On Developing a Distributed CBR Framework through Semantic Web Services. In: *Workshop OWL: Experiences and Directions, at International Conference on Rules and Rule Markup Languages for the Semantic Web*.

Durant-Law, G., 2005. Soft systems methodology and grounded theory combined - a knowledge management research approach? *actKM Online Journal of Knowledge Management*, 2(1), pp.13–23.

Eberle, H., Unger, T. and Leymann, F., 2009. Process fragments. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. pp.398–405.

Ehrig, M., Haase, P., Hefke, M. and Stojanovic, N., 2004. Similarity for Ontologies - a Comprehensive Framework. In: H. Kühn, ed., *In Workshop Enterprise Modelling and Ontology: Ingredients for Interoperability, at PAKM 2004*. Vienna, Austria: University of Vienna.

El-Farr, H.K., 2009. *Knowledge Work and Workers: A Critical Literature Review*.

El-Sappagh, S., Elmogy, M. and Riad, A.M., 2015. A fuzzy-ontology oriented case-based reasoning framework for semantic diabetes diagnosis. *Artificial intelligence in medicine*, 65(3), pp.179–208.

van Elst, L., Aschoff, F., Bernardi, A. and Schwarz, S., 2003. Weakly-structured workflows for knowledge-intensive tasks: an experimental evaluation. In: *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.* IEEE Comput. Soc, pp.340–345.

Emmenegger, S., Hinkelmann, K., Laurenzi, E., Martin, A., Thönssen, B., Witschel, H.-F. and Zhang, C., 2017. An Ontology-based and Case-based Reasoning supported Workplace Learning Approach. In: *Model-Driven Engineering and Software Development, Communications in Computer and Information Science (CCIS), In Press*. Springer Berlin / Heidelberg, p.23.

Emmenegger, S., Laurenzni, E. and Thönssen, B., 2012. Improving Supply-Chain-Management based on Semantically Enriched Risk Descriptions. In: K. Liu and J. Filipe, eds., *Proceedings of the International Conference on Knowledge Management and Information Sharing*. SciTePress - Science and and Technology Publications, pp.70–80.

Endl, R., Knolmayer, G. and Pfahrer, M., 1998. Modeling Processes and Workflows by Business Rules. In: *1st European Workshop on Workflow and Process Management (WPM'98)*.

ESPRIT Consortium AMICE ed., 1989. *Open System Architecture for CIM*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Feldkamp, D., Hinkelmann, K. and Thönssen, B., 2007. KISS – Knowledge-Intensive Service Support: An Approach for Agile Process Management. In: A. Paschke and Y. Biletskiy, eds., *Advances in Rule Interchange and Applications, International Symposium, RuleML 2007, Orlando, Florida, October 25-26, 2007*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.25–38.

Fidjeland, M.K., 2006. *Distributed Knowledge in Case-Based Reasoning : Knowledge Sharing and Reuse within the Semantic Web*. Norwegian University of Science and Technology.

Fox, M.S., Barbuceanu, M. and Gruninger, M., 1996. An organisation ontology for enterprise modeling: Preliminary concepts for linking structure and behaviour. *Computers in Industry*, 29(1–2), pp.123–134.

Fox, M.S., Barbuceanu, M., Gruninger, M. and Lin, J., 1998. An Organisation Ontology for Enterprise Modeling. In: M. Prietula, K. Carley and L. Gasser, eds., *Simulating Organizations: Computational Models of Institutions and Groups*. Menlo Park CA: AAAI/MIT Press, pp.131–152.

Gao, J. and Deng, G., 2010. Semi-automatic Construction of Ontology-based

CBR System for Knowledge Integration. *International Journal of Electrical and Electronics Engineering*, 4(3), pp.297–303.

Greenberg, J., 2005. Understanding Metadata and Metadata Schemes. *Cataloging & Classification Quarterly*, 40(3–4), pp.17–36.

Gregg, D.G., Kulkarni, U.R. and Vinzé, A.S., 2001. Understanding the Philosophical Underpinnings of Software Engineering Research in Information Systems. *Information Systems Frontiers*, 3(2), pp.169–183.

Gruber, T.R., 1993. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2), pp.199–221.

Guba, E.G. and Lincoln, Y.S., 1994. Competing Paradigms in Qualitative Research. In: N.K. Denzin and Y.S. Lincoln, eds., *Handbook of Qualitative Research*. Thousand Oaks, CA: Sage Publications, Inc., pp.105–117.

von Halle, B. and Goldberg, L., 2009. *The Decision Model*. Auerbach Publications.

Hanney, K. and Keane, M., 1997. The adaptation knowledge bottleneck: How to ease it by learning from cases. In: D. Leake and E. Plaza, eds., *Second International Conference on Case-Based Reasoning, ICCBR-97 Providence, RI, USA, July 25–27, 1997*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.359–370.

Hart, P., 1968. The condensed nearest neighbor rule (Corresp.). *IEEE Transactions on Information Theory*, 14(3), pp.515–516.

Hefke, M., 2004. A framework for the successful introduction of KM using CBR and semantic web technologies. *Journal of Universal Computer Science*, 10(6), pp.731–739.

Hefke, M., 2008. *Ontologiebasierte Werkzeuge zur Unterstützung von Organisationen bei der Einführung und Durchführung von Wissensmanagement*. Universität Fridericiana zu Karlsruhe.

Hefke, M. and Abecker, A., 2006a. A CBR-Based Approach for Supporting Consulting Agencies in Successfully Accompanying a Customer's Introduction of Knowledge Management. In: T.R. Roth-Berghofer, M.H. Göker and H.A. Güvenir, eds., *8th European Conference, ECCBR 2006 Fethiye, Turkey, September 4-7, 2006 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.534–548.

Hefke, M. and Abecker, A., 2006b. KMIR - a Knowledge Management Implementation and Recommendation Framework using CBR and Semantic Web Technologies. In: F. Lehner, H. Nösekabel and P. Kleinschmidt, eds., *CBR and Semantic Web Technologies, Proc. MKWI-2006*. Berlin.

Hefke, M., Zacharias, V., Abecker, A. and Wang, Q., 2006. An Extendable Java Framework for Instance Similarities in Ontologies. In: Y. Manolopoulos, J. Filipe, P. Constantopoulos and J. Cordeiro, eds., *Proceedings of the 8th International Conference on Enterprise Information Systems: Databases and*

Information Systems Integration (ICEIS 2006). Paphos, Cyprus: INSTICC, Setúbal, Portugal, pp.263–269.

van Heijst, G., Schreiber, A.T. and Wielinga, B., 1997. Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies*, 46(2–3), pp.183–292.

Hemingway, E., 1932. *Death in the afternoon*. New York: Charles Scribner's Sons.

Hevner, A. and Chatterjee, S., 2010. *Design Research in Information Systems*. Integrated Series in Information Systems. Boston, MA: Springer US.

Hevner, A.R., March, S.T., Park, J. and Ram, S., 2004. Design science in information systems research. *Mis Quarterly*, 28(1), pp.75–105.

Hinkelmann, K., 2014. *Business Logic and Process Logic*.

Hinkelmann, K., 2016. Business Process Flexibility and Decision-Aware Modeling—The Knowledge Work Designer. In: D. Karagiannis, H.C. Mayr and J. Mylopoulos, eds., *Domain-Specific Conceptual Modeling*. Cham: Springer International Publishing, pp.397–414.

Hinkelmann, K., Gerber, A., Karagiannis, D., Thoenssen, B., van der Merwe, A. and Woitsch, R., 2015. A new paradigm for the continuous alignment of business and IT: Combining enterprise architecture modelling and enterprise ontology. *Computers in Industry*.

Hinkelmann, K., Karagiannis, D. and Telesko, R., 2002. PROMOTE – Methodologie und Werkzeug für geschäftsprozessorientiertes Wissensmanagement. In: A. Abecker, ed., *Geschäftsprozessorientiertes Wissensmanagement*.

Hinkelmann, K., Merelli, E. and Thönssen, B., 2010. The Role of Content and Context in Enterprise Repositories. In: L.T. Ly, L.H. Thorn, S. Rinderle-Ma, A. Gerber, K. Hinkelmann, P. Kotze, U. Reimer, A. van der Merwe, W. Mansoor, S. Elnaffar and V. Monfort, eds., *2nd International Workshop on Advanced Enterprise Architecture and Repositories - AER 2010*. Madeira, Portugal: SciTePress.

Houeland, T. and Aamodt, A., 2010. The Utility Problem for Lazy Learners - Towards a Non-eager Approach. In: I. Bichindaritz and S. Montani, eds., *18th International Conference on Case-Based Reasoning, ICCBR 2010, Alessandria, Italy, July 19-22, 2010. Proceedings*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.141–155.

Hox, J.J. and Boeije, H.R., 2005. Data Collection, Primary vs. Secondary. In: K. Kempf-Leonard, ed., *Encyclopedia of Social Measurement*. Elsevier, pp.593–599.

Hruschka, P., 2012. Quality Driven Software Architecture. pp.10–13.

IEEE, 1991. *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. IEEE Std 610*, .

ISO/IEC/IEEE, 2011. *ISO/IEC/IEEE 42010: Conceptual Model*. [online] Available at: <http://www.iso-architecture.org/42010/cm/> [Accessed 25 Jun. 2014].

Ju, K., Su, B., Zhou, D. and Zhang, Y., 2016. An incentive-oriented early warning system for predicting the co-movements between oil price shocks and macroeconomy. *Applied Energy*, 163, pp.452–463.

Kahn, R.L. and Cannell, C.F., 1957. *The dynamics of interviewing: theory, technique, and cases.* Oxford, England: John Wiley & Sons.

Kamsu-Foguem, B. and Noyes, D., 2013. Graph-based reasoning in collaborative knowledge management for industrial maintenance. *Computers in Industry*, 64(8), pp.998–1013.

Kang, D., Lee, J., Choi, S. and Kim, K., 2010. An ontology-based Enterprise Architecture. *Expert Systems with Applications*, 37(2), pp.1456–1464.

Kapetanakis, S., Petridis, M., Knight, B., Ma, J. and Bacon, L., 2010. A Case Based Reasoning Approach for the Monitoring of Business Workflows. In: I. Bichindaritz and S. Montani, eds., *18th International Conference on Case-Based Reasoning, ICCBR 2010, Alessandria, Italy, July 19-22, 2010. Proceedings*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.390–405.

Kaster, D.S., Medeiros, C.B. and Rocha, H. V., 2005. Supporting modeling and problem solving from precedent experiences: the role of workflows and case-based reasoning. *Environmental Modelling & Software*, 20(6), pp.689–704.

Kim, J., Suh, W. and Lee, H., 2002. Document-based workflow modeling: a case-based reasoning approach. *Expert Systems with Applications*, 23(2), pp.77–93.

Kling, J., Manrodt, K., Vitasek, K. and Keith, B., 2015. *Strategic Sourcing in the New Economy: Harnessing the Potential of Sourcing Business Models for Modern Procurement*. Palgrave Macmillan US.

Knublauch, H., Hendler, J.A. and Idehen, K., 2011. *SPIN: SPARQL Inferencing Notation*. [online] W3C Member Submission. Available at: <https://www.w3.org/Submission/spin-overview/> [Accessed 22 Feb. 2011].

Kolodner, J.L., 1993. *Case-based reasoning*. San Mateo, CA: Morgan Kaufmann Publishers.

Kowalski, M., Klüpfel, H., Zelewski, S., Bergenrodt, D. and Saur, A., 2013. Integration of Case-Based and Ontology-Based Reasoning for the Intelligent Reuse of Project-Related Knowledge. In: U. Clausen, M. ten Hompel and M. Klumpp, eds., *Efficiency and Logistics*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.289–299.

Kowalski, M., Zelewski, S., Bergenrodt, D. and Klupfel, H., 2012. Application of new techniques of artificial intelligence in logistics: an ontology-driven case-based reasoning approach. In: *Proceedings of European Simulation and Modelling Conference*. pp.323–328.

Lankhorst, M., 2009. *Enterprise Architecture at Work*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Lankhorst, M.M., Proper, H.A. and Jonkers, H., 2009. The Architecture of the ArchiMate Language. In: T. Halpin, J. Krogstie, S. Nurcan, E. Proper, R. Schmidt, P. Soffer, R. Ukor, W. Aalst, J. Mylopoulos, M. Rosemann, M.J. Shaw and C. Szyperski, eds., *Enterprise, Business-Process and Information Systems Modeling*, Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, pp.367–380.

Leake, D. and Kendall-Morwick, J., 2008. Towards Case-Based Support for e-Science Workflow Generation by Mining Provenance. In: K.-D. Althoff, R. Bergmann, M. Minor and A. Hanft, eds., *9th European Conference, ECCBR 2008*. Trier, Germany: Springer Berlin Heidelberg, pp.269–283.

Leake, D. and Wilson, D., 1998. Categorizing case-base maintenance: Dimensions and directions. In: B. Smyth and P. Cunningham, eds., *4th European Workshop, EWCBR-98 Dublin, Ireland, September 23–25, 1998 Proceedings*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.196–207.

Leake, D.B., 1996. CBR in Context: The Present and Future. In: D.B. Leake, ed., *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. Menlo Park: AAAI Press/MIT Press, pp.1–35.

Leake, D.B. and Wilson, D.C., 2000. Remembering Why to Remember: Performance-Guided Case-Base Maintenance. In: E. Blanzieri and L. Portinale, eds., *5th European Workshop, EWCBR 2000 Trento, Italy, September 6–9, 2000 Proceedings*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.161–172.

Leppänen, M., 2007. A Context Based Enterprise Ontology. In: A. W., ed., *10th International Conference on Business Information Systems (BIS 2007)*. Poznan, Poland: Springer Berlin / Heidelberg, pp.273–286.

López, B., Pous, C., Gay, P., Pla, A., Sanz, J. and Brunet, J., 2011. eXiT*CBR: A framework for case-based medical diagnosis development and experimentation. *Artificial Intelligence in Medicine*, 51(2), pp.81–91.

Lopez de Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M. Lou, Cox, M.T., Forbus, K., Keane, M., Aamodt, A. and Watson, I., 2006. Retrieval, reuse, revision and retention in case-based reasoning. *The Knowledge Engineering Review*, 20(3), p.215.

Madhusudan, T., Zhao, J.L. and Marshall, B., 2004. A case-based reasoning framework for workflow model management. *Data & Knowledge Engineering*, 50(1), pp.87–115.

Manenti, L. and Sartori, F., 2010. Exploiting CReP for Knowledge Retrieval and Use in Complex Domains. *Metadata and Semantic Research*, 108, pp.160–171.

Markovitch, S. and Scott, P.D., 1993. Information filtering: Selection mechanisms in learning systems. *Machine Learning*, 10(2), pp.113–151.

Martin, A., 2010. *Linked Enterprise Models and Objects providing Context and Content for creating Metadata*. University of Applied Sciences Northwestern Switzerland FHNW.

Martin, A., 2013. Wissensarbeit ist nicht Routinearbeit. *UnternehmerZeitung (UZ)*, pp.32–33.

Martin, A. and Brun, R., 2010. Agile Process Execution with KISSmir. In: *5th International Workshop on Semantic Business Process Management collocated with 7th Extended Semantic Web Conference*. Heraklion, Greece.

Martin, A., Emmenegger, S., Hinkelmann, K. and Thönssen, B., 2016. A viewpoint-based case-based reasoning approach utilising an enterprise architecture ontology for experience management. *Enterprise Information Systems*, pp.1–25.

Martin, A., Emmenegger, S. and Wilke, G., 2013. Integrating an enterprise architecture ontology in a case-based reasoning approach for project knowledge. In: *Proceedings of the First International Conference on Enterprise Systems: ES 2013*. Cape Town: IEEE, pp.1–12.

Matthes, D., 2011. *Enterprise Architecture Frameworks Kompendium*. Xpert.press. Berlin, Heidelberg: Springer Berlin Heidelberg.

Maximini, K. and Maximini, R., 2007. *Collaborative Agent-Based Knowledge Engine*. *wi2.uni-trier.de*. Trier, Germany.

Maximini, K., Maximini, R. and Bergmann, R., 2003. An Investigation of Generalized Cases. In: *Case-Based Reasoning Research and Development*. p.1064.

McBride, B., 2001. Jena: Implementing the RDF model and syntax specification. In: S. Decker, D. Fensel, A.P. Sheth and S. Staab, eds., *Proceedings of the 2nd International Workshop on the Semantic Web.* Hongkong: CEUR-WS.org.

McCauley, D., 2010. Achieving Agility. In: K.D. Swenson, ed., *Mastering the Unpredictable*. Meghan-Kiffer Press, pp.257–275.

Ben Mena, T., Bellamine-Ben Saoud, N., Ben Ahmed, M. and Pavard, B., 2007. Towards a Methodology for Context Sensitive Systems Development. In: *6th International and Interdisciplinary Conference, CONTEXT 2007, Roskilde, Denmark, August 20-24, 2007. Proceedings*. pp.56–68.

Merton, R.K., 1965. *On the Shoulders of Giants: The Post-Italianate Edition*. University of Chicago Press.

Métrailler, A., 2011. Integration of change and evolution in the lifecycle of SMEs information systems. In: C. Boldyreff, S. Islam, M. Leonard and B. Thalheim, eds., *Proceedings of the CAiSE Doctoral Consortium 2011*. London, United Kingdom: CEUR-WS.org.

Miller, G.A., 1956. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 63(2), pp.81–97.

Minor, M., Bergmann, R., Görg, S. and Walter, K., 2010a. Towards Case-Based Adaptation of Workflows. In: I. Bichindaritz and S. Montani, eds., *18th International Conference on Case-Based Reasoning, ICCBR 2010*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.421–435.

Minor, M., Tartakovski, A., Schmalen, D. and Bergmann, R., 2010b. Agile Workflow Technology for Long-Term Processes: Enhanced by Case-Based Change Reuse. In: V. Sugumaran, ed., *Methodological Advancements in Intelligent Information Technologies*. IGI Global, pp.279–298.

Montani, S. and Leonardi, G., 2012. Retrieval and Clustering for Business Process Monitoring: Results and Improvements. In: B. Agudo and I. Watson, eds., *Case-Based Reasoning Research and Development SE - 21*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.269–283.

zur Muehlen, M., 2008. *How much BPMN do you need?* [online] Webblog of the Stevens Institute of Technology, Center for Business Process Innovation. Available at: <http://www.bpm-research.com/2008/03/03/how-much-bpmn-do-you-need/> [Accessed 4 Nov. 2015].

zur Muehlen, M. and Recker, J., 2013. We Still Don't Know How Much BPMN Is Enough, But We Are Getting Closer. In: *Seminal Contributions to Information Systems Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.445–451.

Muehlen, M. Zur and Recker, J., 2008. How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In: *Advanced Information Systems Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.465–479.

Müller, G. and Bergmann, R., 2015. Generalization of Workflows in Process-Oriented Case-Based Reasoning. In: *The 28th International FLAIRS Conference, At Hollywood, Florida, USA*. pp.391–396.

Nägele, R. and Schreiner, P., 2002. Potenziale und Grenzen von Business Process Management Tools für geschäftsprozessorientiertes Wissensmanagement. In: A. Abecker, ed., *Geschäftsprozessorientiertes Wissensmanagement: Effektive Wissensnutzung bei der Planung und Umsetzung von Geschäftsprozessen*. Berlin: Springer, pp.25–46.

Nelkner, T., Hu, B., Martin, A., Brander, S., Braun, S., Riss, U., Attwell, G., Hinkelmann, K. and Berrio de Diego, M., 2011. *D2.3 / D3.3 Design and Delivery of Prototype Version V2 of PLME / OLME*.

Niles, I. and Pease, A., 2001. Towards a standard upper ontology. *Formal Ontology in Information Systems*.

NISO, 2004. *Understanding Metadata. National Information Standards*. NISO Press - National Information Standards Organization.

OASIS, 2007. *Web Services Business Process Execution Language Version 2.0*. [online] Available at: <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>.

OASIS, 2012. *Web Services Human Task (WS-HumanTask) Specification Version 1.1*.

of Salisbury, J., 2009. *The Metalogicon: A Twelfth-century Defense of the Verbal and Logical Arts of the Trivium*. Paul Dry Books.

OMG, 2008. *Business Motivation Model (BMM), Version 1.0*. [online] Available at: <http://www.omg.org/spec/BMM/1.0/>.

OMG, 2011. *Business Process Model and Notation (BPMN), Version 2.0*. [online] Available at: <http://www.omg.org/spec/BPMN/2.0/>.

OMG, 2014. *Case Management Model and Notation (CMMN), Version 1.0*. [online] Available at: <http://www.omg.org/spec/CMMN/1.0/>.

OMG, 2016. *Case Management Model and Notation (CMMN), Version 1.1 - Beta*. [online] Available at: <http://www.omg.org/spec/CMMN/1.1/Beta/>.

Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A. and Sinz, E.J., 2010. Memorandum on design-oriented information systems research. *European Journal of Information Systems*, 20(1), pp.7–10.

Owen, C.L., 1998. Design research: building the knowledge base. *Design Studies*, 19(1), pp.9–20.

Palmer, N., 2011. BPM and ACM. In: K.D. Swenson, N. Palmer and B. Silver, eds., *Taming the Unpredictable Real World Adaptive Case Management: Case Studies and Practical Guidance*. Future Strategies Inc.

Patton, M.Q., 2015. *Qualitative Research & Evaluation Methods: Integrating Theory and Practice*. 4th ed ed. SAGE Publications Inc.

Peffers, K., Tuunanen, T., Rothenberger, M.A. and Chatterjee, S., 2007. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), pp.45–77.

Pla, A., López, B., Gay, P. and Pous, C., 2013. eXiT*CBR.v2: Distributed case-based reasoning tool for medical prognosis. *Decision Support Systems*, 54(3), pp.1499–1510.

Plaza, E., 1995. Cases as terms: A feature term approach to the structured representation of cases. In: M. Veloso and A. Aamodt, eds., *First International Conference, ICCBR-95 Sesimbra, Portugal, October 23–26*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.265–276.

Plaza, E., 2009. On reusing other people's experiences. *Künstliche Intelligenz*, (21), pp.18–23.

Polančič, G., 2014. *BPMN 2.0 Subclasses of Process Modeling Elements*. [online] Available at: <https://www.goodelearning.com/downloads/business-

process/subclassess-of-bpmn-process-modeling-elements>.

Potes Ruiz, P.A., Kamsu-Foguem, B. and Noyes, D., 2013. Knowledge reuse integrating the collaboration from experts in industrial maintenance management. *Knowledge-Based Systems*, 50, pp.171–186.

Recio-García, J.A., Díaz-Agudo, B. and González-Calero, P., 2008. *jCOLIBRI2 Tutorial*.

Recio-García, J.A., González-Calero, P.A. and Díaz-Agudo, B., 2014. jcolibri2: A framework for building Case-based reasoning systems. *Science of Computer Programming*, 79(Group 910494), pp.126–145.

Recio-Garía, J.A. and Díaz-Agudo, B., 2007. Ontology based CBR with jCOLIBRI. In: R. Ellis, T. Allen and A. Tuson, eds., *Proceedings of AI-2006, the Twenty-sixth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer-Verlag London, pp.149–162.

Recio, J., Díaz-Agudo, B., Gómez-Martín, M. and Wiratunga, N., 2005. Extending jCOLIBRI for Textual CBR. In: H. Muñoz-Ávila and F. Ricci, eds., *Case-Based Reasoning Research and Development SE - 33*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.421–435.

Recker, J., zur Muehlen, M., Siau, K., Erickson, J. and Indulska, M., 2009. Measuring Method Complexity : UML versus BPMN. In: *Proceedings of the Fifteenth Americas Conference on Information Systems*. Association for Information Systems.

Reichert, M. and Weber, B., 2012. *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Reinartz, T., Iglezakis, I. and Roth–Berghofer, T., 2001. Review and Restore for Case-Base Maintenance. *Computational Intelligence*, 17(2), pp.214–234.

Remenyi, D. and Williams, B., 1998. *Doing research in business and management: an introduction to process and method*. SAGE.

Richter, M.M., 1995. *The Knowledge Contained in Similarity Measures*.

Richter, M.M., 1998. Introduction. In: M. Lenz, H.-D. Burkhard, B. Bartsch-Spörl and S. Wess, eds., *Case-Based Reasoning Technology SE - 1*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.1–15.

Richter, M.M. and Weber, R.O., 2013. *Case-Based Reasoning: A Textbook*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Riesbeck, R.C. and Schank, C.K., 1989. *Inside case-based reasoning*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Rinderle, S., Weber, B., Reichert, M. and Wild, W., 2005. Integrating process learning and process evolution–a semantics based approach. *Business Process*

Management, pp.252–267.

Robson, C., 2002. *Real world research: a resource for social scientists and practitioner* ... Wiley-Blackwell.

Ross, J.W., Weill, P. and Robertson, D., 2006. *Enterprise Architecture as Strategy: Creating a Foundation for Business Execution*. Boston, MA: Harvard Business Press.

Ross, R.G. and Lam, G.S.W., 2011. *Building Business Solutions: Business Analysis with Business Rules*. Business Rule Solutions Inc.

Rossi, M. and Brinkkemper, S., 1996. Complexity metrics for systems development methods and techniques. *Information Systems*, 21(2), pp.209–227.

Roth-Berghofer, T. and Bahls, D., 2008. Explanation Capabilities of the Open Source Case-Based Reasoning Tool myCBR. In: M. Petridis and N. Wiratunga, eds., *UK Workshop on Case-Based Reasoning UKCBR 2008*. pp.23–34.

Roth-Berghofer, T., Garcia, J.A.R., Sauer, C.S., Bach, K., Althoff, K.-D., Diaz-Agudo, B. and Calero, P.A.G., 2012. Building Case-based Reasoning Applications with myCBR and COLIBRI Studio. In: M. Petridis, T. Roth-Berghofer and N. Wiratunga, eds., *Proceedings of the 17th UK Workshop on Case-Based Reasoning*. Cambridge, UK: School of Computing, Engineering and Mathematics, University of Brighton, UK, pp.71–82.

Ruch, F.L. and Zimbardo, P.G., 1974. Die therapeutische Modifikation des Verhaltens. In: *Lehrbuch der Psychologie*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.457–506.

Sadiq, S., Sadiq, W. and Orlowska, M., 2001. Pockets of flexibility in workflow specification. In: *Conceptual Modeling — ER 2001 Lecture Notes in Computer Science*. Springer, pp.513–526.

Saltzer, J.H. and Kaashoek, M.F., 2009. *Principles of Computer System Design: An Introduction*. An Introduction Series. Elsevier Science.

Saunders, M., Lewis, P. and Thornhill, A., 2007. Research Methods for Business Students.

Saunders, M., Lewis, P. and Thornhill, A., 2015. *Research methods for business students*. New York, USA: Pearson Education Limited.

Schäfermeyer, M., Rosenkranz, C. and Holten, R., 2012. The Impact of Business Process Complexity on Business Process Standardization. *Business & Information Systems Engineering*, 4(5), pp.261–270.

Schank, R.C., 1982. *Dynamic memory: a theory of reminding and learning in computers and people*. New York: Cambridge University Press.

Schank, R.C., 1999. *Dynamic Memory Revisited*. Cambridge: Cambridge University Press.

Schank, R.C. and Abelson, R.P., 1977. *Scripts, plans, goals, and understanding: an inquiry into human knowledge structures*. New York: Hillsdale, N.J. : L. Erlbaum Associates.

Schumacher, P., Minor, M., Walter, K. and Bergmann, R., 2012. Extraction of procedural knowledge from the web. In: *Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion*. New York, USA: ACM Press, p.739.

Secchi, P., Ciaschi, R. and Spence, D., 1999. A Concept for an ESA Lessons Learned System. In: P. Secchi, ed., *Proceedings of alerts and lessons learned: An effective way to prevent failures and problems (Tech. Rep. WPP-167)*. Noordwijk, Netherlands, pp.57–61.

Sene, A., Kamsu-Foguem, B. and Rumeau, P., 2015. Telemedicine framework using case-based reasoning with evidences. *Computer Methods and Programs in Biomedicine*, 121(1), pp.21–35.

Shaft, T. and Vessey, I., 2006. The role of cognitive fit in the relationship between software comprehension and modification. *MIS Quarterly*, 30(1), pp.29–55.

Shapiro, R., 2010. *Update on BPMN Release 2.0*. [online] Webinar Presentation. Available at: <http://www.wfmc.org/download-document/bpmn-2.0-industry-update-presentation.html> [Accessed 14 Dec. 2013].

Shen, Y., Colloc, J., Jacquet-Andrieu, A. and Lei, K., 2015. Emerging medical informatics with case-based reasoning for aiding clinical decision in multi-agent system. *Journal of biomedical informatics*, 56, pp.307–317.

Shimazu, H., 1998. A textual case-based reasoning system using XML on the world-wide web. In: B. Smyth and P. Cunningham, eds., *Advances in Case-Based Reasoning SE - 24*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.274–285.

Siau, K. and Rossi, M., 2011. Evaluation techniques for systems analysis and design modelling methods - a review and comparative analysis. *Information Systems Journal*, 21(3), pp.249–268.

Silver, B., 2011. *BPMN Method and Style, 2nd Edition*. Aptos, CA, USA: Cody-Cassidy Press.

Smyth, B. and Cunningham, P., 1996. The utility problem analysed. In: I. Smith and B. Faltings, eds., *Third European Workshop EWCBR-96 Lausanne, Switzerland, November 14–16, 1996 Proceedings*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.392–399.

Smyth, B. and Keane, M.T., 1995. Remembering To Forget A Competence-Preserving Case Deletion Policy for Case-Based Reasoning Systems. In: *14th International Joint Conference on Artificial Intelligence, IJCAI '95*. pp.377–383.

Smyth, B. and McKenna, E., 1999. Building Compact Competent Case-Bases. In: K.-D. Althoff, R. Bergmann and L.K. Branting, eds., *Third International Conference on Case-Based Reasoning, ICCBR-99 Seeon Monastery, Germany, July 27-30, 1999 Proceedings*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.329–342.

Sommerville, I., 2011. *Software engineering*. 9th editio ed. Boston: Pearson.

Staab, S., 2011. Ontologies and Similarity. In: A. Ram and N. Wiratunga, eds., *19th International Conference on Case-Based Reasoning, ICCBR 2011, London, UK, September 12-15, 2011. Proceedings*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.11–16.

Stahl, A. and Roth-Berghofer, T.R., 2008. Rapid Prototyping of CBR Applications with the Open Source Tool myCBR. In: K.-D. Althoff, R. Bergmann, M. Minor and A. Hanft, eds., *9th European Conference, ECCBR 2008*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp.615–629.

Starke, G. and Hruschka, P., 2011. *Software-Architektur kompakt*. IT kompakt. Heidelberg: Spektrum Akademischer Verlag.

Starke, G. and Hruschka, P., 2012. *arc42 Template for Software and System Architectures*. [online] Available at: <http://arc42.org/>.

Studer, R., Benjamins, R. V. and Fensel, D., 1998. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1–2), pp.161–197.

Sweller, J., van Merrienboer, J.J.G. and Paas, F.G.W.C., 1998. Cognitive Architecture and Instructional Design. *Educational Psychology Review*, 10(3), pp.251–296.

Swenson, K.D., 2010. *Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done*. Meghan-Kiffer Press.

Swenson, K.D., 2013. Position: BPMN Is Incompatible with ACM. In: M. La Rosa and P. Soffer, eds., *Business Process Management Workshops: BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp.55–58.

Swenson, K.D., Palmer, N., Pucher, M.J. and MD, C.W., 2012. *How Knowledge Workers Get Things Done*. Future Strategies Inc.

Swenson, K.D., Palmer, N. and Silver, B., 2011. *Taming the Unpredictable Real World Adaptive Case Management: Case Studies and Practical Guidance*. Future Strategies Inc.

The Open Group, 2009a. *ArchiMate® 1.0 Specification*. [online] Available at: <http://pubs.opengroup.org/architecture/archimate-doc/ts_archimate/>.

The Open Group, 2009b. *The Open Group Architecture Framework (TOGAF) Version 9*. Zaltbommel, Netherlands: Van Haren Publishing.

The Open Group, 2012. *ArchiMate® 2.0 Specification*. [online] Available at: <http://pubs.opengroup.org/architecture/archimate2-doc/>.

Thönssen, B., 2010. An Enterprise Ontology Building the Bases for Automatic Metadata Generation. In: S. Sánchez-Alonso and I.N. Athanasiadis, eds., *Proceedings of the 4th International Conference on Metadata and Semantics, MTSR1200*. Madrid: Springer Berlin Heidelberg, pp.195–210.

Thönssen, B. and Lutz, J., 2012. Semantically Enriched Obligation Management: An Approach for Improving the Handling of Obligations Represented in Contracts. In: A. Fred, J.L.G. Dietz, K. Liu and J. Filipe, eds., *Proceedings of 4th Conference on Knowledge Management and Information Sharing (KMIS2012)*. Barcelona, Spain: Springer Berlin Heidelberg.

Thönssen, B., Witschel, H.-F., Hinkelmann, K. and Martin, A., 2016. *Experience Knowledge Mechanisms and Representation*.

Thönssen, B. and Wolff, D., 2012. A Broader View on Context Models towards Supporting Business Process Agility. In: *Semantic Technologies for Business and Information Systems Engineering*. IGI Global, pp.337–358.

Trochim, W., 2006. *The Research Methods Knowledge Base, 2nd Edition*. [online] Available at: <http://www.socialresearchmethods.net/kb/> [Accessed 11 Apr. 2016].

University of South Africa, 2007. *Policy on Research Ethics*.

Uschold, M., King, M., Moralee, S. and Zorgios, Y., 1998. The Enterprise Ontology. *The Knowledge Engineering Review*, 13(1), pp.31–89.

Vaishnavi, V. and Kuechler, B., 2004. *Design Science Research in Information Systems*. [online] Page on Design Science Research in Information Systems (IS); last updated: November 15, 2015. Available at: <http://www.desrist.org/design-research-in-information-systems/> [Accessed 15 Nov. 2015].

Verbeek, H., Buijs, J., van Dongen, B. and van der Aalst, W., 2010. Prom 6: The process mining toolkit. *Proc. of BPM Demonstration Track*, 615, pp.34–39.

De Villiers, M.R., 2005. Three approaches as pillars for interpretive information systems research: development research, action research and grounded theory. *SAICSIT; Vol. 150*.

W3C, 2012a. *OWL 2 Web Ontology Language*. [online] Available at: <http://www.w3.org/TR/owl2-overview/>.

W3C, 2012b. *Vocabularies*. [online] Available at: <http://www.w3.org/standards/semanticweb/ontology> [Accessed 16 Dec. 2013].

W3C, 2014a. *RDF 1.1 Concepts and Abstract Syntax*. [online] Available at: <http://www.w3.org/TR/rdf11-concepts/>.

W3C, 2014b. *RDF 1.1 Turtle*. [online] Available at: <http://www.w3.org/TR/turtle/>.

W3C, 2014c. *RDF Schema 1.1*. [online] Available at: <http://www.w3.org/TR/rdf-schema/>.

Wang, Y., Hu, T. and Zhang, S., 2003. Ontology-based reconfigurable case-based reasoning system for knowledge integration. In: *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*. New York, NY, USA: IEEE, pp.4878–4883.

Wargitsch, C., Wewers, T. and Theisinger, F., 1997. WorkBrain: Merging Organizational Memory and Workflow Management Systems. In: *Proceedings of KI'97 Workshop on Knowledge-Based Systems for Knowledge Management in Enterprises*.

Watson, I., 1999. Case-based reasoning is a methodology not a technology. *Knowledge-Based Systems*, 12(5–6), pp.303–308.

Watson, I., 2003. Understanding case-based reasoning. In: I. Watson, ed., *Applying Knowledge Management: Techniques for Building Corporate Memories*. San Francisco: Elsevier, pp.23–46.

Weber, B., Reichert, M., Rinderle-Ma, S. and Wild, W., 2009. Providing Integrated Life Cycle Support in Process-Aware Information Systems. *International Journal of Cooperative Information Systems*, 18(1), pp.115–165.

Weber, B. and Wild, W., 2005. Conversational case-based reasoning support for business process management. *Architecture*.

Weber, B., Wild, W. and Breu, R., 2004. CBRFlow: Enabling Adaptive Workflow Management Through Conversational Case-Based Reasoning. In: P. Funk, G. Calero and P. A, eds., *Advances in CaseBased Reasoning*. Springer, pp.434–448.

Weber, R., Aha, D.W. and Becerra-Fernandez, I., 2000. Categorizing Intelligent Lessons Learned Systems. In: D.W. Aha and R. Weber, eds., *AAAI Technical Report WS-00-03*. Menlo Park, California: AAAI Press, pp.63–67.

Weber, R.O., Ashley, K.D. and Bruninghaus, S., 2005. Textual case-based reasoning. *The Knowledge Engineering Review*, 20(3), p.255.

WfMC, 1995. *The Workflow Reference Model*. *The Workflow Management Coalition Specification*.

Wheeler, R. and Aitken, S., 2000. Multiple algorithms for fraud detection. *Knowledge-Based Systems*, 13(2–3), pp.93–99.

Wilke, W., Vollrath, I., Altho, K.-D. and Bergmann, R., 1997. A Framework for Learning Adaptation Knowledge Based on Knowledge Light Approaches. In: *Proceedings of the Fifth German Workshop on Case-Based Reasoning*. pp.235–242.

Wilson, D.C. and Leake, D.B., 2001. Maintaining Case-Based Reasoners: Dimensions and Directions. *Computational Intelligence*, 17(2), pp.196–213.

Witschel, H.-F., Hu, B., Riss, U. V., Thönssen, B., Brun, R., Martin, A. and Hinkelmann, K., 2010. A Collaborative Approach to Maturing Process-related Knowledge. In: *In Proceedings of the 8th International Conference on Business Process Management*. New York, USA: Springer.

Witschel, H.-F., Martin, A., Emmenegger, S. and Lutz, J., 2015. A new Retrieval Function for Ontology-Based Complex Case Descriptions. In: *International Workshop Case-Based Reasoning CBR-MD 2015*. Hamburg: ibai-publishing.

Wohed, P., Russell, N., ter Hofstede, A.H.M., Andersson, B. and van der Aalst, W.M.P., 2009. Patterns-based evaluation of open source BPM systems: The cases of jBPM, OpenWFE, and Enhydra Shark. *Information and Software Technology*, 51(8), pp.1187–1216.

Xiao Hang Wang, Da Qing Zhang, Tao Gu and Hung Keng Pung, 2004. Ontology based context modeling and reasoning using OWL. In: *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second*. IEEE, pp.18–22.

Yin, R.K., 2003. *Case Study Research: Design and Methods*. 3th ed. ed. Los Angeles Calif: Sage Publications, Inc.

Yokoyama, T., 1990. An object-oriented and constraint-based knowledge representation system for design object modeling. In: *Sixth Conference on Artificial Intelligence for Applications*. New York, NY, USA: IEEE Comput. Soc. Press, pp.146–152.

Zachman, J.A., 1987. A framework for information systems architecture. *IBM Systems Journal*, 26(3).

Zachman, J.A., 2008. *John Zachman's Concise Definition of the The Zachman Framework™*. [online] Zachman International, Inc. Available at: <https://www.zachman.com/about-the-zachman-framework> [Accessed 1 Jan. 2010].

Zörner, S., 2012. *Softwarearchitekturen dokumentieren und kommunizieren*. München: Carl Hanser Verlag GmbH & Co. KG.

Appendix-A: Case Data

Table-Appx 1: Basic Data - Case A

Name	***Peter Nicolasia***
Nationality	South African
Degree	Bachelor of commerce in Business management
Final degree university	University of South Africa (UNISA)
Additional information	-
Performed activities	Analyse Application, Check Approval and Prepare Response (Rejection)
Problems	Can degree be accepted?
Solutions	No, 3-year South African bachelor of commerce cannot be accepted

Table-Appx 2: Remarks/Suggestions – Case A

Remarks or suggestions
Missing information (content) about BSc degrees in task pattern

Table-Appx 3: Basic Data - Case C

Name	***Urs Frenacher***
Nationality	Swiss
Degree	Bachelor of Law
Final degree university	University of Bern, Switzerland
Additional information	-
Performed activities	Analyse Application, Check Completeness, Acceptance (with condition)
Problems	Degree in different area
Solutions	Accept, but needs to do pre-master (condition)

Table-Appx 4: Remarks/Suggestions – Case C

Remarks or suggestions
Including more fields into the initial entry form (e.g. adding the field of study, thus allowing to make better recommendations) was considered a bad idea because of the high effort of entering it
The participant repeatedly used phrases like "I have my own…", "I'm doing this a bit differently…", indicating that she will follow her own way in many situations, sees sharing of resources/experience between secretaries critical

Table-Appx 5: Basic Data - Case D

Name	***Andrea Andanti***
Nationality	Swiss
Degree	None
Final degree university	FHNW, Switzerland
Additional information	Student is studying BK (PT) in Brugg, plans to finish Bachelor degree in September 2012
Performed activities	Analyse Application, Check Completeness, Acceptance (with condition)
Problems	Bachelor Degree missing
Solutions	Accept, but he has to graduate (condition)

Table-Appx 6: Remarks/Suggestions – Case D

Remarks or suggestions
Process model: here, participant would start with the activity "check completeness of certificates" because it is obviously problematic here
Participant would like to get a reminder (via a task or email) about having to ask the student for the certificate later

Table-Appx 7: Case Data – Case A

Activity	*Role*	*Problem*	*Solution*	*Resources*	*Observations, remarks*
1. Analyse application documents					
2. Check approval of qualification/accreditation of university	Study assistant	Degree accepted? ("Three-year vs four-year bachelors")	3-year bachelor of commerce (w/o honours) cannot be accepted (see anabin about South African bachelor of commerce degrees)	anabin	Interviewee went directly to anabin to check approval
3. Prepare rejection	Study assistant			Rejection letter template	
4. Reject application	Study assistant			Rejection letter template	

Table-Appx 8: Case Data – Case C

Activity	*Role*	*Problem*	*Solution*	*Resources*	*Observations, remarks*
1. Analyse application documents					
2. Check approval of qualification/accreditation of university	Study assistant	Degree in different area ("Student has degree in a complete different area")	Accept, but needs to do pre-master (see acceptance letter templates)		Such a case will be discussed with the dean
3. Prepare acceptance					
4. Determine tuition fee					
5. Accept application formally	Study assistant			Acceptance letter template (pre-master)	Knowledge about the context is critical

Table-Appx 9: Case Data – Case D

Activity	*Role*	*Problem*	*Solution*	*Resources*	*Observations, remarks*
1. Analyse application documents					
2. Check approval of qualification/accreditation of university	Study assistant	Bachelor degree missing ("Bachelor degree is still missing because the student is still studying")	Can hand in later		
3. Prepare acceptance					
4. Determine tuition fee					
5. Accept application formally	Study assistant			Acceptance letter template (with conditions)	

Appendix-B: Case Data as Graph Visualisation

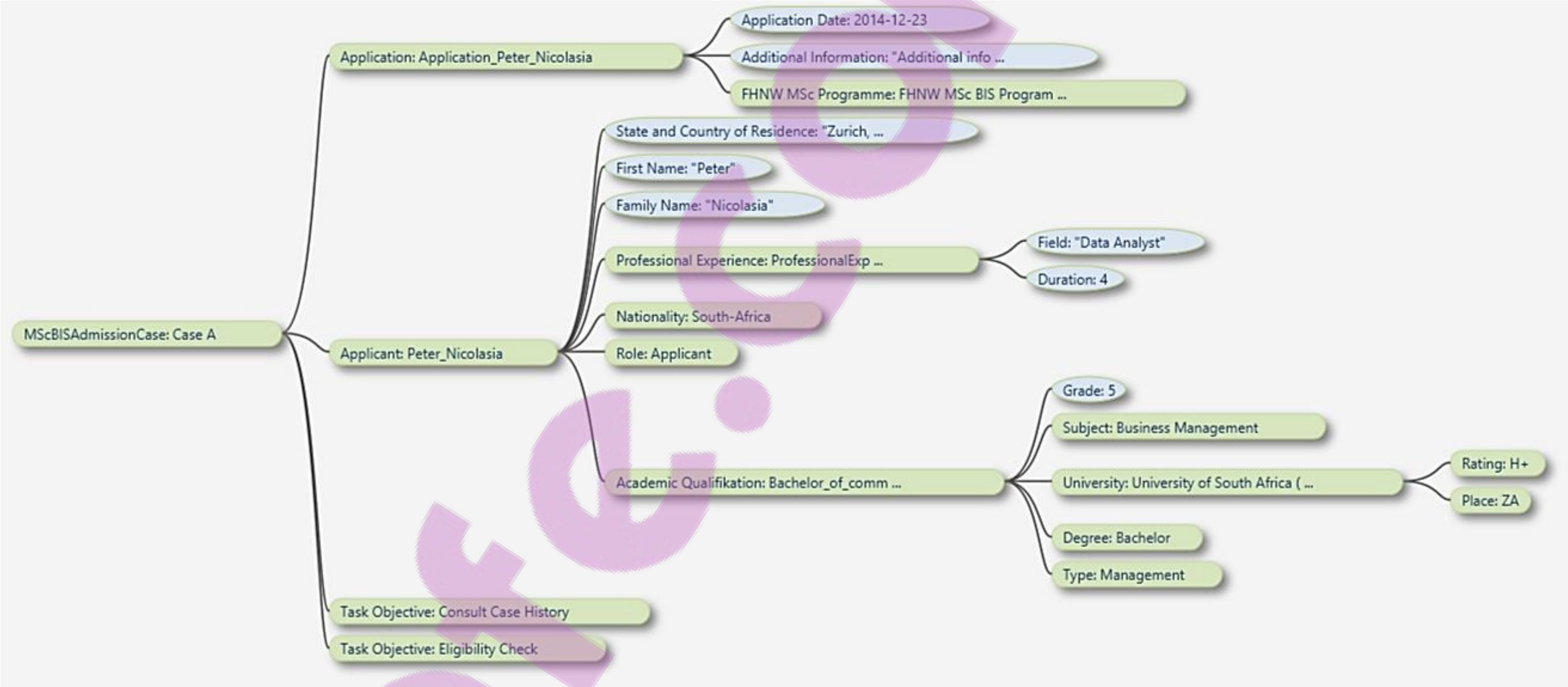

Figure-Appx 1: Case A

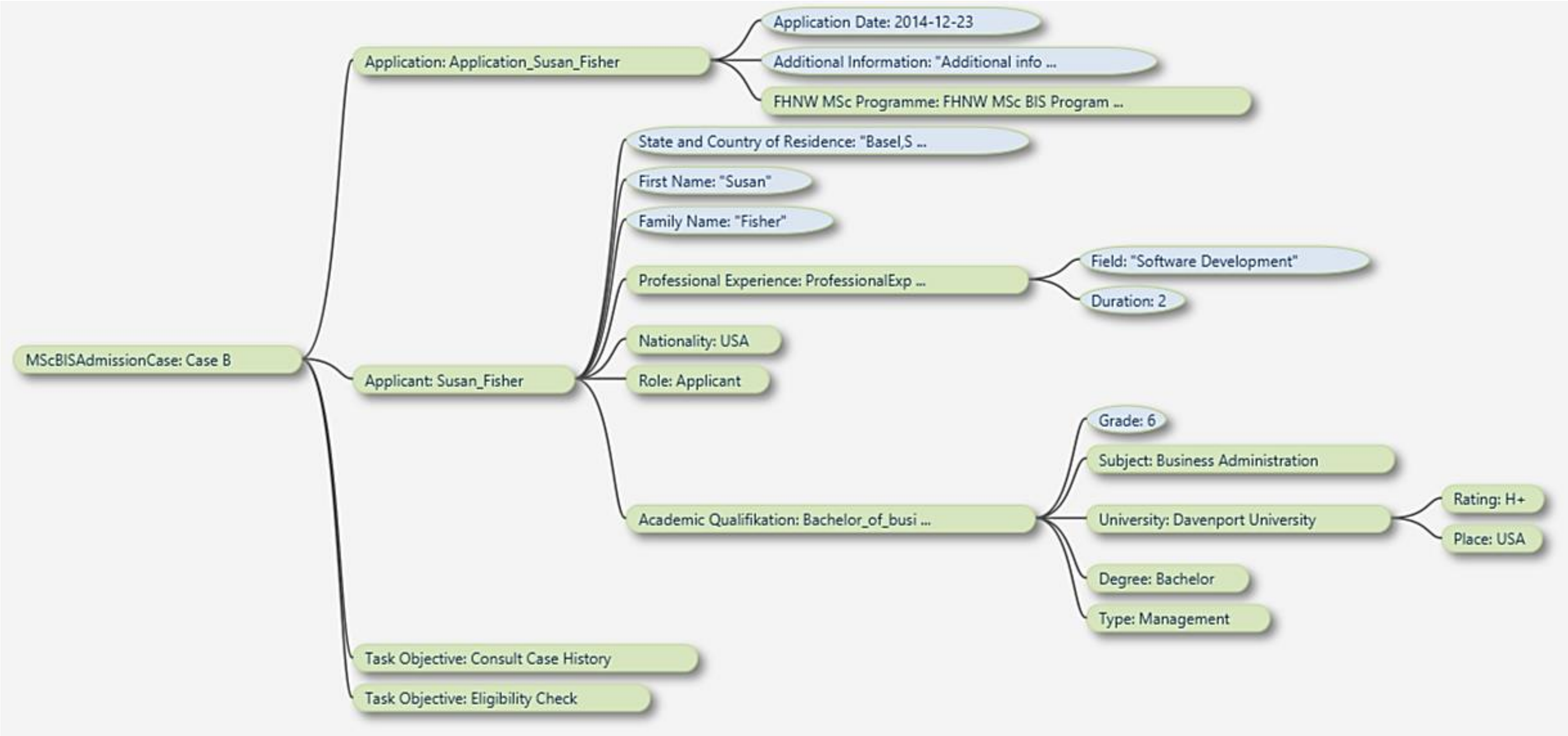

Figure-Appx 2: Case B

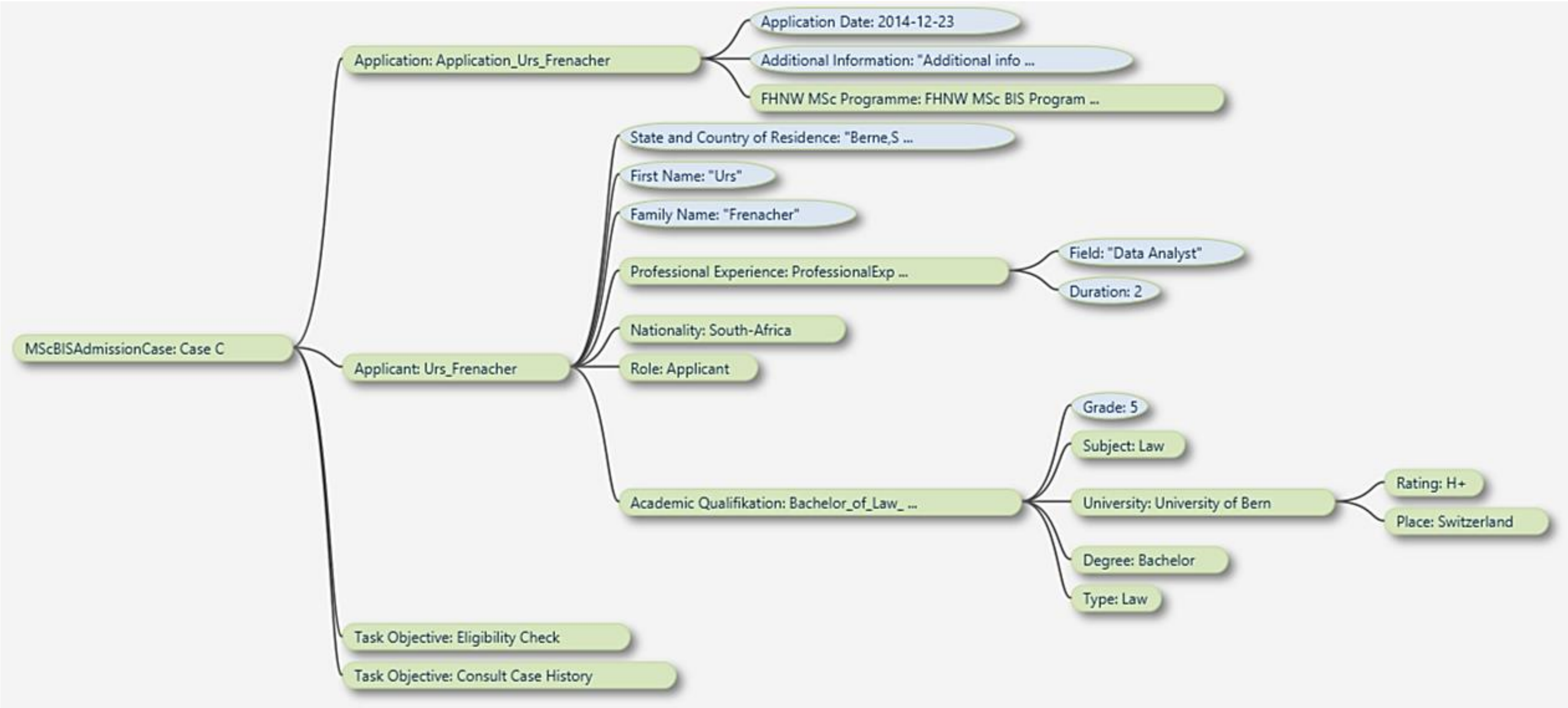

Figure-Appx 3: Case C

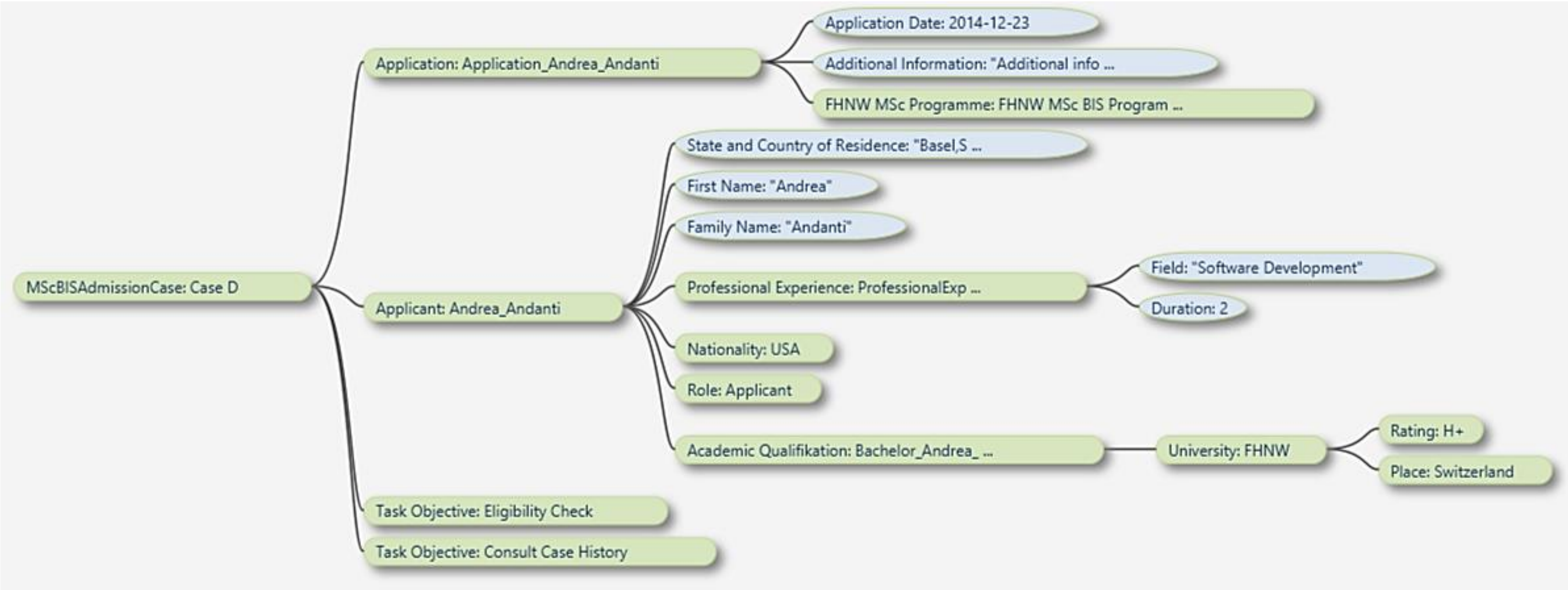

Figure-Appx 4: Case D

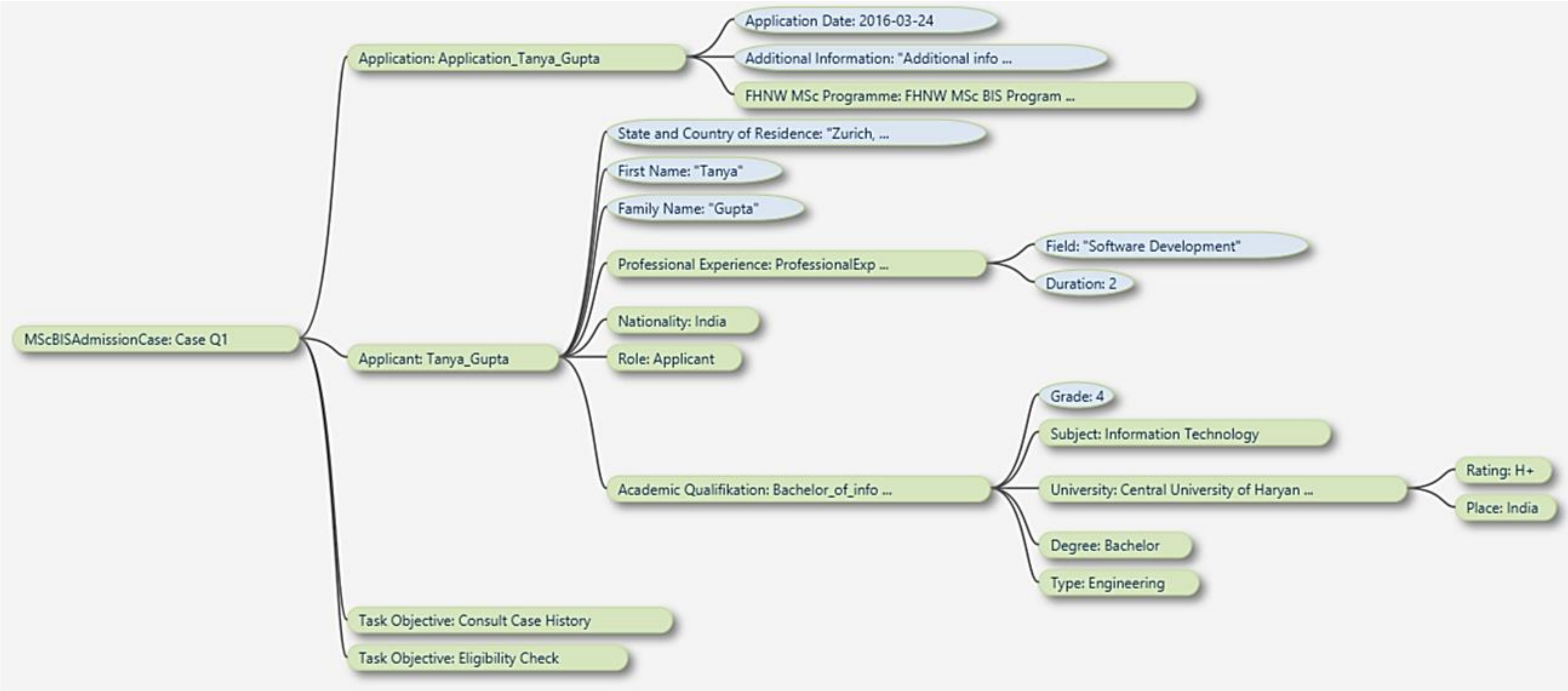

Figure-Appx 5: Query Case Q1

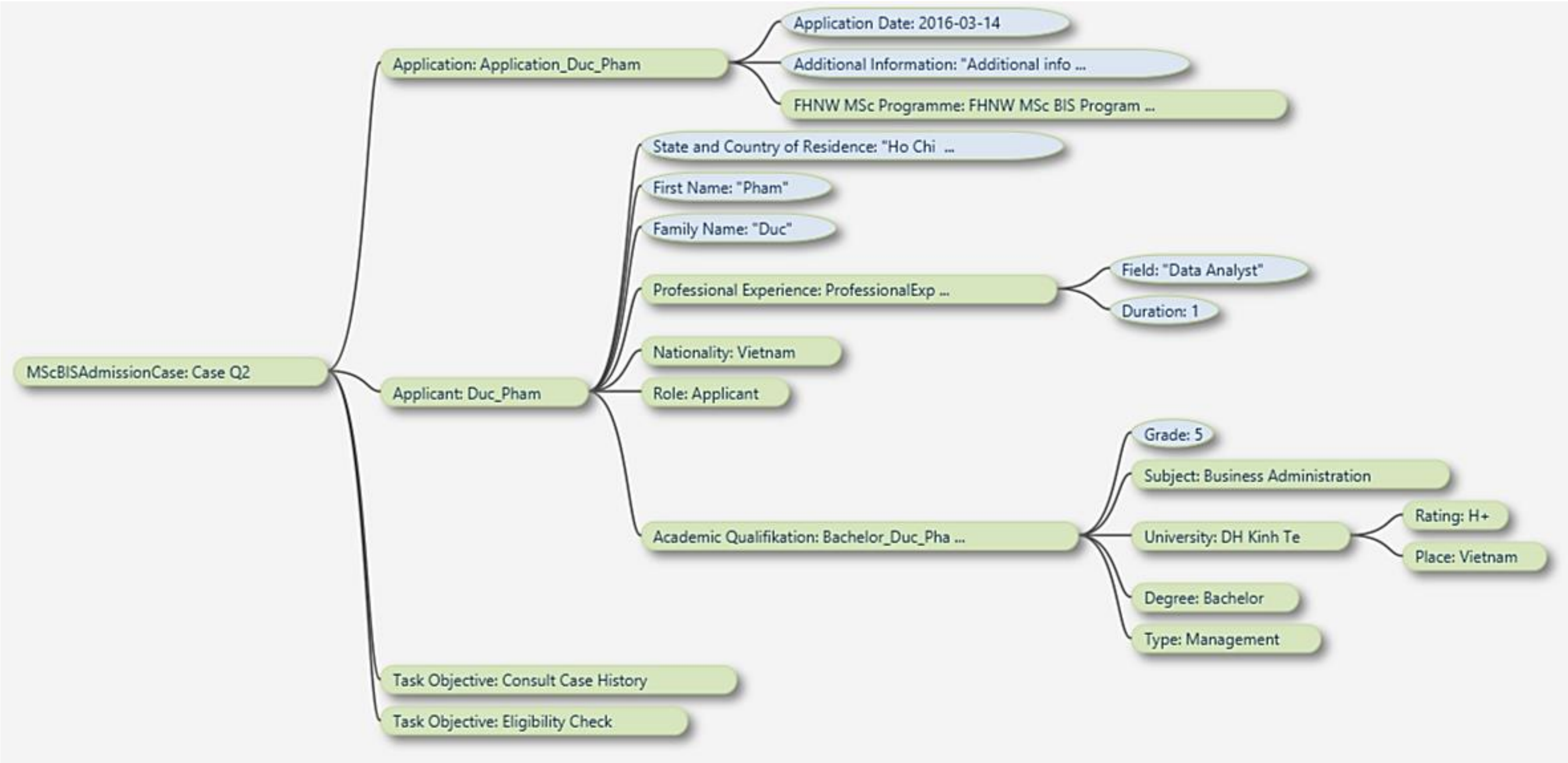

Figure-Appx 6: Query Case Q2

Appendix-C: Excerpt from the Interview Documentation

Development and Evaluation Interview - Case Model and Application Scenario

Admission Process at the University of Applied Sciences Northwestern Switzerland (FHNW)

Aims of the interviews

1. Evaluate the application scenario and confirm process.
2. Elicitate sub-activities in "validate eligibility" activity.
3. Identify the information need in "validate eligibility" activity.
4. Identify information gap concerning data, old cases and procedural knowledge using concrete case(s)
5. Evaluate and discuss experience – Case Content
6. Evaluate and discuss experience – Case Characterisation

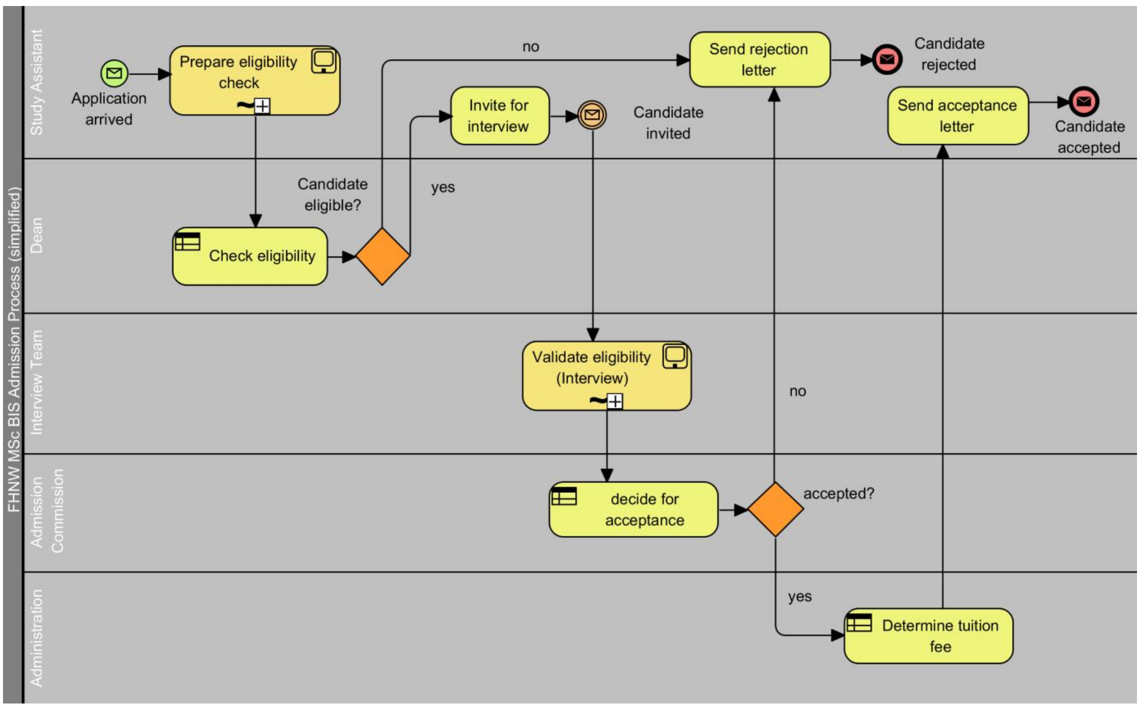

Figure-Appx 7: MSc BIS Admission Process as presented

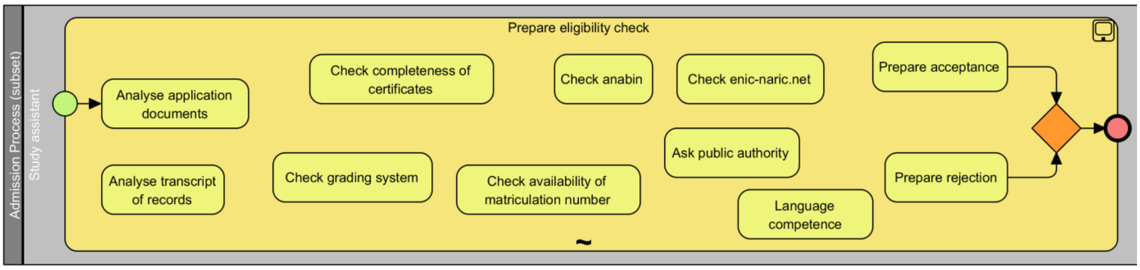

Figure-Appx 8: MSc BIS Admission Sub- Process as presented

MSc BIS Admission Case Characterisation as presented

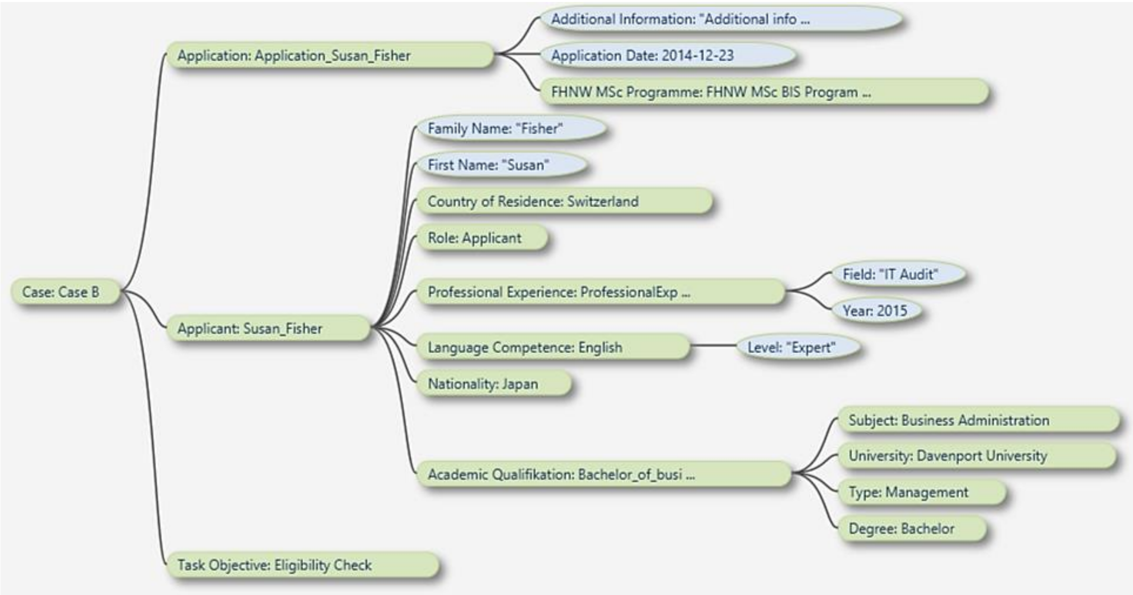

Figure-Appx 9: MSc BIS Admission Case Characterisation as presented

Questionnaire Block 1: Admission Process

1. Could you re-phrase the admission process?
2. Can you confirm the drawn process? Is something missing?
3. In your opinion - how will it be executed?
    a. How do you use the case files?
    b. Do you get certain information electronically?
    c. Where is the data stored?
4. Could you explain certain activities you know?
    a. How does the work happen?
    b. Who collaborate with whom?

Questionnaire Block 2: New Approach

1. What could be improved in general?
2. What should a new approach/system support (aspects, etc.)?
3. What should be the basic functionality?
4. How should a new system be embedded into the working environment?
5. How should the process logic be supported?
    a. In collaboration with a workflow system?
6. Which information elements should be captured/considered?
7. Which information sources should be considered?
8. Does it make sense to implement a task cantered approach?
    a. Would the tasks be managed somehow?
9. Does it make sense to implement a case based approach?
    a. Should it be possible to retrieve cases?
    b. Should it be possible to reuse an element from old cases?
    c. Does it make sense to implement a role based approach?
    d. Should a case based system consider the current situation?
    e. Which elements should a case consist of?
        i. Is there a need for using an enterprise oriented vocabulary?
        ii. Domain ontology/vocabulary?
        iii. Enterprise ontology/vocabulary?
    f. Which information could be stored as cases?
    g. How can case lesson be described?
        i. Free text?
        ii. Models? (BPMN, CMMN, or BPFM similar modelling language)?
10. Inclusion of different viewpoints and concerns needed?

Appendix-D: Similarity Model Evaluation Data

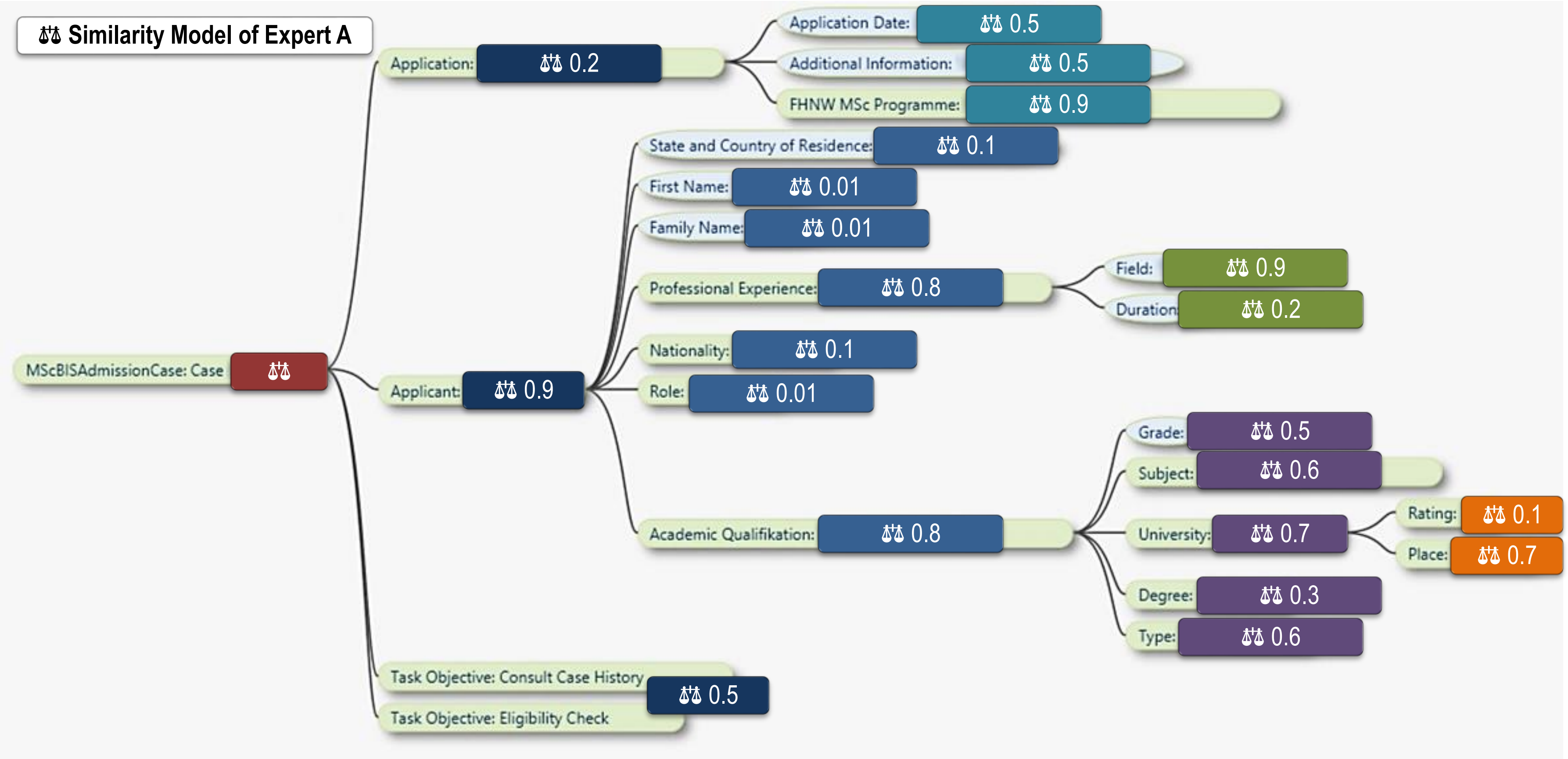

Figure-Appx 10: Similarity Configuration Model of Expert A

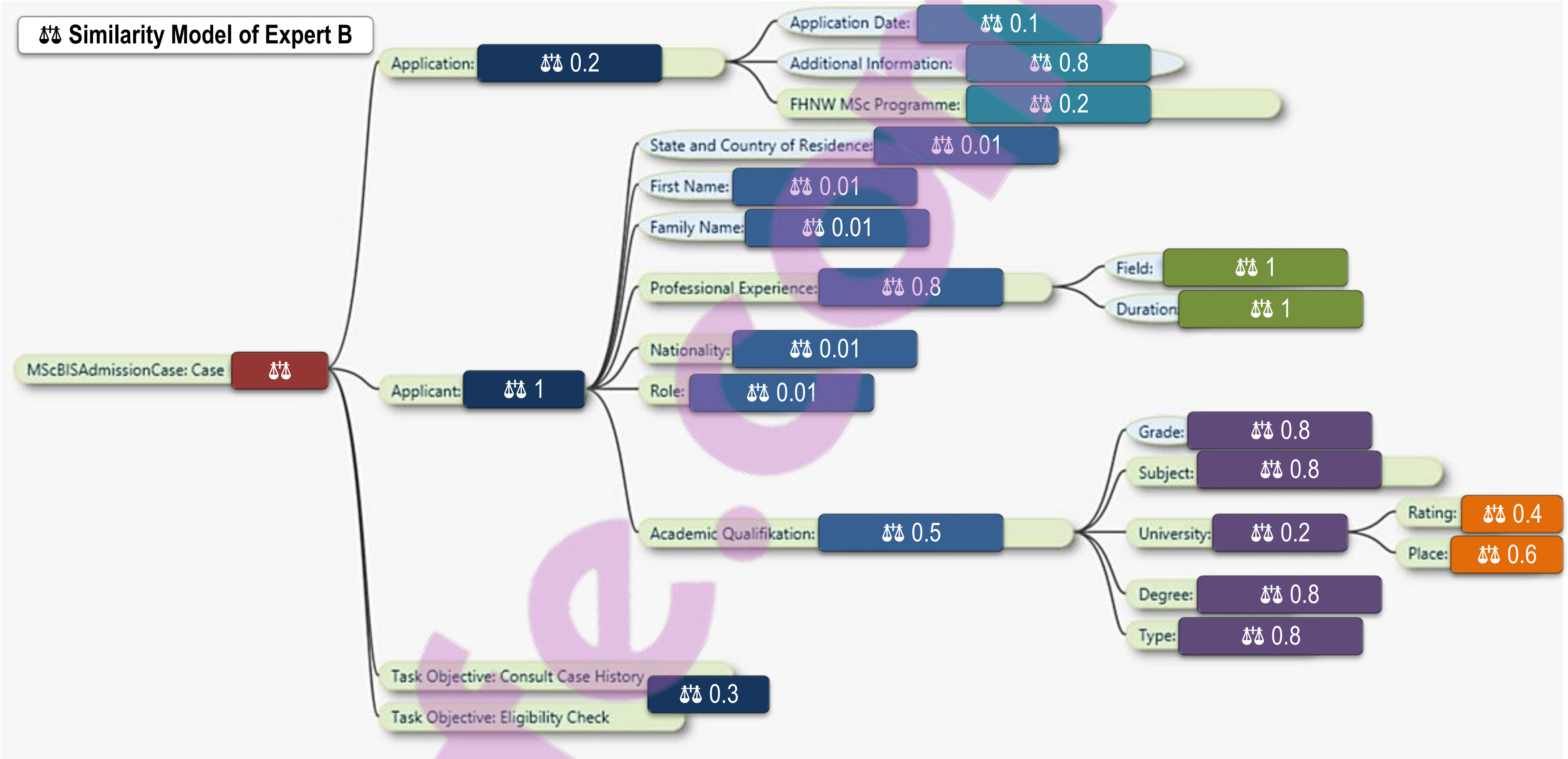

Figure-Appx 11: Similarity Configuration Model of Expert B

Appendix-E: Configuration of Admission Process Scenario

Listing-Appx 1: MSc BIS Ontology

```
# baseURI: http://ikm-group.ch/MScBISOntology
# prefix: mscbis

@prefix archi: <http://ikm-group.ch/archiMEO/archimate#> .
@prefix dcterms: <http://purl.org/dcterms#> .
@prefix elements: <http://purl.org/dc/elements/1.1#> .
@prefix eo: <http://ikm-group.ch/archiMEO/eo#> .
@prefix foaf: <http://xmlns.com/foaf/spec#> .
@prefix mscbis: <http://ikm-group.ch/MScBISOntology#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix top: <http://ikm-group.ch/archiMEO/top#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://ikm-group.ch/MScBISOntology>
  rdf:type owl:Ontology ;
  owl:versionInfo "Created with TopBraid Composer"^^xsd:string ;
.
mscbis:AcademicQualification
  rdf:type owl:Class ;
  rdfs:subClassOf mscbis:MScBISElements ;
.
mscbis:Applicant
  rdfs:comment "A new MSc BIS applicant"^^xsd:string ;
  rdfs:label "Applicant"^^xsd:string ;
  rdf:type archi:BusinessRole;
.
mscbis:Application
  rdf:type owl:Class ;
  rdfs:subClassOf mscbis:MScBISElements ;
.
mscbis:Bachelor
  rdf:type mscbis:Degree ;
  rdfs:label "Bachelor"@en ;
.
mscbis:Business_Administration
  rdf:type mscbis:DegreeSubject ;
  rdfs:label "Business Administration"@en ;
.
mscbis:Information_Technology
  rdf:type mscbis:DegreeSubject ;
  rdfs:label "Information Technology"@en ;
.
mscbis:Business_Information_Systems
  rdf:type mscbis:DegreeSubject ;
  rdfs:label "Business Information Systems"@en ;
.
mscbis:Computer_Science
  rdf:type mscbis:DegreeSubject ;
  rdfs:label "Computer Science"@en ;
.
mscbis:Finance
```

```
  rdf:type mscbis:DegreeSubject ;
  rdfs:label "Finance"@en ;
.
mscbis:Mathematics
  rdf:type mscbis:DegreeSubject ;
  rdfs:label "Mathematics"@en ;
.
mscbis:Economics
  rdf:type mscbis:DegreeSubject ;
  rdfs:label "Economics"@en ;
.
mscbis:Business_Management
  rdf:type mscbis:DegreeSubject ;
  rdfs:label "Business Management"@en ;
.
mscbis:Electrical_Engineering
  rdf:type mscbis:DegreeSubject ;
  rdfs:label "Electrical Engineering"@en ;
.
mscbis:Davenport_University
  rdf:type mscbis:University ;
  mscbis:universityPlace mscbis:UniversityCountry_USA ;
  mscbis:universityRating mscbis:UniversityRating_H_plus ;
  rdfs:label "Davenport University"@en ;
.
mscbis:Central_University_of_Haryana
  rdf:type mscbis:University ;
  mscbis:universityPlace mscbis:UniversityCountry_India ;
  mscbis:universityRating mscbis:UniversityRating_H_plus ;
  rdfs:label "Central University of Haryana"@en ;
.
mscbis:DH_Kinh_Te
  rdf:type mscbis:University ;
  mscbis:universityPlace mscbis:UniversityCountry_Vietnam ;
  mscbis:universityRating mscbis:UniversityRating_H_plus ;
  rdfs:label "DH Kinh Te"@en ;
.
mscbis:Degree
  rdf:type owl:Class ;
  rdfs:subClassOf mscbis:MScBISElements ;
.
mscbis:DegreeSubject
  rdf:type owl:Class ;
  rdfs:subClassOf mscbis:MScBISElements ;
.
mscbis:DegreeType
  rdf:type owl:Class ;
  rdfs:subClassOf mscbis:MScBISElements ;
.
mscbis:FHNW
  rdf:type mscbis:University ;
  mscbis:universityPlace mscbis:UniversityCountry_Switzerland ;
  mscbis:universityRating mscbis:UniversityRating_H_plus ;
  rdfs:label "FHNW"@en ;
.
mscbis:InterviewTeam
  rdf:type eo:Employee ;
  rdfs:label "MSc BIS Interview Team"@en ;
.
mscbis:Law
  rdf:type mscbis:DegreeSubject ;
  rdfs:label "Law"@en ;
.
```

```
mscbis:Law_Type
  rdf:type mscbis:DegreeType ;
  rdfs:label "Law"@en ;
.
mscbis:MScBISElements
  rdf:type owl:Class ;
  rdfs:subClassOf owl:Thing ;
.
mscbis:MScBISProgramme
  rdf:type mscbis:MScProgramme ;
  rdfs:label "FHNW MSc BIS Programme"@en ;
.
mscbis:MScProgramme
  rdf:type owl:Class ;
  rdfs:subClassOf mscbis:MScBISElements ;
.
mscbis:Management
  rdf:type mscbis:DegreeType ;
  rdfs:label "Management"@en ;
.
mscbis:Engineering
  rdf:type mscbis:DegreeType ;
  rdfs:label "Engineering"@en ;
.
mscbis:Nationality
  rdf:type owl:Class ;
  rdfs:subClassOf mscbis:MScBISElements ;
  rdfs:subClassOf top:Country ;
.
mscbis:ProfessionalExperience
  rdf:type owl:Class ;
  rdfs:subClassOf mscbis:MScBISElements ;
.
mscbis:Residence
  rdf:type owl:Class ;
  rdfs:subClassOf mscbis:MScBISElements ;
  rdfs:subClassOf top:PhysicalLocation ;
.
mscbis:Residence_Switzerland
  rdf:type mscbis:Residence ;
  rdfs:label "Switzerland"@en ;
.
mscbis:StudyAssistant
  rdf:type eo:Employee ;
  rdfs:label "MSc BIS Study Assistant"@en ;
.
mscbis:UB
  rdf:type mscbis:University ;
  mscbis:universityPlace mscbis:UniversityCountry_Switzerland ;
  mscbis:universityRating mscbis:UniversityRating_H_plus ;
  rdfs:label "University of Bern"@en ;
.
mscbis:UNISA
  rdf:type mscbis:University ;
  mscbis:universityPlace mscbis:UniversityCountry_ZA ;
  mscbis:universityRating mscbis:UniversityRating_H_plus ;
  rdfs:label "University of South Africa (UNISA)"@en ;
.
mscbis:University
  rdf:type owl:Class ;
  rdfs:subClassOf mscbis:MScBISElements ;
.
mscbis:UniversityCountry
```

```
  rdf:type owl:Class ;
  rdfs:label "University Country"@en ;
  rdfs:subClassOf mscbis:MScBISElements ;
.
mscbis:UniversityCountry_Switzerland
  rdf:type mscbis:UniversityCountry ;
  rdfs:label "Switzerland"@en ;
.
mscbis:UniversityCountry_USA
  rdf:type mscbis:UniversityCountry ;
  rdfs:label "USA"@en ;
.
mscbis:UniversityCountry_India
  rdf:type mscbis:UniversityCountry ;
  rdfs:label "India"@en ;
.
mscbis:UniversityCountry_Vietnam
  rdf:type mscbis:UniversityCountry ;
  rdfs:label "Vietnam"@en ;
.
mscbis:UniversityCountry_ZA
  rdf:type mscbis:UniversityCountry ;
  rdfs:label "ZA"@en ;
.
mscbis:UniversityRating
  rdf:type owl:Class ;
  rdfs:label "Rating"@en ;
  rdfs:subClassOf mscbis:MScBISElements ;
.
mscbis:UniversityRating_H_plus
  rdf:type mscbis:UniversityRating ;
  rdfs:label "H+"@en ;
.
mscbis:UniversityRating_H_minusplus
  rdf:type mscbis:UniversityRating ;
  rdfs:label "H-/+"@en ;
.
mscbis:UniversityRating_H_minus
  rdf:type mscbis:UniversityRating ;
  rdfs:label "H-"@en ;
.
mscbis:academicQualificationConsistsOfDegree
  rdf:type owl:ObjectProperty ;
  rdfs:domain mscbis:AcademicQualification ;
  rdfs:label "Degree"@en ;
  rdfs:range mscbis:Degree ;
.
mscbis:academicQualificationConsistsOfDegreeSubject
  rdf:type owl:ObjectProperty ;
  rdfs:domain mscbis:AcademicQualification ;
  rdfs:label "Subject"@en ;
  rdfs:range mscbis:DegreeSubject ;
.
mscbis:academicQualificationConsistsOfDegreeType
  rdf:type owl:ObjectProperty ;
  rdfs:domain mscbis:AcademicQualification ;
  rdfs:label "Type"@en ;
  rdfs:range mscbis:DegreeType ;
.
mscbis:academicQualificationConsistsOfFinalDegreeUniversity
  rdf:type owl:ObjectProperty ;
  rdfs:domain mscbis:AcademicQualification ;
  rdfs:label "University"@en ;
```

```
  rdfs:range mscbis:University ;
.
mscbis:academicQualificationConsistsOfGrade
  rdf:type owl:DatatypeProperty ;
  rdfs:domain mscbis:AcademicQualification ;
  rdfs:label "Grade"@en ;
  rdfs:range xsd:integer ;
.
mscbis:applicationHasAdditionalInformation
  rdf:type owl:DatatypeProperty ;
  rdfs:domain mscbis:Application ;
  rdfs:label "Additional Information"@en ;
  rdfs:range xsd:string ;
.
mscbis:applicationHasDate
  rdf:type owl:DatatypeProperty ;
  rdfs:domain mscbis:Application ;
  rdfs:label "Application Date"@en ;
  rdfs:range xsd:date ;
.
mscbis:applicationToProgramme
  rdf:type owl:ObjectProperty ;
  rdfs:domain mscbis:Application ;
  rdfs:label "FHNW MSc Programme"@en ;
  rdfs:range mscbis:MScProgramme ;
.
mscbis:documentHasFilePath
  rdf:type owl:DatatypeProperty ;
  rdfs:domain foaf:Document ;
  rdfs:range xsd:string ;
.
mscbis:personHasAcademicQualifikation
  rdf:type owl:ObjectProperty ;
  rdfs:domain eo:Person ;
  rdfs:label "Academic Qualifikation"@en ;
  rdfs:range mscbis:AcademicQualification ;
.
mscbis:personHasBusinessRole
  rdf:type owl:ObjectProperty ;
  rdfs:domain eo:Person ;
  rdfs:label "Role"@en ;
  rdfs:range archi:BusinessRole ;
.
mscbis:personHasCountryOfResidence
  rdf:type owl:DatatypeProperty ;
  rdfs:domain eo:Person ;
  rdfs:label "State and Country of Residence"@en ;
  rdfs:range xsd:string ;
.
mscbis:personHasFamilyName
  rdf:type owl:DatatypeProperty ;
  rdfs:domain eo:Person ;
  rdfs:label "Family Name"@en ;
  rdfs:range xsd:string ;
.
mscbis:personHasFirstName
  rdf:type owl:DatatypeProperty ;
  rdfs:domain eo:Person ;
  rdfs:label "First Name"@en ;
  rdfs:range xsd:string ;
.
mscbis:personHasNationality
  rdf:type owl:ObjectProperty ;
```

```
  rdfs:domain eo:Person ;
  rdfs:label "Nationality"@en ;
  rdfs:range top:Country ;
.
mscbis:personHasProfessionalExperience
  rdf:type owl:ObjectProperty ;
  rdfs:domain eo:Person ;
  rdfs:label "Professional Experience"@en ;
  rdfs:range mscbis:ProfessionalExperience ;
.
mscbis:professionalExperienceField
  rdf:type owl:DatatypeProperty ;
  rdfs:domain mscbis:ProfessionalExperience ;
  rdfs:label "Field"@en ;
  rdfs:range xsd:string ;
.
mscbis:professionalExperienceFieldDuration
  rdf:type owl:DatatypeProperty ;
  rdfs:domain mscbis:ProfessionalExperience ;
  rdfs:label "Duration"@en ;
  rdfs:range xsd:integer ;
.
mscbis:professionalExperienceFieldYear
  rdf:type owl:DatatypeProperty ;
  rdfs:domain mscbis:ProfessionalExperience ;
  rdfs:label "Year"@en ;
  rdfs:range xsd:integer ;
.
mscbis:universityPlace
  rdf:type owl:ObjectProperty ;
  rdfs:domain mscbis:University ;
  rdfs:label "Place"@en ;
  rdfs:range top:PhysicalLocation ;
  rdfs:range mscbis:UniversityCountry ;
.
mscbis:universityRating
  rdf:type owl:ObjectProperty ;
  rdfs:domain mscbis:University ;
  rdfs:label "Rating"@en ;
  rdfs:range mscbis:UniversityRating ;
.
eo:Person
  rdfs:subClassOf mscbis:MScBISElements ;
.
```

Listing-Appx 2: MSc BIS Case Model

```
# baseURI: http://ikm-group.ch/MScBISOntologyCaseModel
# imports: http://ikm-group.ch/MScBISOntology
# imports: http://ikm-group.ch/ProcessExecutionOntology
# imports: http://ikm-group.ch/cbr

@prefix archi: <http://ikm-group.ch/archiMEO/archimate#> .
@prefix cbr: <http://ikm-group.ch/cbr#> .
@prefix dcterms: <http://purl.org/dcterms#> .
@prefix elements: <http://purl.org/dc/elements/1.1#> .
@prefix eo: <http://ikm-group.ch/archiMEO/eo#> .
@prefix foaf: <http://xmlns.com/foaf/spec#> .
@prefix mscbis: <http://ikm-group.ch/MScBISOntology#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix po: <http://ikm-group.ch/ProcessExecutionOntology#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sim: <http://ikm-group.ch/similarity#> .
@prefix top: <http://ikm-group.ch/archiMEO/top#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

mscbis:Applicant
  rdf:type cbr:Role ;
.
mscbis:MScBISAdmissionCase
  rdf:type owl:Class ;
  rdfs:subClassOf mscbis:MScBISCase ;
.
mscbis:MScBISCase
  rdf:type owl:Class ;
  rdfs:subClassOf cbr:Case ;
.
mscbis:TaskObjective_Eligibility_Check
  rdf:type po:TaskObjective ;
  rdfs:label "Eligibility Check"@en ;
.
mscbis:TaskObjective_Eligibility_History
  rdf:type po:TaskObjective ;
  rdfs:label "Consult Case History"@en ;
.
mscbis:caseFileItemRepresentedBy
  rdf:type owl:ObjectProperty ;
  rdfs:domain cbr:CaseItem ;
  rdfs:range eo:EnterpriseObject ;
.
mscbis:mscBISAdmissionCaseIsCharacterizedByApplication
  rdf:type owl:ObjectProperty ;
  rdfs:domain mscbis:MScBISAdmissionCase ;
  rdfs:label "Application"@en ;
  rdfs:range mscbis:Application ;
.
mscbis:mscBISAdmissionCaseIsCharacterizedByPerson
  rdf:type owl:ObjectProperty ;
  rdfs:domain mscbis:MScBISAdmissionCase ;
  rdfs:label "Applicant"@en ;
  rdfs:range eo:Person ;
.
mscbis:mscBISAdmissionCaseIsCharacterizedByTaskObjective
  rdf:type owl:ObjectProperty ;
  rdfs:domain mscbis:MScBISAdmissionCase ;
  rdfs:label "Task Objective"@en ;
  rdfs:range po:TaskObjective ;
```

```
.
mscbis:universityPlace
  rdf:type owl:ObjectProperty ;
  rdfs:domain mscbis:University ;
  rdfs:range top:PhysicalLocation ;
.
mscbis:universityRating
  rdf:type owl:ObjectProperty ;
  rdfs:domain mscbis:University ;
  rdfs:label "Rating"@en ;
  rdfs:range mscbis:UniversityRating ;
.
<http://ikm-group.ch/MScBISOntologyCaseModel>
  rdf:type owl:Ontology ;
  owl:imports <http://ikm-group.ch/MScBISOntology> ;
  owl:imports <http://ikm-group.ch/ProcessExecutionOntology> ;
  owl:imports <http://ikm-group.ch/cbr> ;
  owl:versionInfo "Created with TopBraid Composer" ;
.
po:TaskObjective
  rdfs:subClassOf mscbis:MScBISElements ;
.
archi:BusinessRole
  rdfs:subClassOf cbr:Role ;
.
cbr:AdmissionConcern
  rdf:type cbr:Concern ;
  cbr:concernsBelongToRole mscbis:StudyAssistant ;
  rdfs:label "MSc BIS Admission"@en ;
.
cbr:EligibilityConcern
  rdf:type cbr:Concern ;
  cbr:concernsBelongToRole mscbis:InterviewTeam ;
  rdfs:label "MSc BIS Interview Team"@en ;
.
```

Listing-Appx 3: MSc BIS Case Similarity Model

```
# baseURI: http://ikm-group.ch/MScBISOntologyCaseSimilarityModel
# imports: http://ikm-group.ch/MScBISOntology
# imports: http://ikm-group.ch/MScBISOntologyCaseModel
# imports: http://ikm-group.ch/ProcessExecutionOntology
# imports: http://ikm-group.ch/cbr

@prefix archi: <http://ikm-group.ch/archiMEO/archimate#> .
@prefix cbr: <http://ikm-group.ch/cbr#> .
@prefix dcterms: <http://purl.org/dcterms#> .
@prefix elements: <http://purl.org/dc/elements/1.1#> .
@prefix eo: <http://ikm-group.ch/archiMEO/eo#> .
@prefix foaf: <http://xmlns.com/foaf/spec#> .
@prefix mscbis: <http://ikm-group.ch/MScBISOntology#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix po: <http://ikm-group.ch/ProcessExecutionOntology#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sim: <http://ikm-group.ch/similarity#> .
@prefix top: <http://ikm-group.ch/archiMEO/top#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

mscbis:AdmissionProcessView
  rdf:type cbr:CaseView ;
  cbr:caseViewAddressesConcerns cbr:AdmissionConcern ;
.
mscbis:Degree
  sim:similarity mscbis:simAP_Degree_AdmissionProcessView ;
  sim:similarity mscbis:simAP_Degree_EligibilityView ;
.
mscbis:DegreeSubject
  sim:similarity mscbis:simAP_DegreeSubject_AdmissionProcessView ;
  sim:similarity mscbis:simAP_DegreeSubject_EligibilityView ;
.
mscbis:DegreeType
  sim:similarity mscbis:simAP_DegreeType_AdmissionProcessView ;
  sim:similarity mscbis:simAP_DegreeType_EligibilityView ;
.
mscbis:EligibilityView
  rdf:type cbr:CaseView ;
  cbr:caseViewAddressesConcerns cbr:EligibilityConcern ;
.
mscbis:MScBISAdmissionCase
  sim:similarity mscbis:simCS_MScBISCase_AdmissionProcessView ;
  sim:similarity mscbis:simCS_MScBISCase_EligibilityView ;
.
mscbis:MScProgramme
  sim:similarity mscbis:simAP_MScProgramme_AdmissionProcessView ;
  sim:similarity mscbis:simAP_MScProgramme_EligibilityView ;
.
mscbis:Nationality
  sim:similarity mscbis:simAP_Nationality_AdmissionProcessView ;
  sim:similarity mscbis:simAP_Nationality_EligibilityView ;
.
mscbis:Residence
  sim:similarity mscbis:simAP_Residence_AdmissionProcessView ;
  sim:similarity mscbis:simAP_Residence_EligibilityView ;
.
mscbis:University
  sim:similarity mscbis:simAP_University_AdmissionProcessView ;
  sim:similarity mscbis:simAP_University_EligibilityView ;
.
```

```
mscbis:UniversityCountry
  sim:similarity mscbis:simAP_UniversityCountry_AdmissionProcessView ;
  sim:similarity mscbis:simAP_UniversityCountry_EligibilityView ;
.
mscbis:UniversityRating
  sim:similarity mscbis:simAP_UniversityRating_AdmissionProcessView ;
  sim:similarity mscbis:simAP_UniversityRating_EligibilityView ;
.
mscbis:academicQualificationConsistsOfDegree
  sim:similarity
mscbis:simOP_academicQualificationConsistsOfDegree_AdmissionProcessVie
w ;
  sim:similarity
mscbis:simOP_academicQualificationConsistsOfDegree_EligibilityView ;
.
mscbis:academicQualificationConsistsOfDegreeSubject
  sim:similarity
mscbis:simOP_academicQualificationConsistsOfDegreeSubject_AdmissionPro
cessView ;
  sim:similarity
mscbis:simOP_academicQualificationConsistsOfDegreeSubject_EligibilityV
iew ;
.
mscbis:academicQualificationConsistsOfDegreeType
  sim:similarity
mscbis:simOP_academicQualificationConsistsOfDegreeType_AdmissionProces
sView ;
  sim:similarity
mscbis:simOP_academicQualificationConsistsOfDegreeType_EligibilityView
;
.
mscbis:academicQualificationConsistsOfFinalDegreeUniversity
  sim:similarity
mscbis:simOP_academicQualificationConsistsOfFinalDegreeUniversity_Admi
ssionProcessView ;
  sim:similarity
mscbis:simOP_academicQualificationConsistsOfFinalDegreeUniversity_Elig
ibilityView ;
.
mscbis:academicQualificationConsistsOfGrade
  sim:similarity mscbis:simDP_grade_AdmissionProcessView ;
  sim:similarity mscbis:simDP_grade_EligibilityView ;
.
mscbis:applicationHasAdditionalInformation
  sim:similarity
mscbis:simDP_applicationHasAdditionalInformation_AdmissionProcessView
;
  sim:similarity
mscbis:simDP_applicationHasAdditionalInformation_EligibilityView ;
.
mscbis:applicationHasDate
  sim:similarity mscbis:simDP_applicationHasDate_AdmissionProcessView
;
  sim:similarity mscbis:simDP_applicationHasDate_EligibilityView ;
.
mscbis:applicationToProgramme
  sim:similarity
mscbis:simOP_applicationToProgramme_AdmissionProcessView ;
  sim:similarity mscbis:simOP_applicationToProgramme_EligibilityView ;
.
mscbis:mscBISAdmissionCaseIsCharacterizedByApplication
```

```
  sim:similarity
mscbis:simOP_mscBISAdmissionCaseIsCharacterizedByApplication_Admission
ProcessView ;
  sim:similarity
mscbis:simOP_mscBISAdmissionCaseIsCharacterizedByApplication_Eligibili
tyView ;
.
mscbis:mscBISAdmissionCaseIsCharacterizedByPerson
  sim:similarity
mscbis:simOP_mscBISAdmissionCaseIsCharacterizedByPerson_AdmissionProce
ssView ;
  sim:similarity
mscbis:simOP_mscBISAdmissionCaseIsCharacterizedByPerson_EligibilityVie
w ;
.
mscbis:mscBISAdmissionCaseIsCharacterizedByTaskObjective
  sim:similarity
mscbis:simOP_mscBISAdmissionCaseIsCharacterizedByTaskObjective_Admissi
onProcessView ;
  sim:similarity
mscbis:simOP_mscBISAdmissionCaseIsCharacterizedByTaskObjective_Eligibi
lityView ;
.
mscbis:personHasAcademicQualifikation
  sim:similarity
mscbis:simOP_personHasAcademicQualifikation_AdmissionProcessView ;
  sim:similarity
mscbis:simOP_personHasAcademicQualifikation_EligibilityView ;
.
mscbis:personHasBusinessRole
  sim:similarity
mscbis:simOP_personHasBusinessRole_AdmissionProcessView ;
  sim:similarity mscbis:simOP_personHasBusinessRole_EligibilityView ;
.
mscbis:personHasCountryOfResidence
  sim:similarity
mscbis:simDP_personHasCountryOfResidence_AdmissionProcessView ;
  sim:similarity
mscbis:simDP_personHasCountryOfResidence_EligibilityView ;
.
mscbis:personHasFamilyName
  sim:similarity mscbis:simDP_personHasFamilyName_AdmissionProcessView
;
  sim:similarity mscbis:simDP_personHasFamilyName_EligibilityView ;
.
mscbis:personHasFirstName
  sim:similarity mscbis:simDP_personHasFirstName_AdmissionProcessView
;
  sim:similarity mscbis:simDP_personHasFirstName_EligibilityView ;
.
mscbis:personHasNationality
  sim:similarity
mscbis:simOP_personHasNationality_AdmissionProcessView ;
  sim:similarity mscbis:simOP_personHasNationality_EligibilityView ;
.
mscbis:personHasProfessionalExperience
  sim:similarity
mscbis:simOP_personHasProfessionalExperience_AdmissionProcessView ;
  sim:similarity
mscbis:simOP_personHasProfessionalExperience_EligibilityView ;
.
mscbis:professionalExperienceField
```

```
  sim:similarity
mscbis:simDP_professionalExperienceField_AdmissionProcessView ;
  sim:similarity
mscbis:simDP_professionalExperienceField_EligibilityView ;
.
mscbis:professionalExperienceFieldDuration
  sim:similarity
mscbis:simDP_professionalExperienceFieldDuration_AdmissionProcessView
;
  sim:similarity
mscbis:simDP_professionalExperienceFieldDuration_EligibilityView ;
.
mscbis:simAP_Application_AdmissionProcessView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_Application_EligibilityView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_BusinessRole_AdmissionProcessView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_BusinessRole_EligibilityView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_Country_AdmissionProcessView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_Country_EligibilityView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_DegreeSubject_AdmissionProcessView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
```

```
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_DegreeSubject_EligibilityView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_DegreeType_AdmissionProcessView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_DegreeType_EligibilityView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_Degree_AdmissionProcessView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_Degree_EligibilityView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_MScProgramme_AdmissionProcessView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:equals ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_MScProgramme_EligibilityView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:equals ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_Nationality_AdmissionProcessView
  rdf:type sim:AnnotationPropertySimilarity ;
```

```
    sim:annotationLanguage "en" ;
    sim:annotationProperty rdfs:label ;
    sim:belongsToCaseView mscbis:AdmissionProcessView ;
    sim:localSimilarityFunction sim:levenshtein ;
    sim:weight "1"^^xsd:float ;
  .
  mscbis:simAP_Nationality_EligibilityView
    rdf:type sim:AnnotationPropertySimilarity ;
    sim:annotationLanguage "en" ;
    sim:annotationProperty rdfs:label ;
    sim:belongsToCaseView mscbis:EligibilityView ;
    sim:localSimilarityFunction sim:levenshtein ;
    sim:weight "1"^^xsd:float ;
  .
  mscbis:simAP_Residence_AdmissionProcessView
    rdf:type sim:AnnotationPropertySimilarity ;
    sim:annotationLanguage "en" ;
    sim:annotationProperty rdfs:label ;
    sim:belongsToCaseView mscbis:AdmissionProcessView ;
    sim:localSimilarityFunction sim:levenshtein ;
    sim:weight "1"^^xsd:float ;
  .
  mscbis:simAP_Residence_EligibilityView
    rdf:type sim:AnnotationPropertySimilarity ;
    sim:annotationLanguage "en" ;
    sim:annotationProperty rdfs:label ;
    sim:belongsToCaseView mscbis:EligibilityView ;
    sim:localSimilarityFunction sim:levenshtein ;
    sim:weight "1"^^xsd:float ;
  .
  mscbis:simAP_TaskObjective_AdmissionProcessView
    rdf:type sim:AnnotationPropertySimilarity ;
    sim:annotationLanguage "en" ;
    sim:annotationProperty rdfs:label ;
    sim:belongsToCaseView mscbis:AdmissionProcessView ;
    sim:localSimilarityFunction sim:equals ;
    sim:weight "1"^^xsd:float ;
  .
  mscbis:simAP_TaskObjective_EligibilityView
    rdf:type sim:AnnotationPropertySimilarity ;
    sim:annotationLanguage "en" ;
    sim:annotationProperty rdfs:label ;
    sim:belongsToCaseView mscbis:EligibilityView ;
    sim:localSimilarityFunction sim:equals ;
    sim:weight "1"^^xsd:float ;
  .
  mscbis:simAP_UniversityCountry_AdmissionProcessView
    rdf:type sim:AnnotationPropertySimilarity ;
    sim:annotationLanguage "en" ;
    sim:annotationProperty rdfs:label ;
    sim:belongsToCaseView mscbis:AdmissionProcessView ;
    sim:localSimilarityFunction sim:levenshtein ;
    rdfs:label "Country"@en ;
  .
  mscbis:simAP_UniversityCountry_EligibilityView
    rdf:type sim:AnnotationPropertySimilarity ;
    sim:annotationLanguage "en" ;
    sim:annotationProperty rdfs:label ;
    sim:belongsToCaseView mscbis:EligibilityView ;
    sim:localSimilarityFunction sim:levenshtein ;
    rdfs:label "Country"@en ;
  .
  mscbis:simAP_UniversityRating_AdmissionProcessView
```

```
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "2"^^xsd:float ;
.
mscbis:simAP_UniversityRating_EligibilityView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "2"^^xsd:float ;
.
mscbis:simAP_University_AdmissionProcessView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_University_EligibilityView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_mscBISTaskObjective_AdmissionProcessView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:equals ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simAP_mscBISTaskObjective_EligibilityView
  rdf:type sim:AnnotationPropertySimilarity ;
  sim:annotationLanguage "en" ;
  sim:annotationProperty rdfs:label ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:equals ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simCS_MScBISCase_AdmissionProcessView
  rdf:type sim:RootCaseClassSimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
.
mscbis:simCS_MScBISCase_EligibilityView
  rdf:type sim:RootCaseClassSimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
.
mscbis:simDP_applicationHasAdditionalInformation_AdmissionProcessView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:softTFIDFJaroWinkler ;
  sim:weight "1"^^xsd:float ;
.
```

```
mscbis:simDP_applicationHasAdditionalInformation_EligibilityView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:softTFIDFJaroWinkler ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simDP_applicationHasDate_AdmissionProcessView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "0.01"^^xsd:float ;
.
mscbis:simDP_applicationHasDate_EligibilityView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "0.01"^^xsd:float ;
.
mscbis:simDP_grade_AdmissionProcessView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:softTFIDFJaroWinkler ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simDP_grade_EligibilityView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:softTFIDFJaroWinkler ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simDP_personHasCountryOfResidence_AdmissionProcessView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simDP_personHasCountryOfResidence_EligibilityView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simDP_personHasFamilyName_AdmissionProcessView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simDP_personHasFamilyName_EligibilityView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simDP_personHasFirstName_AdmissionProcessView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simDP_personHasFirstName_EligibilityView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
```

```
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simDP_professionalExperienceFieldDuration_AdmissionProcessView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simDP_professionalExperienceFieldDuration_EligibilityView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simDP_professionalExperienceField_AdmissionProcessView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simDP_professionalExperienceField_EligibilityView
  rdf:type sim:DatatypePropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:localSimilarityFunction sim:levenshtein ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_academicQualificationConsistsOfDegreeSubject_AdmissionPro
cessView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_academicQualificationConsistsOfDegreeSubject_EligibilityV
iew
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_academicQualificationConsistsOfDegreeType_AdmissionProces
sView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_academicQualificationConsistsOfDegreeType_EligibilityView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_academicQualificationConsistsOfDegree_AdmissionProcessVie
w
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_academicQualificationConsistsOfDegree_EligibilityView
  rdf:type sim:ObjectPropertySimilarity ;
```

```
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_academicQualificationConsistsOfFinalDegreeUniversity_Admi
ssionProcessView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_academicQualificationConsistsOfFinalDegreeUniversity_Elig
ibilityView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_applicationToProgramme_AdmissionProcessView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_applicationToProgramme_EligibilityView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_mscBISAdmissionCaseIsCharacterizedByApplication_Admission
ProcessView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_mscBISAdmissionCaseIsCharacterizedByApplication_Eligibili
tyView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_mscBISAdmissionCaseIsCharacterizedByPerson_AdmissionProce
ssView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "2"^^xsd:float ;
.
mscbis:simOP_mscBISAdmissionCaseIsCharacterizedByPerson_EligibilityVie
w
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "2"^^xsd:float ;
.
mscbis:simOP_mscBISAdmissionCaseIsCharacterizedByTaskObjective_Admissi
onProcessView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
```

```
  sim:weight "3"^^xsd:float ;
.
mscbis:simOP_mscBISAdmissionCaseIsCharacterizedByTaskObjective_Eligibi
lityView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "3"^^xsd:float ;
.
mscbis:simOP_personHasAcademicQualifikation_AdmissionProcessView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "3"^^xsd:float ;
.
mscbis:simOP_personHasAcademicQualifikation_EligibilityView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "3"^^xsd:float ;
.
mscbis:simOP_personHasBusinessRole_AdmissionProcessView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "0.1"^^xsd:float ;
.
mscbis:simOP_personHasBusinessRole_EligibilityView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "0.1"^^xsd:float ;
.
mscbis:simOP_personHasNationality_AdmissionProcessView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_personHasNationality_EligibilityView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_personHasProfessionalExperience_AdmissionProcessView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "2"^^xsd:float ;
.
mscbis:simOP_personHasProfessionalExperience_EligibilityView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "2"^^xsd:float ;
.
mscbis:simOP_universityIsLocated_AdmissionProcessView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
```

```
mscbis:simOP_universityIsLocated_EligibilityView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_universityPlace_AdmissionProcessView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_universityPlace_EligibilityView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_universityRating_AdmissionProcessView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:AdmissionProcessView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:simOP_universityRating_EligibilityView
  rdf:type sim:ObjectPropertySimilarity ;
  sim:belongsToCaseView mscbis:EligibilityView ;
  sim:globalSimilarityFunction sim:average ;
  sim:weight "1"^^xsd:float ;
.
mscbis:universityPlace
  sim:similarity mscbis:simOP_universityPlace_AdmissionProcessView ;
  sim:similarity mscbis:simOP_universityPlace_EligibilityView ;
.
mscbis:universityRating
  sim:similarity mscbis:simOP_universityRating_AdmissionProcessView ;
  sim:similarity mscbis:simOP_universityRating_EligibilityView ;
.
<http://ikm-group.ch/MScBISOntologyCaseSimilarityModel>
  rdf:type owl:Ontology ;
  owl:imports <http://ikm-group.ch/MScBISOntology> ;
  owl:imports <http://ikm-group.ch/MScBISOntologyCaseModel> ;
  owl:imports <http://ikm-group.ch/ProcessExecutionOntology> ;
  owl:imports <http://ikm-group.ch/cbr> ;
  owl:versionInfo "Created with TopBraid Composer" ;
.
po:TaskObjective
  sim:similarity mscbis:simAP_TaskObjective_AdmissionProcessView ;
  sim:similarity mscbis:simAP_TaskObjective_EligibilityView ;
.
archi:BusinessRole
  sim:similarity mscbis:simAP_BusinessRole_AdmissionProcessView ;
  sim:similarity mscbis:simAP_BusinessRole_EligibilityView ;
.
top:Country
  sim:similarity mscbis:simAP_Country_AdmissionProcessView ;
  sim:similarity mscbis:simAP_Country_EligibilityView ;
.
```

Appendix-F: Exemplary ICEBERG-PE BPM Service

Listing-Appx 4: Exemplary ICEBERG-PE BPM service consumes CBR core service

```
@Named
public class EligibilityCheckService implements Serializable {
  private final static Logger LOGGER = Logger.getLogger(EligibilityCheckService.class.getName());

  @Inject @ProcessVariable
  private Object mscbis_MScBISAdmissionCase_URI;

  @Inject @ProcessVariable
  private Object mscbis_personHasCountryOfResidence;

  @Inject @ProcessVariable
  private Object mscbis_applicationHasAdditionalInformation;

  @Inject
  private BusinessProcess businessProcess;

  @Inject
  private CBRServicesInterface cbrService;

  public void checkEligibility() {
    LOGGER.info("checkEligibility called!!!");
      // 29 lines...
  }

  public void checkEligibilityCompleted() {
    LOGGER.info("checkEligibilityCompleted called!!!");
    CaseViewVO caseViewVO = cbrService.findCaseViewByUri(MSCBIS_NS.ADMISSION_PROCESS_VIEW.toString());
    CaseInstanceVO result = cbrService.getCaseInstance((String) mscbis_MScBISAdmissionCase_URI, caseViewVO);
    //Person
    ObjectPropertyInstanceVO caseIsChByPerson = new ObjectPropertyInstanceVO();
    for (ObjectPropertyInstanceVO objectPropertyInstanceVO : result.getObjectProperties()) {
```

```
        if (objectPropertyInstanceVO.getTypeUri().equals(MSCBIS_NS.MSCBIS +
            "mscBISAdmissionCaseIsCharacterizedByPerson")) {
          caseIsChByPerson = objectPropertyInstanceVO;
          break;
        }
      }
      List<IndividualVO> persons = caseIsChByPerson.getRangeClassInstances();
      CaseInstanceVO person = (CaseInstanceVO) persons.get(0);
      List<LiteralPropertyValueVO> literalProperties = person.getLiteralProperties();
      LiteralPropertyValueVO literalPropertyValueVO = new LiteralPropertyValueVO();
      literalPropertyValueVO.setUri(MSCBIS_NS.MSCBIS + "personHasCountryOfResidence");
      literalPropertyValueVO.setValue((String) mscbis_personHasCountryOfResidence);
      literalProperties.add(literalPropertyValueVO);
      person.setLiteralProperties(literalProperties);
      persons.add(person);
      //Additional Information
      ObjectPropertyInstanceVO caseIsChByApplication = new ObjectPropertyInstanceVO();
      for (ObjectPropertyInstanceVO objectPropertyInstanceVO : result.getObjectProperties()) {
        if (objectPropertyInstanceVO.getTypeUri().equals(MSCBIS_NS.MSCBIS +
            "mscBISAdmissionCaseIsCharacterizedByApplication")) {
          caseIsChByApplication = objectPropertyInstanceVO;
          break;
        }
      }
      List<IndividualVO> applications = caseIsChByApplication.getRangeClassInstances();
      CaseInstanceVO application = (CaseInstanceVO) applications.get(0);
      for (LiteralPropertyValueVO literalPropertyValueVORes : application.getLiteralProperties()){
        if (literalPropertyValueVORes.getUri().equals(MSCBIS_NS.MSCBIS + "applicationHasAdditionalInformation")){
          literalPropertyValueVO = literalPropertyValueVORes;
          break;
        }
      }
      literalPropertyValueVO.setValue((String) mscbis_applicationHasAdditionalInformation);
      cbrService.createOrUpdateCase(caseViewVO, result);
    }
```

Appendix-G: Case Examples in CBR Repository

Listing-Appx 5: Case A

```
# baseURI: http://ikm-group.ch/MScBISCaseDataOntologyCaseA
# imports: http://ikm-group.ch/MScBISOntology
# imports: http://ikm-group.ch/MScBISOntologyCaseModel
# imports: http://ikm-group.ch/MScBISOntologyCaseSimilarityModel
# imports: http://ikm-group.ch/ProcessExecutionOntology
# imports: http://ikm-group.ch/cbr

@prefix archi: <http://ikm-group.ch/archiMEO/archimate#> .
@prefix cbr: <http://ikm-group.ch/cbr#> .
@prefix dcterms: <http://purl.org/dcterms#> .
@prefix elements: <http://purl.org/dc/elements/1.1#> .
@prefix eo: <http://ikm-group.ch/archiMEO/eo#> .
@prefix foaf: <http://xmlns.com/foaf/spec#> .
@prefix mscbis: <http://ikm-group.ch/MScBISOntology#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix po: <http://ikm-group.ch/ProcessExecutionOntology#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sim: <http://ikm-group.ch/similarity#> .
@prefix top: <http://ikm-group.ch/archiMEO/top#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://ikm-group.ch/MScBISCaseDataOntologyCaseA>
  rdf:type owl:Ontology ;
  owl:imports <http://ikm-group.ch/MScBISOntology> ;
  owl:imports <http://ikm-group.ch/MScBISOntologyCaseModel> ;
  owl:imports <http://ikm-group.ch/MScBISOntologyCaseSimilarityModel>;
  owl:imports <http://ikm-group.ch/ProcessExecutionOntology> ;
  owl:imports <http://ikm-group.ch/cbr> ;
  owl:versionInfo "Created with TopBraid Composer" ;
.
mscbis:Application_Peter_Nicolasia
  rdf:type mscbis:Application ;
  mscbis:applicationHasAdditionalInformation "Additional information:
nothing" ;
  mscbis:applicationHasDate "2014-12-23"^^xsd:date ;
  mscbis:applicationToProgramme mscbis:MScBISProgramme ;
.
mscbis:Bachelor_of_commerce_in_Business_management_Peter_Nicolasia
  rdf:type mscbis:AcademicQualification ;
  mscbis:academicQualificationConsistsOfDegree mscbis:Bachelor ;
  mscbis:academicQualificationConsistsOfDegreeSubject
mscbis:Business_Management ;
  mscbis:academicQualificationConsistsOfDegreeType mscbis:Management ;
  mscbis:academicQualificationConsistsOfFinalDegreeUniversity
mscbis:UNISA ;
  mscbis:academicQualificationConsistsOfGrade 5 ;
.
mscbis:LearnedMScBISAdmissionCase_A
  rdf:type cbr:CaseStateHistory ;
  cbr:belongsToCase mscbis:MScBISAdmissionCase_A ;
  cbr:caseStateTimestamp "2015-02-04T23:30:31.08"^^xsd:dateTime ;
  cbr:hasCaseState cbr:Learned_CaseState ;
.
mscbis:MScBISAdmissionCase_A
```

```
  rdf:type mscbis:MScBISAdmissionCase ;
  mscbis:mscBISAdmissionCaseIsCharacterizedByApplication
mscbis:Application_Peter_Nicolasia ;
  mscbis:mscBISAdmissionCaseIsCharacterizedByPerson
mscbis:Peter_Nicolasia ;
  mscbis:mscBISAdmissionCaseIsCharacterizedByTaskObjective
mscbis:TaskObjective_Eligibility_Check ;
  mscbis:mscBISAdmissionCaseIsCharacterizedByTaskObjective
mscbis:TaskObjective_Eligibility_History ;
  cbr:caseItem mscbis:case_item_MScBISAdmissionCase_A_Applicant ;
  cbr:caseItem mscbis:case_item__CV_Peter_Nicolasia ;
  cbr:caseReleasedAtDate "2014-08-01"^^xsd:date ;
  cbr:cbrCaseHasSolutionDescription "<html dir=\"ltr\"><head></head>
<body contenteditable=\"true\"><h1><font face=\"Segoe UI\" size=\"6\">
Case A - Process Fragment:</font></h1> <p><font face=\"Segoe UI\"
size=\"6\"><img src=\"file_repository/case_a.png\"> <br></font></p>
<p></p></body></html>"^^rdf:HTML ;
  cbr:reusedCases mscbis:MScBISAdmissionCase_B ;
  rdfs:label "Case A"@en ;
.
mscbis:Peter_Nicolasia
  rdf:type eo:Person ;
  mscbis:personHasAcademicQualifikation
mscbis:Bachelor_of_commerce_in_Business_management_Peter_Nicolasia ;
  mscbis:personHasBusinessRole mscbis:Applicant ;
  mscbis:personHasCountryOfResidence "Zurich,Switzerland" ;
  mscbis:personHasFamilyName "Nicolasia" ;
  mscbis:personHasFirstName "Peter" ;
  mscbis:personHasNationality eo:South-Africa ;
  mscbis:personHasProfessionalExperience
mscbis:ProfessionalExperience_Peter_Nicolasia_1 ;
  eo:personPerformsBusinessRole mscbis:Applicant ;
  foaf:personHasFamilyName "Nicolasia" ;
  foaf:personHasFirstName "Peter" ;
.
mscbis:ProfessionalExperience_Peter_Nicolasia_1
  rdf:type mscbis:ProfessionalExperience ;
  mscbis:professionalExperienceField "Data Analyst" ;
  mscbis:professionalExperienceFieldDuration 4 ;
.
mscbis:case_document__CV_Peter_Nicolasia
  rdf:type foaf:Document ;
  eo:documentHasSubjectBusinessRole mscbis:Applicant ;
  cbr:documentHasFilePath "file_repository/case_a.png" ;
  elements:documentHasFormat eo:png ;
  elements:documentHasTitle "CV Peter Nicolasia" ;
  elements:documentHasType eo:Image ;
  dcterms:documentHasCreationDate "2014-02-
12T23:03:40.009"^^xsd:dateTime ;
  dcterms:documentHasLatestAccessDate "2014-02-
12T23:03:40.009"^^xsd:dateTime ;
  dcterms:documentHasModifiedDate "2014-02-
12T23:03:40.009"^^xsd:dateTime ;
.
mscbis:case_item_MScBISAdmissionCase_A_Applicant
  rdf:type cbr:CaseItem ;
  cbr:caseItemRepresentedBy mscbis:Peter_Nicolasia ;
.
mscbis:case_item__CV_Peter_Nicolasia
  rdf:type cbr:CaseItem ;
  cbr:caseItemRepresentedBy mscbis:case_document__CV_Peter_Nicolasia ;
.
```

Listing-Appx 6: Case B

```
# baseURI: http://ikm-group.ch/MScBISCaseDataOntologyCaseB
# imports: http://ikm-group.ch/MScBISOntology
# imports: http://ikm-group.ch/MScBISOntologyCaseModel
# imports: http://ikm-group.ch/MScBISOntologyCaseSimilarityModel
# imports: http://ikm-group.ch/ProcessExecutionOntology
# imports: http://ikm-group.ch/cbr

@prefix archi: <http://ikm-group.ch/archiMEO/archimate#> .
@prefix cbr: <http://ikm-group.ch/cbr#> .
@prefix dcterms: <http://purl.org/dcterms#> .
@prefix elements: <http://purl.org/dc/elements/1.1#> .
@prefix eo: <http://ikm-group.ch/archiMEO/eo#> .
@prefix foaf: <http://xmlns.com/foaf/spec#> .
@prefix mscbis: <http://ikm-group.ch/MScBISOntology#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix po: <http://ikm-group.ch/ProcessExecutionOntology#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sim: <http://ikm-group.ch/similarity#> .
@prefix top: <http://ikm-group.ch/archiMEO/top#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://ikm-group.ch/MScBISCaseDataOntologyCaseB>
  rdf:type owl:Ontology ;
  owl:imports <http://ikm-group.ch/MScBISOntology> ;
  owl:imports <http://ikm-group.ch/MScBISOntologyCaseModel> ;
  owl:imports <http://ikm-group.ch/MScBISOntologyCaseSimilarityModel>
;
  owl:imports <http://ikm-group.ch/ProcessExecutionOntology> ;
  owl:imports <http://ikm-group.ch/cbr> ;
  owl:versionInfo "Created with TopBraid Composer" ;
.
mscbis:Application_Susan_Fisher
  rdf:type mscbis:Application ;
  mscbis:applicationHasAdditionalInformation "Additional information:
Student has been working in Switzerland for 4 years" ;
  mscbis:applicationHasDate "2014-12-23"^^xsd:date ;
  mscbis:applicationToProgramme mscbis:MScBISProgramme ;
.
mscbis:Bachelor_of_business_administration_BBA_in_Management_Susan_Fis
her
  rdf:type mscbis:AcademicQualification ;
  mscbis:academicQualificationConsistsOfDegree mscbis:Bachelor ;
  mscbis:academicQualificationConsistsOfDegreeSubject
mscbis:Business_Administration ;
  mscbis:academicQualificationConsistsOfDegreeType mscbis:Management ;
  mscbis:academicQualificationConsistsOfFinalDegreeUniversity
mscbis:Davenport_University ;
  mscbis:academicQualificationConsistsOfGrade 6 ;
.
mscbis:LearnedMScBISAdmissionCase_B
  rdf:type cbr:CaseStateHistory ;
  cbr:belongsToCase mscbis:MScBISAdmissionCase_B ;
  cbr:caseStateTimestamp "2015-02-04T23:30:31.08"^^xsd:dateTime ;
  cbr:hasCaseState cbr:Learned_CaseState ;
.
mscbis:MScBISAdmissionCase_B
  rdf:type mscbis:MScBISAdmissionCase ;
  mscbis:mscBISAdmissionCaseIsCharacterizedByApplication
mscbis:Application_Susan_Fisher ;
```

```
  mscbis:mscBISAdmissionCaseIsCharacterizedByPerson
mscbis:Susan_Fisher ;
  mscbis:mscBISAdmissionCaseIsCharacterizedByTaskObjective
mscbis:TaskObjective_Eligibility_Check ;
  mscbis:mscBISAdmissionCaseIsCharacterizedByTaskObjective
mscbis:TaskObjective_Eligibility_History ;
  cbr:caseItem mscbis:case_item_MScBISAdmissionCase_B_Applicant ;
  cbr:caseItem mscbis:case_item__CV_Susan_Fisher ;
  cbr:caseReleasedAtDate "2014-08-01"^^xsd:date ;
  cbr:cbrCaseHasSolutionDescription "<html
dir=\"ltr\"><head></head><body contenteditable=\"true\"><h1><font
face=\"Segoe UI\" size=\"6\">Case B -
Process Fragment:</font></h1><p><font face=\"Segoe UI\"
size=\"6\"><img
src=\"file_repository/case_b.png\"><br></font></p><p></p></body></html
>" ;
  cbr:reusedCases mscbis:MScBISAdmissionCase_A ;
  rdfs:label "Case B"@en ;
.
mscbis:ProfessionalExperience_Susan_Fisher_1
  rdf:type mscbis:ProfessionalExperience ;
  mscbis:professionalExperienceField "Software Development" ;
  mscbis:professionalExperienceFieldDuration 2 ;
.
mscbis:Susan_Fisher
  rdf:type eo:Person ;
  mscbis:personHasAcademicQualifikation
mscbis:Bachelor_of_business_administration_BBA_in_Management_Susan_Fis
her ;
  mscbis:personHasBusinessRole mscbis:Applicant ;
  mscbis:personHasCountryOfResidence "Basel,Switzerland" ;
  mscbis:personHasFamilyName "Fisher" ;
  mscbis:personHasFirstName "Susan" ;
  mscbis:personHasNationality eo:USA ;
  mscbis:personHasProfessionalExperience
mscbis:ProfessionalExperience_Susan_Fisher_1 ;
  eo:personPerformsBusinessRole mscbis:Applicant ;
  foaf:personHasFamilyName "Fisher" ;
  foaf:personHasFirstName "Susan" ;
.
mscbis:case_document__CV_Susan_Fisher
  rdf:type foaf:Document ;
  eo:documentHasSubjectBusinessRole mscbis:Applicant ;
  cbr:documentHasFilePath "file_repository/case_b.png" ;
  elements:documentHasFormat eo:PDF ;
  elements:documentHasTitle "CV Susan Fisher" ;
  elements:documentHasType eo:Image ;
  dcterms:documentHasCreationDate "2014-02-
12T23:03:40.009"^^xsd:dateTime ;
  dcterms:documentHasLatestAccessDate "2014-02-
12T23:03:40.009"^^xsd:dateTime ;
  dcterms:documentHasModifiedDate "2014-02-
12T23:03:40.009"^^xsd:dateTime ;
.
mscbis:case_item_MScBISAdmissionCase_B_Applicant
  rdf:type cbr:CaseItem ;
  cbr:caseItemRepresentedBy mscbis:Susan_Fisher ;
.
mscbis:case_item__CV_Susan_Fisher
  rdf:type cbr:CaseItem ;
  cbr:caseItemRepresentedBy mscbis:case_document__CV_Susan_Fisher ;
.
```