Table des Matières

Liste des figures 1

Liste des tables 4

Introduction 5

Chapitre A Les sémantiques dans les plateformes logicielles 9
1 La notion de plateforme logicielle 9
1.1 Définition 9
1.2 Le langage d'assemblage de programmes 10
1.3 Trame d'analyse 14
2 Les plateformes logicielles 15
2.1 The Modular simulation Component (MODCOM) 15
2.2 Virtual Laboratory Environment (VLE) 16
2.3 Open Modelling Interface (OpenMI) 19
2.4 Récapitulatif des caractéristiques des supports génériques 22
3 Les plateformes agronomiques 24
3.1 Les programmes agronomiques en question 25
3.2 Description des plateformes agronomiques choisies 26
3.2.1 Une construction ad hoc, la plateforme DSSAT 26
3.2.1.1 Les versions successives 26
3.2.1.2 DSSAT dans la trame d'analyse DRS 29
3.2.2 APES, instance de MODCOM 32
3.2.3 SEAMLESS-IF, instance d'OpenMI 38
3.3 Etude comparative des plateformes agronomiques 42
3.3.1 Architecture et sémantique 42
3.3.2 Impact du sens littéral des termes 45
3.3.3 Conclusion partielle 49
4 La constitution du dictionnaire 50
4.1 Reformatage d'un terme d'après un langage imposé 50
4.1.1 L'expérience APES 50
4.1.2 Discussion 53
4.2 Typage d'un terme et Pragmatique 54
4.2.1 Les deux formes de typages 55
4.2.2 Discussion 55
5 Conclusion de la partie A 56

tel-00458265, version 1 - 19 Feb 2010

Chapitre B Accès au sens implicite 57
1 Bibliographie linguistique 57
1.1 La question de la sémantique dans la production d'un énoncé 57
1.2 L'interrogation dans la production d'un énoncé 58
1.3 L'interrogation et la structure de réponse associée 59
1.3.1 Où et la Localisation 59
1.3.2 Pourquoi et la Raison 60
1.3.3 Comment et la Manière 61
1.3.4 Qui et l'Actant 62
1.3.5 Quand et la Temporalité 64
1.3.6 Quoi/Que et acte 65
2 Vers une structure de graphes 67
2.1 Exploitation de la littérature et formulation de graphes 67
2.1.1 Où et la Localisation 67
2.1.2 Pourquoi et la Raison 68
2.1.3 Comment et la Manière 69
2.1.4 Qui et l'Actant 70
2.1.5 Quand et la Temporalité 71
2.1.6 Quoi/que et acte 72
2.2 Structure de description et assemblage 73
2.2.1 L'hypothèse d'une structure commune aux groupes fonctionnels 73
2.2.1.1 La classe Organisation 74
2.2.1.2 La classe Référentiel 76
2.2.2.3 La classe Sémantique 77
2.2.2 La structure de description 78
2.3 Description du contexte 80
3 Proposition d'un modèle formel d'accès aux connaissances pragmatiques 81
3.1 Formalisation syntaxique d'un terme 82
3.2 Dictionnaires 84
3.3 Mise en œuvre du modèle formel 85
3.3.1 Les réseaux sémantiques 85
3.3.2 Représentation du modèle formel au moyen des graphes conceptuels 87
4 Discussion 90

Chapitre C Illustration 93
1 Exemple de description d'un système complexe 93
1.1 Description littérale du système biologique 94
1.2 Représentation formelle du système biologique 94
1.3 Vocabulaire et sens implicite 100
1.4 Transcription informatique 102
2 Discussion 104
Conclusion 107

tel-00458265, version 1 - 19 Feb 2010

Glossaire 111

Références 113
1 Productions de l'auteur du mémoire 113
1.1 Projet Européen SEAMLESS (2005-2009) 113
1.2 Réseau excellence Endure 114
1.3 Travaux en lien avec le mémoire 114
1.4 Autres travaux (dans revues internationales à comité de lecture) 114
2 Introduction 115
3 Chapitre A 117
4 Chapitre B 122
5 Chapitre C et Conclusion 125

Annexe 126

tel-00458265, version 1 - 19 Feb 2010

Liste des figures

01 Bipolarité (a) et valence d'un programme (b). 10
02 Les 18 formes d'interrelations entre deux éléments monovalents (Klir et Valach, 1967). 11
03 Sémantique d'une construction résultant de la mise en relation de deux briques monovalentes : (a) action, (b) interaction, (c) rétroaction, (d) parallélisme. 11
04 Interrelations entre deux éléments bivalents, appliqué au fonctionnement de la vie d'un animal (Grenievski, 1965). 12
05 Insertion de programmes dans MODCOM.. 15
06 Ordre d'appel des composants par MODCOM. 16
07 Insertion de programmes au sein de la plateforme VLE. 17
08 Exemple d'enchaînement de processeurs DEVS (Quesnel et al., 2009). 19
09 Enchaînement en cascade permis par OpenMI.. 20
10 Standard d'échange OpenMI version 1.0 (Gijsbers et al., 2005). 21
11 Architecture d'un programme de simulation d'une culture dans DSSAT (Jones, 2003). 28
12 Etat de la plateforme APES en 2006. (a) Composants logiciels (Donatelli et al., 2005). (b) exemple de cycle d'exécution. 33
13 Etat de la plateforme APES en 2008. (a) composants logiciels d'après (Donatelli, 2010). (b)Evolutions portées sur le cycle d'exécution de la Figure 12b. 34
14 Exemple de superposition de l'enchaînement d'appel et du besoin en données d'entrée des composants 'Crops' et 'Soil_water' pour 2006 (a) et 2008 (b). (b) est issu de (van Evert et al., 2007). 36
15 Les trois couches applicatives de la plateforme SEAMLESS-IF (van Ittersum et al., 2006). 39
16 Chaîne d'exécution des programmes de la couche End-User instaurée dans la plateforme SEAMLESS-IF en 2006 (a) et 2008 (b) issues respectivement de (van Ittersum et al., 2006) et (van Ittersum et al., 2008). 40
17 Les concepts 'Crop', 'Product' et leurs dérivés insérés dans l'ontologie de SEAMLESS-IF (Janssen et al., 2009). 41
18 Programme KAMEL, formulé dans le langage SIMILE, utilisé pour produire le composant Soilwater2. 52
19 Etude morphosyntaxique du Où (Hadermann, 1993). 60
20 Formes du temps selon Maingueneau (1994). 64
21 Typologie du verbe selon (François, 1989). 65
22 Typologie du lieu (a) et itinéraire (b). 68
23 Relation entre le lieu et son référentiel dans le cas d'une double localisation. 68
24 Sémantique de la localisation. 68
25 Relation de voisinage pour accéder au lieu dans le cas où celui-ci est un élément topologique. 68

tel-00458265, version 1 - 19 Feb 2010

26 Typologie des réponses obtenues par l'usage du 'pourquoi normal' (a) et constitution d'une chaîne causale (b). 69
27 Typologie des réponses obtenues par l'usage du 'pourquoi incolore'. 69
28 Typologie de la manière lorsque le circonstant est incident au verbe. D'après (El Hasnaoui, 2008). 69
29 Typologie des formes du savoir, d'après (Charlot, 1997). 69
30 Deux types d'actants (a) et transformation d'une construction tri-actancielle en bi-actancielle (b), d'après (Lazard, 1994). 70
31 Typologie de la désignation d'une personne. 70
32 Typologie du nombre. 71
33 Forme sagittale. 71
34 Le questionnement du temps. 71
35 Forme de la temporalité (a) et limites de la durée (b), d'après (Declerck, 1997). 71
36 Typologie des temps, d'après (Mascherin, 2008). 72
37 Logique relative des temps (Grenier, 2001). 72
38 Typologie du verbe d'après Tesnière (1988). 72
39 Typologie des verbes d'action, d'après (Mélis, 1983). 72
40 Classification des verbes d'après Dubois et Dubois-Charlier (1997). 73
41 Structure du groupe fonctionnel selon trois classes. 74
42 Organisation du groupe fonctionnel de manière. Typologie de la constitution (a) et structure de la façon de faire (b). 74
43 Organisation du groupe fonctionnel de l'actant. Typologie de l'actant (a) et chaîne agentielle (b). 75
44 Organisation du groupe fonctionnel de l'Acte. Typologie de l'acte (a) et chaîne d'actions (b). 76
45 Organisation générale d'un groupe fonctionnel : typologie de la structure (a) et structure de la chaîne (b). 76
46 Sémantique de l'Actant. 78
47 Sémantique de la Temporalité. 78
48 Sémantique de la manière. 78
49 Structure de Description, établie au moyen du principe d'incidence, pour une construction uni-actancielle (a) et bi-actancielle (b). 79
50 Simultanéité de conduite d'un acte exprimé par le circonstant de manière du point de vue syntaxique (a) et exprimé au moyen des groupes fonctionnels (b). 80
51 Typologie du contexte d'après (Sarfati, 2002). 80
52 Exemple de relations entre éléments des classes Organisation (GO), Sémantique (GS) et Référentiel (GR). 82
53 Description multi-niveaux d'un terme 84
54 Exemple de graphe de connaissances selon la méthode des réseaux partitionnés, d'après Kayser (1997).. 86
55 Exemple de graphe conceptuel (Kayser, 1997) constitué d'un vocabulaire et de deux graphes de connaissances. 87
56 Succession d'éléments décrite au moyen des graphes conceptuels. 89

tel-00458265, version 1 - 19 Feb 2010

57 Structure de description décrite au moyen des graphes conceptuels. 89
58 Exemple de description d'un groupe fonctionnel. 89
59 simplification de la présentation d'un graphe de Connaissances, dans lequel les marqueurs de la classe Organisation sont mis en avant 95
60 Graphe de connaissances du contexte d'élaboration de l'illustration. 96
61 Graphes de connaissances, représentant la relation mutualiste de *Lobesia botrana* et *Botrytis cinerea* en regard de la vigne (colonne de gauche), et classes Sémantiques des groupes fonctionnels d'Agent et d'Objet (colonne de droite).. 100
62 Vocabulaires des groupes fonctionnels d'Agent (a), d'Objet (b), de Localisation (c) et de Temporalité (d).. 102
63 Interrelations entre *B_cinerea*, *L. botrana* et le grain de raisin de la vigne. 103

tel-00458265, version 1 - 19 Feb 2010

Liste des tables

1 Méthode de transmission des données d'un programme P_A à l'attention d'un programme P_B, et niveau de couplage de P_B vis-à-vis de P_A (Pressman, 1992). 13
2 Trame d'analyse DRS des plateformes logicielles. 14
3 Synthèse de description des supports génériques MODCOM, OpenMI et VLE. 24
4 Evolution des versions successives de DSSAT selon la trame d'analyse DRS. 31
5 Evolution des versions successives d'APES selon la trame d'analyse DRS. 37
6 Synthèse des versions de SEAMLESS-IF selon la trame d'analyse DRS. 42
7 Principales caractéristiques des plateformes logicielles utilisées 50
8 Description de l'entrée verbale 'aboyer', d'après (François, 2007)..... 66

tel-00458265, version 1 - 19 Feb 2010

Rapport-Gratuit.com

tel-00458265, version 1 - 19 Feb 2010

Introduction

De la génétique à l'agronomie et l'écologie, les sciences du vivant s'intéressent à des systèmes complexes dont le comportement n'a longtemps pu être approché que grâce à l'observation et l'expérimentation. En outre, alors que la recherche est organisée par champs disciplinaires, les systèmes observés constituent le plus souvent autant de sous-systèmes en interaction au sein d'un système étudié plus vaste. En agriculture, par exemple, la production agricole, la gestion de la fertilité des sols, la qualité des produits issus d'une filière de production, ou encore l'impact environnemental des pratiques mobilisent plusieurs systèmes complexes comme le sol, les peuplements végétaux, les organisations de producteurs ou encore les bassins versants, etc. Les systèmes envisagés par les disciplines du vivant sont donc pour la plupart des systèmes complexes constitués d'un grand nombre d'entités en interaction, dont le comportement et l'évolution ne peuvent pas être appréhendés par un observateur unique.

A partir des années 1960, l'apparition de l'informatique a permis l'essor d'une alternative à l'expérimentation pour la synthèse et l'utilisation des savoirs disponibles au sein des disciplines et entre disciplines. Dans le cas du fonctionnement des cultures, par exemple, les premiers programmes heuristiques ont été développés afin d'améliorer la compréhension du comportement des peuplements végétaux en regard des processus physiologiques impliqués dans la croissance et le développement des plantes (Bouman, 1996). Au fil du temps, la diversification des objets d'étude a abouti à la production d'un nombre important de programmes de simulation par les équipes de recherche de toutes disciplines. En sus d'une fonctionnalité dédiée à l'explication des mécanismes biologiques complexes, l'intérêt des programmes a rapidement été reconnu dans les domaines de l'extrapolation et de la prédiction, avec pour conséquence la création de nouveaux savoirs liés à l'association de programmes. A titre d'exemple, si on se réfère aux programmes développés ou utilisés au CIRAD (Centre de coopération Internationale en Recherche Agronomique pour le Développement) durant les 15 dernières années, on observe une évolution des fonctionnalités et des supports de développement. Dans les années 1980-90, c'est le langage procédural qui est utilisé pour le développement d'applications dédiées à la croissance des cultures. Dans ce groupe, on peut citer par exemple GOSSYM (GOSSYpium hirsutum L. simulation Model ; Baker et al., 1983) ou STICS (Simulateur mulTIdisciplinaire pour les Cultures Standard ; Brisson et al., 2002), développés en FORTRAN d'abord, puis en « C ». A la fin des années 90, le langage objet commence à être utilisé, à l'exemple de CAPSIS (Computer-Aided Projection of Strategies In Sylviculture ; Courbaud et al., 2001) à l'usage de l'agroforesterie ou OPENALEA (software environment for plant modeling ; Pradal et al., 2008) pour l'architecture des plantes. CAPSIS et OPENALEA utilisent respectivement les langages JAVA et PYTHON. Parallèlement à ces langages d'informaticiens professionnels, des langages dédiés à la modélisation sont adoptés depuis 2000, permettant aux chercheurs des sciences du vivant d'assurer eux-mêmes le développement des programmes. Dans cette catégorie on peut citer SIMBA (SIMulation of BAnana crop harvest ; Tixier et al., 2004) et KAMEL (Braudeau et al., 2009), à l'usage respectivement des fonctionnements d'une culture de banane et de l'hydrologie du sol. Ces programmes utilisent respectivement les langages STELLA® et SIMILE (Muetzelfeldt et Massheder, 2003).

Au début des années 1980, l'idée d'accéder à de nouveaux savoirs par voie d'assemblage de programmes a donné naissance aux plateformes logicielles. De telles constructions visent le

tel-00458265, version 1 - 19 Feb 2010

réemploi des programmes afin de permettre, en théorie, à une équipe d'accéder à des objets de recherche pluridisciplinaire par association de savoirs. L'hypothèse sous-jacente à la construction de ces plateformes logicielles est que l'assemblage de programmes de simulation de sous-systèmes permet d'accéder à la simulation du système global auquel ils participent. L'idée de plateforme logicielle dédiée à l'agronomie a émergé au sein du projet IBSNAT (International Benchmark Sites Network for Agrotechnology Transfer ; IBSNAT, 1993), dans l'objectif de disposer d'un support de transfert de technologies dans le domaine du fonctionnement des cultures agricoles. La plateforme DSSAT (Jones et al., 2003) permet, à partir d'une bibliothèque de composants « culture » (blé, soja, riz etc..) et du sol, d'accéder à la dimension temporelle de l'évolution de la fertilité des sols sous l'effet des successions culturales. Cette idée de plateforme, née aux Etats-Unis, a ensuite été adoptée en Europe avec les projets SEAMLESS (System for Environmental and Agricultural Modelling - Linking European Science and Society ; van Ittersum et al., 2008) et RECORD (Bergez et al., 2007). L'enjeu actuel des plateformes logicielles est de répondre aux nouvelles questions posées à la recherche en regard de l'évolution de la demande sociale. La remise en cause de l'usage des pesticides, par exemple, renvoie à des systèmes de production complexes incluant les bio-agresseurs (Martin et al., 2008b). Par ailleurs, le Développement Durable implique que les processus de développement soient appréhendés selon le triple point de vue de leur durabilité écologique, de leur viabilité économique et de l'équité sociale (Tuot, 2007). Enfin, le changement climatique pose la question de l'adaptabilité des sociétés et des modes de production. Dans cette optique, les savoirs associés au sein de plateformes logicielles comme RECORD ou SEAMLESS-IF dépassent le cadre des sciences techniques pour alimenter les travaux en économie, politique et sciences sociales. Dans le contexte d'assemblage de savoirs pluridisciplinaires inhérent aux plateformes logicielles, la cohérence de la juxtaposition des savoirs contenus dans les programmes est source de questionnement.

Le principe prévalant à la construction de ces plateformes logicielles est celui de la réutilisation des programmes et des supports d'assemblages préexistants. Pour reprendre les exemples présentés plus haut, RECORD et SEAMLESS-IF utilisent respectivement les supports génériques d'assemblage VLE (Virtual Laboratory Environment ; Ramat et Preux, 2003 ; Quesnel et al., 2009) et OpenMI (Open Model Integration environment ; Gregersen et al., 2007). Le principe de réutilisation des programmes suppose que l'on puisse associer des programmes (i) établis selon des contextes historiques et/ou locaux contrastés et (ii) développés selon des méthodes informatiques contrastées. L'origine des contrastes provient alors de l'évolution des concepts et paradigmes scientifiques en vigueur et de la pluralité des écoles de pensée. Le principe de réutilisation pose donc la question de la cohérence de la juxtaposition des programmes en tant qu'assemblage de savoirs exprimés sous forme logicielle. Le principe de réutilisation des supports génériques quant-à-lui repose sur l'utilisation d'un langage d'assemblage préétabli. Ce langage dispose d'un jeu fini de règles syntaxiques, qui définissent les possibilités d'association des programmes. Ce faisant, elles contraignent aussi les possibilités d'association. A titre d'exemple, MODCOM n'utilise que la relation d'ordre total (Hillyer, 2003). Par construction, un tel support ne peut pas rendre compte du parallélisme nécessaire pour aborder certains systèmes du vivant (Systèmes Multi Agents ; Ferber, 1995). L'utilisation d'un langage d'assemblage préétabli pose donc la question des limites de formulation d'un assemblage de programmes au travers d'une syntaxe en regard de l'objectif de composition attendu.

Au sein du langage, Sabah (1988) distingue quatre niveaux de connaissances : (i) les connaissances morphologiques qui portent sur la forme canonique des termes et leur emploi, (ii) les connaissances syntaxiques qui expriment les relations entre les termes sous forme de

LE NUMERO 1 MONDIAL DU MÉMOIRES

règles, (iii) les connaissances sémantiques pour l'expression du sens littéral et enfin (iv) les connaissances pragmatiques qui permettent d'accéder aux aspects implicites du sens.
Dans le cadre de l'Ingénierie Logicielle, on trouve de nombreux travaux relatifs aux techniques de conception et de mise en relation de programmes (Pressman, 1992 ; 1996 ; 2001 ; 2004 ; 2009). Vis-à-vis du langage, ces travaux traitent donc exclusivement de syntaxe et la question de la sémantique n'est pas abordée. En intelligence Artificielle, la sémantique et son lien à la syntaxe sont traités. Les langages logico-mathématiques (langages logiques, réseaux sémantiques, logiques de description) se composent (i) d'un ensemble de littéraux, (ii) de règles syntaxiques et d'inférence permettant de produire de nouveaux énoncés (les conclusions) à partir d'énoncés préexistants (les prémisses), et (iii) une fonction d'interprétation qui permet de vérifier le respect des clauses de vérité des conclusions en regard de celles des prémisses. Les Langages Logiques (classiques et non classiques) comportent (i) un ensemble de symboles (constante, variable, prédicat, fonction, etc.), (ii) une syntaxe permettant de combiner les symboles au moyen de connecteurs logiques (négation, ou/et, quantification, etc.) pour la production de formules, et (iii) d'une sémantique attachée aux symboles et aux formules. Dans ce type de langages, la sémantique « s'exprime souvent en termes booléens (la situation est vraie ou ne l'est pas) » (Sabah, 1988). Un réseau sémantique est un graphe orienté et étiqueté dans lequel un nœud représente un concept, et un arc reliant des nœuds représente une relation sémantique entre les concepts reliés (Kayser, 1997 ; Schreiber, 2008). Dans la famille des réseaux sémantiques, les Graphes Conceptuels (Sowa, 1984 ; Chein et Mugnier, 2008) présupposent de l'existence d'une organisation relative des concepts à assembler. Un graphe conceptuel est défini par (i) un vocabulaire, (ii) une collection de graphes reliant les concepts au moyen des relations sémantiques, et (iii) une collection de relations, dites de coréférence, qui permettent de représenter un même objet au moyen de plusieurs concepts. Pour terminer sur les langages logico-mathématiques, la Logique de Description combine le Langage Logique et les Réseaux Sémantiques (Baader et al., 2003). Par rapport aux Graphes Conceptuels, le vocabulaire des Logiques de Description est complété par un ensemble d'opérateurs spécifiques à la Logique adoptée, qu'elle soit classique (prédicat ou du second ordre) ou non classique (modale, etc.). Le langage de manipulation des ontologies OWL (Web Ontology Language), conçu à l'attention du Web Sémantique, est une application de la Logique de Description (Smith et al, 2004).
Les langages logico-mathématiques considèrent les trois premiers niveaux de connaissance évoqués par Sabah (1988) : morphologie, syntaxe et sémantique. Dans les Graphes Conceptuels comme dans les Logiques de Description, la sémantique des concepts et des relations est spécifiée dans le vocabulaire, exprimé sous forme de graphe. Ce graphe est établi au moyen de la relation de subsomption 'Is A'.

Les connaissances pragmatiques ne sont pas considérées dans ces langages. A partir du moment où les termes et leur emploi sont définis dans le cadre du langage, la prise en compte de l'aspect implicite du sens n'est pas nécessaire. Or, dans la thèse, nous nous intéressons à l'association de programmes des sciences du vivant développés au moyen de différents langages informatiques. Dans cet exercice, les termes (les programmes) et leur emploi sont extrinsèques au langage d'assemblage. Dans la mesure où chaque programme à associer est développé dans un cadre particulier, il n'est pas possible d'ignorer les aspects implicites du sens inhérents à chaque programme pour leur assemblage. Le principe de réutilisation impose un cadre particulier à l'exercice. En premier lieu, les programmes à assembler réunissent des savoirs redevables de thématiques particulières au sein d'un même champ disciplinaire (la pédologie, l'écophysiologie, l'entomologie etc. dans le champ de l'agronomie par exemple). Au sein des thématiques, il existe une pluralité de paradigmes scientifiques intra-disciplinaires, en lien avec les équipes et l'évolution des connaissances. En second lieu,

tel-00458265, version 1 - 19 Feb 2010

l'assemblage amène à considérer des programmes établis dans des champs disciplinaires distincts. Enfin, l'assemblage est réalisé au moyen d'un langage particulier, choisi par les équipes, et doté de caractéristiques propres.

L'exercice amène aux questions de la thèse :
Q1 : quels sont les aspects implicites du sens dans l'assemblage ?
Q2 : les caractéristiques morphologiques, syntaxiques et sémantiques du langage d'assemblage limitent-elles la capacité à représenter les systèmes complexes étudiés par les sciences du vivant ?
et Q3 : Comment accéder aux connaissances pragmatiques dans le cadre de l'assemblage de programmes ?

Afin de répondre à Q1 et Q2, nous avons choisi d'étudier trois supports génériques d'assemblage, à savoir MODCOM, OpenMI et VLE, et trois plateformes logicielles agronomiques, DSSAT, APES, et SEAMLESS-IF (Chapitre A). Les plateformes logicielles agronomiques APES et SEAMLESS-IF utilisent respectivement les supports génériques MODCOM et OpenMI. Ces exemples nous amènent à considérer quatre langages d'assemblage : génériques pour MODCOM, OpenMI et VLE, et *ad hoc* pour DSSAT. Pour notre étude, nous avons mis au point une trame d'analyse originale établie sur la base de travaux en cybernétique, ingénierie logicielle et intelligence artificielle. La trame considère les niveaux morphologiques, syntaxiques et sémantiques des langages d'assemblage tels que définis par Sabah (1988). Au travers de l'analyse, nous montrons que les règles syntaxiques inhérentes aux langages d'assemblages limitent l'aptitude de ceux-ci à représenter certains systèmes complexes. Nous montrons aussi que l'efficience des méthodes destinées à faire coïncider les entrées/sorties (métadonnées et ontologies) dépend du vocabulaire employé dans les programmes. Nous concluons de la nécessité de prendre en compte l'aspect implicite du sens pour résoudre la question du sens communiqué par les programmes (= sortie *versus* entrée).

Afin de répondre à Q3, nous proposons d'établir un vocabulaire permettant l'accès au sens implicite (Chapitre B). Pour ce faire, nous proposons un modèle de description, établi à partir de la linguistique pragmatique et de travaux en morphosyntaxe. L'interrogation partielle en mode direct permet de conférer une structure de description basée sur les sept pronoms et adverbes interrogatifs que propose la langue française. A chacun de ces pronoms et adverbes, nous associons une typologie de réponses établie à partir de la bibliographie en linguistique. L'organisation de ces typologies nous permet d'établir une structure formelle de description du contenu des programmes en vue d'établir le vocabulaire. Dans le chapitre C, nous discutons de l'utilisation d'un formalisme de réseau sémantique existant (graphes conceptuels) comme support d'implémentation du modèle de description, et l'intérêt du travail est illustré. En conclusion, nous statuons sur nos apports en regard des questions posées et livrons des questions de recherche en perspective à ce travail.

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

Chapitre A
Les sémantiques dans les plateformes logicielles

La notion de plateforme logicielle mérite d'être explicitée dans le sens où, pour un agronome, il s'agit de la construction résultante de l'assemblage des programmes, et pour les informaticiens, c'est une notion plus ou moins floue qu'il convient d'éclairer à partir de la bibliographie.

1 La notion de plateforme logicielle

L'objet de ce paragraphe est double, d'une part donner une définition de la notion de plateforme propre au contexte de notre travail et d'autre part, une fois cette définition admise, établir une trame d'analyse des plateformes logicielles.

1.1 Définition

Différentes définitions de la notion de plateforme logicielle (framework) existent dans la littérature. Selon Campbell et al (1991), "a framework is an architectural design for object-oriented systems. It describes the components of the system and the way they interact". D'après (Fayad et Schmidt, 1997), "a framework is a reusable, 'semi-complete' application that can be specialized to produce custom applications". D'après (Brugali et al., 1997), "a framework is an integrated set of domain-specific software components that can be reused to create applications". D'après (Hillyer et al., 2003), "a framework enables the assembly of simulation models from previously and independently developed models". Enfin, d'après (Murray et al., 2004), "a framework is a reusable design that requires software components to function".

Ces définitions s'accordent sur le fait qu'une plateforme logicielle permet de réunir des éléments, qu'ils soient objets pour Campbell et Fayad, composants logiciels pour Brugali et Murray, ou programmes pour Hillyer. Cependant, elles divergent sur ce qu'est formellement une plateforme. Pour Campbell, il s'agit d'une conception architecturale (architectural design) qui supporte l'interaction d'éléments. Pour Fayad, c'est une application « générique » qui permet de produire des applications spécifiques. Pour Hillyer, il s'agit d'un support d'assemblage de programmes développés indépendamment. Pour Murray enfin, il s'agit d'une application qui nécessite l'insertion d'éléments pour fonctionner. Ces auteurs ont par conséquent un point de vue différent sur la notion de plateforme : depuis la structure d'assemblage à l'outil qui nécessite l'instanciation pour être utilisé.

Bien que chaque auteur en livre une définition particulière**,** nous admettrons qu'***une plateforme logicielle est une structure d'accueil d'éléments exogènes. Cette structure d'accueil est dotée d'un langage dévolu à l'assemblage des éléments, et dont la composition constitue un ensemble opérationnel.***

1.2 Le langage d'assemblage de programmes

Les concepts directeurs de l'assemblage ont été énoncés très tôt en cybernétique. Dans cette discipline, les éléments sont appréhendés sous la forme de briques élémentaires bipolaires, c'est-à-dire disposant d'un jeu d'entrées et de sorties (Figure 01a). Chaque brique est caractérisée par une valence (Klir et Valach, 1967), c'est-à-dire un nombre de fonctions exécutables par une source extérieure (Figure 01b). Une brique constituée de plusieurs fonctions exécutables est appelée multivalente.

Figure 01 : Bipolarité (a) et valence d'un programme (b).

La composition de l'ensemble opérationnel s'établit par l'explicitation de l'ordre d'appel des briques. Selon (Klir et Valach, 1967), deux types de relations fondamentales permettent de spécifier l'ordre d'appel : la série et le parallélisme. Par relation, Klir et Valach (1967) entendent un lien orienté dont la source est située en sortie d'une brique et le but en entrée d'une autre brique. Vis-à-vis des fonctions exécutables des briques, la source du lien orienté se situe à la fin de l'exécution d'une fonction d'une brique et le but correspond à l'appel d'une fonction d'une autre brique. Les différentes combinaisons de relations permettent de disposer de dix huit formes possibles d'interrelations entre deux briques monovalentes (Figure 02), c'est-à-dire ne disposant que d'une seule fonction exécutable.
Du point de vue sémantique (signification), la relation orientée correspond au concept d'action exercée par une brique sur la suivante. A ces 2 types de relations fondamentales (série et parallélisme) peuvent être associées l'action, l'interaction, la rétroaction et le parallélisme selon les schémas de mise en interrelation présentés en Figure 03. Les formes 1, 2 et 3 de la Figure 02 présentent les trois cas de rétroaction (ou feedback), qui correspond à l'effet de retour d'une action sur celui qui l'a engendré. Les formes 4, 5 et 6 correspondent à l'action d'une brique sur une autre combinée avec les trois formes de rétroaction. Les formes 7, 8 et 9 correspondent à l'interaction (ou action réciproque) entre les deux briques combinées avec les trois formes de rétroaction. Les formes de 10 à 18 répètent ces mêmes constructions pour un montage en parallèle des deux briques (Le Moigne, 1977).

En termes mathématiques, la composition de différentes formes de la Figure 02 permet de produire des structures de graphe simples. Une séquence composée d'éléments associés selon la forme 4 par exemple correspond à une structure d'ordre total. D'autres structures plus complexes sont également permises, à l'exemple du treillis par association de formes 2-4, 9-5 etc.

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

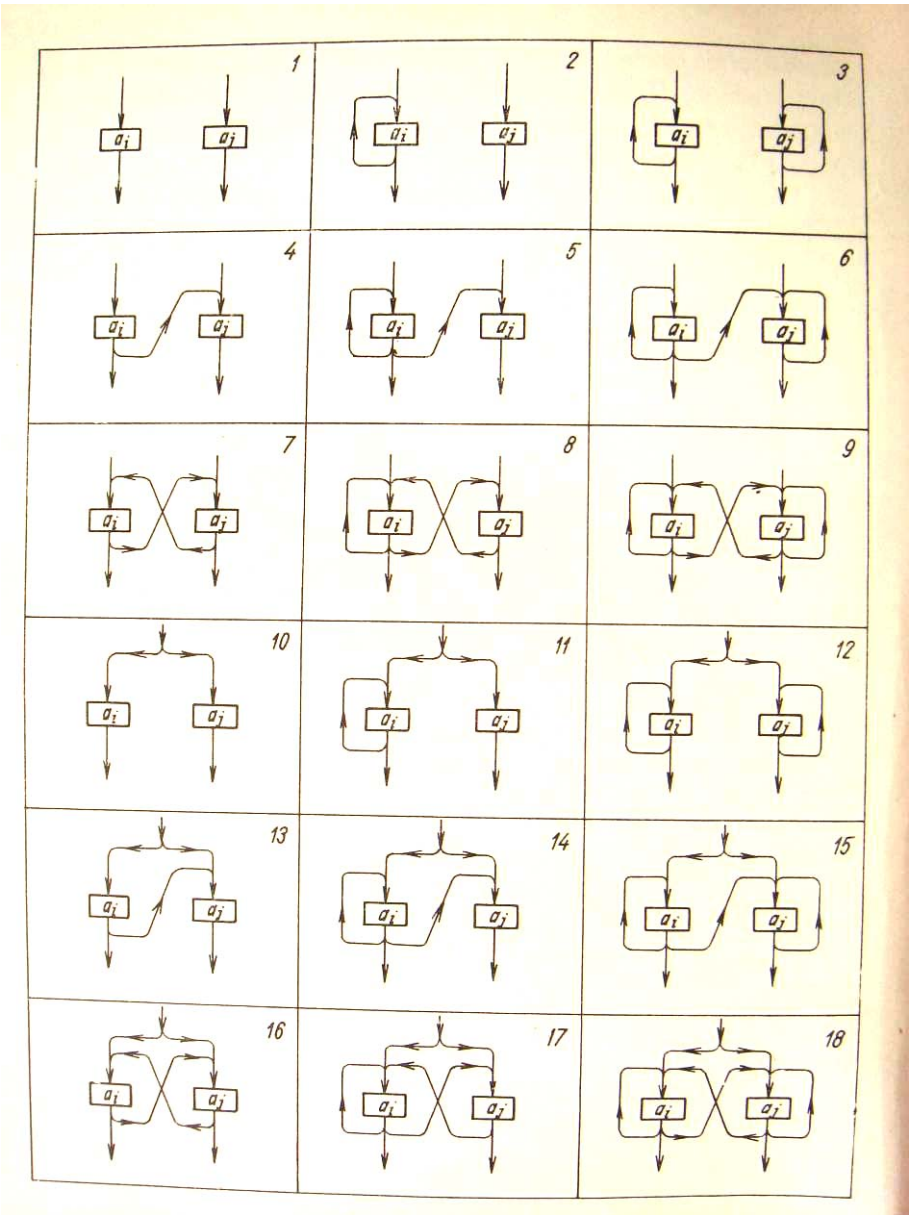


Figure 02 : Les 18 formes d'interrelations entre deux éléments monovalents (Klir et Valach, 1967).

Figure 03 : Sémantique d'une construction résultant de la mise en relation de deux briques monovalentes : (a) action, (b) interaction, (c) rétroaction, (d) parallélisme.

L'augmentation de la valence des briques a pour effet d'accroître le nombre possible de relations entre paires de briques. (Grenievski, 1965) par exemple, représente le fonctionnement, très simplifié, d'un animal au moyen de deux briques bivalentes (Figure 04). La brique C_1 représente le transformateur énergétique, la brique C_i le transformateur informationnel, et le symbole ∞ l'environnement de l'animal. Les flèches noires représentent la transmission de données énergétiques, et les doubles flèches la transmission de données

informationnelles. Les fonctions de la brique C_1 sont de transformer l'énergie relative à l'alimentation et au déplacement de l'animal. Celles de la brique C_i sont de traiter les informations relatives aux deux fonctions de C_1. La flèche initiée par le symbole ∞ et pointant sur C_1 et sur C_i représente respectivement une entrée d'énergie (nourriture par exemple) et une entrée d'informations (appel, etc.). La flèche sortant de C_1 et de C_i, et pointant sur le symbole ∞, correspond respectivement à une dépense d'énergie (déplacement de l'animal) et à l'émission d'informations à son entourage (cri, etc.). L'interaction entre C_1 et C_i permet, par exemple, à l'animal de se déplacer pour se nourrir. Certaines informations provenant de l'environnement ne pouvant être traitées à leur arrivée, la rétroaction sur C_i correspond au retardement de traitement de certaines informations.

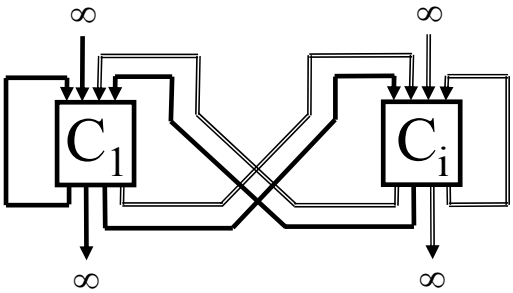


Figure 04 : Interrelations entre deux éléments bivalents, appliqué au fonctionnement de la vie d'un animal (Grenievski, 1965).

Dans le cas où le nombre de briques multivalentes est supérieur à deux, il est possible de combiner les échanges entre les briques. Les structures de graphes reliant les briques qui peuvent être élaborées en sont d'autant plus complexes.

L'instauration d'une relation entre deux briques a pour objet de transmettre des données (au sens large) entre briques. La transmission demande en premier lieu que soit établie la correspondance entre les entrées/sorties des briques. Cette opération s'effectue en utilisant le sens littéral de la donnée véhiculée. L'informatique propose deux méthodes de mise en correspondance. La première consiste à adjoindre une métadonnée à chaque entrée et chaque sortie, c'est-à-dire un ensemble d'informations caractérisant le type d'information véhiculée (NISO, 2004), puis en l'identification des correspondances entre métadonnées. Cette méthode permet (i) d'identifier le besoin de reformatage des informations transmises, à l'exemple du respect des unités dans lesquelles celles-ci sont exprimées, et (ii) de recenser les entrées qui n'ont pas de correspondance. La résolution de ce problème peut parfois être obtenue par combinaison d'informations détenues par la brique émettrice. Si ce n'est pas le cas, il faut alors adjoindre une brique tierce située en amont de l'enchaînement pour assurer cette fonction.
L'autre méthode consiste à représenter le sens littéral d'une information au moyen d'un concept. La description structurée et formelle des concepts d'un domaine et des relations entretenues entre concepts constitue une ontologie (Grüber, 1993). L'ontologie se différencie ainsi de la métadonnée par la présence d'un langage formel d'assemblage dont les relations sont de diverses natures (inclusion, association, etc.).

L'établissement d'une relation entre briques demande en second lieu que soient connectées physiquement les entrées et les sorties des briques pour le transfert des données. (Pressman, 1992) a recensé six méthodes de transmission de données d'une brique P_A à une brique P_B

tel-00458265, version 1 - 19 Feb 2010

(Tableau 1). La méthode la plus simple, ou « Data Coupling », correspond à l'échange de données par jonction directe des entrées/sorties des briques. Cette méthode ne requiert pas de modification des briques, ni de mécanisme particulier pour l'échange des données. A l'opposé du « Data Coupling », le « Content Coupling » correspond à la situation où une brique P_A accède directement à l'espace de données d'une brique P_B (partage de mémoire, etc.). L'adoption de ce mode de transmission demande toutefois la mise en place d'un mécanisme particulier permettant à P_A d'accéder à l'espace de données de P_B. L'inconvénient d'une telle méthode est que le fonctionnement de P_B peut à tout moment être altéré par l'arrivée inopinée d'une donnée de P_A. Dans la mesure où P_A manipule directement les données de P_B, cette altération peut induire des comportements anormaux de P_B. Par exemple, l'incohérence de la valeur d'une donnée peut provoquer la mise en échec de P_B (plantage). Entre ces deux extrêmes, et parmi les quatre autres méthodes, la méthode « Stamp Coupling », par exemple, correspond à un cas intermédiaire. La différence au « Data Coupling » provient de l'incompatibilité des entrées/sorties. Dans ce cas, la mise en coïncidence entre briques requiert la mise en place d'une interface d'échange.

(Pressman, 1992) a classé ces six méthodes sur une échelle de couplage définie comme une « mesure de l'interdépendance relative des briques ». L'échelle comporte 3 niveaux (Faible, Moyen et Fort) auxquels est associé un degré de complexité de mise au point du mécanisme de transmission des données et de maîtrise des erreurs dues à la transmission, noté « rien ou ++ ». Dans la mesure où les programmes sont indépendants, le « Data Coupling » par exemple est qualifié de couplage faible. Le mécanisme de transmission à mettre en place est simple, à savoir la connexion directe des entrées/sorties des briques, et la maîtrise des erreurs aisée de par leur indépendance fonctionnelle et structurelle. La méthode « Stamp Coupling » correspond également à couplage faible, avec toutefois une source d'erreur possible liée au recours à une interface conçue spécialement pour relier les deux briques, notée ++. Le « Content Coupling » est qualifié de couplage fort et complexe. La mise au point d'une telle méthode est complexe, à l'exemple du partage d'une zone mémoire. La dépendance de P_B vis-à-vis de P_A sans contrôle rend la méthode peu fiable. Enfin, la méthode rend difficile l'interchangeabilité des programmes. Quelle que soit la construction, (Pressman, 1992) recommande l'adoption du niveau de couplage le plus faible possible.

Tableau 1 : Méthode de transmission des données d'un programme P_A à l'attention d'un programme P_B, et niveau de couplage de P_B vis-à-vis de P_A (Pressman, 1992 : 336).

Intitulé de la méthode	Description de la méthode de transmission	Niveau de Couplage
Data coupling	Transmission des données de P_A vers P_B en l'état.	Faible
Stamp coupling	Formatage des données de P_A au moyen d'une interface partagée par P_A et P_B.	Faible ++
Control coupling	Influence des données de P_A sur le fonctionnement de P_B (algorithme).	Moyen
External coupling	Formatage des données de P_A au moyen d'une interface définie indépendamment de P_A et P_B.	Moyen ++
Common coupling	Partage par P_A et P_B d'un espace de données global.	Fort
Content coupling	Accès direct de P_A à l'espace de données de P_B.	Fort ++

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

1.3 Trame d'analyse

Associer des programmes revient à manipuler une collection de briques au moyen d'un langage. Au sein du langage, Sabah (1988) distingue quatre niveaux de connaissances : les connaissances morphologiques des termes (forme canonique du terme, racines et terminaisons, emploi), les connaissances syntaxiques, qui expriment les relations entre les termes sous forme de règles, les connaissances sémantiques pour l'expression du sens littéral, et enfin les connaissances pragmatiques, qui permettent d'accéder aux aspects implicites du sens. En ce qui concerne les connaissances pragmatiques, nous n'avons pas trouvé de méthode formelle dans la littérature qui en permettrait l'explicitation. L'objet du chapitre B est de proposer une méthode en vue d'y accéder.

Dans le cadre de l'assemblage, au terme est associée la propriété morphologique de valence. La valence intervient au niveau syntaxique en spécifiant le nombre de termes qui peuvent être accolés « simultanément ». La collection des termes constitue un dictionnaire terminologique. Au niveau syntaxique, la littérature permet d'identifier des règles : la règle R1 définit l'ordre de juxtaposition des termes au moyen des relations série et parallèle (Klir et Valach, 1967) ; la règle R3 définit les modalités de composition des termes via la méthode de transmission (Pressman, 1992). A celles-ci, nous ajoutons une règle R2, dévolue à la mise en correspondance des entrées et des sorties des briques.

Au niveau sémantique, le traitement sémantique (S2) de R2 correspond à l'évaluation de la cohérence sémantique du rapprochement des termes au travers de l'utilisation de métadonnées (NISO, 2004) ou d'ontologies (Grüber, 1993). Ces méthodes permettent de mettre en coïncidence les entrées/sorties des termes en utilisant leur sens littéral. Ce faisant, ces méthodes ne considèrent pas le sens implicite des entrées/sorties. La décomposition des modalités d'assemblage selon les trois règles (R1, R2 et R3) conduit à expliciter le sens des règles R1 et R3, initialement implicites au langage. L'étude de R1 et R3 nous amène donc à traiter de la sémantique pragmatique induite par leur application. En ce qui concerne la sémantique-pragmatique (appelée pragmatique dans la suite du document), la pragmatique de R1 (notée S1) émerge au travers de la combinaison des relations série et parallèle, et s'exprime au moyen des notions d'action, d'interaction, de rétroaction et de parallélisme (Le Moigne, 1977). La pragmatique de R3 (notée S3) est caractérisée en utilisant la notion de niveau de couplage (Pressman, 1992).

Le Tableau 2 récapitule la trame d'analyse constituée des quatre niveaux de connaissances définis par Sabah (1988). Dans la suite du document, la trame d'analyse est appelée trame DRS (pour Dictionnaire, Règles syntaxiques et Sémantiques).

Tableau 2 : Trame d'analyse DRS des plateformes logicielles.

Dictionnaire	Déclaration des termes (valence)		
Syntaxe	R1 : interrelation (série, parallèle)	R2 : conjugaison des termes (mise en correspondance des E/S par l'usage de métadonnées ou d'ontologies)	R3 : modalité de composition (méthode de transmission)
Sémantique		S2 : Cohérence sémantique du rapprochement	
Pragmatique	S1 : Sémantique de l'interrelation (Action, interaction, rétroaction, parallélisme)		S3 : Sémantique de la composition (niveau de couplage)

tel-00458265, version 1 - 19 Feb 2010

2 Les plateformes logicielles

Dans une perspective de réutilisation, les informaticiens ont élaboré des supports génériques dévolus à la construction de plateformes logicielles. Ce chapitre présente les supports génériques MODCOM, OpenMI et VLE utilisés dans les sciences du vivant (SV). Leur description respecte la trame d'analyse DRS présentée au §1.3. De façon à faciliter la lecture du document, les phrases clés de l'analyse des plateformes, au crible de la trame d'analyse DRS, sont précédées du symbole « ☞ ». Ces phrases clés résument les éléments présentés plus tôt. Dans la suite du document, elles sont utilisées comme support de comparaison des plateformes logicielles.
Dans la mesure où le travail présenté s'appuie sur les concepts développés par Sabah (1988), nous adopterons la terminologie utilisée par cet auteur. Dans la suite du document, la brique ou le programme manipulé par le langage d'assemblage sera donc appelée « terme ».

2.1 The Modular simulation Component (MODCOM)

MODCOM est un support dédié à la construction de plateformes pour l'assemblage de programmes développés en agro-écologie (Hillyer, 2003). MODCOM permet la mise en relation de programmes numériques et dispose d'une bibliothèque de méthodes d'intégration numérique d'équations différentielles (Runge-Kutta et Euler). Son fonctionnement repose sur le standard industriel COM (Component Object Model) établi par Microsoft, définissant des modalités de connexion de composants logiciels (ActiveX, DLL, etc.).

Constitution du dictionnaire

MODCOM est un noyau central qui met en relation des programmes développés en utilisant un langage de l'environnement .Net (C#, ...). Il permet également le rattachement de composants logiciels, extérieurs à MODCOM, par adjonction d'une interface logicielle de liaison (Figure 05). Cette interface logicielle de liaison accommode le fonctionnement du composant avec celui de MODCOM et supporte l'exécution du programme inclus dans le composant.

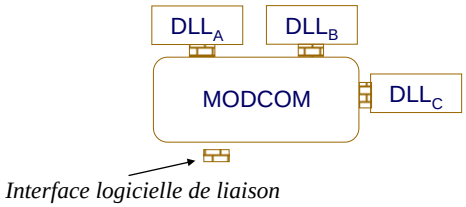


Figure 05 : Insertion de programmes dans MODCOM. Connexion de composants logiciels, transmis sous forme de DLL, à MODCOM par adjonction d'une interface logicielle de liaison.

☞ *Les termes du dictionnaire MODCOM sont soit des programmes insérés dans MODCOM, soit des programmes insérés dans les composants logiciels rattachés à MODCOM. La valence des programmes est égale à 1 quel que soit le mode d'insertion du programme.*

R1 et S1

MODCOM permet d'appeler en boucle une succession de composants ordonnés selon un ordre total (Figure 06). Il s'agit donc d'une construction séquentielle. A l'issue d'une boucle, l'intégration numérique s'effectue simultanément pour tous les composants. La Figure 06 montre un exemple de chaîne où le composant A est exécuté, puis le B, et enfin le C.
L'interrelation entre paires de programmes permise par MODCOM correspond à l'image 4 de la Figure 02.

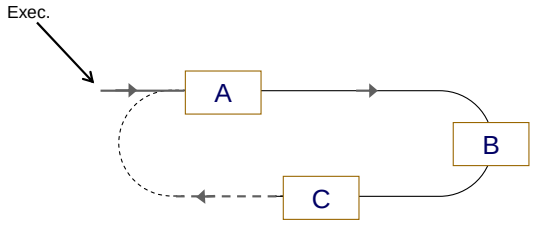


Figure 06 : Ordre d'appel des composants par MODCOM. Les traits continus indiquent l'ordre d'appel, et les traits pointillés la réitération.

☞ *En termes sémantique, R1 de MODCOM permet uniquement l'action.*

R2 et S2

MODCOM ne propose pas de mécanisme d'assistance de mise en coïncidence des entrées/sorties des programmes. L'opération doit être conduite par l'utilisateur.

R3 et S3

MODCOM assure la transmission des données entre programme. Pour ce faire, MODCOM dispose d'une mémoire globale dans laquelle sont stockées les données que chaque composant doit mettre à disposition des autres. Au cours de l'exécution, MODCOM se charge de récupérer puis de transmettre ces données aux composants demandeurs. Le mode opératoire correspond à l'External Coupling' (Tableau 1).

☞ *En termes sémantique, R3 de MODCOM implique un niveau de couplage Moyen++.*

2.2 Virtual Laboratory Environment (VLE)

VLE est un environnement de développement de programmes numériques. Il est conçu sur les bases théoriques de DEVS (Discrete Event System Specification), un formalisme modulaire permettant de considérer les processus discrets, continus et hydrides (Zeigler et al., 2000). D'après Amblard et Phan (2006), les méthodes de résolution des équations numériques proposées dans cet environnement permettent de faire évoluer le temps, l'espace et l'état des variables du programme de manière discrète et/ou continue. L'architecture du langage proposé par VLE a été élaborée dans le champ de l'électronique (Kim et Kim, 1998). La

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

tel-00458265, version 1 - 19 Feb 2010

terminologie utilisée pour sa description correspond donc au vocabulaire employé en électronique.

Constitution du dictionnaire

Le langage proposé par VLE (Ramat et Preux, 2003 ; Quesnel et al., 2009) manipule exclusivement les éléments au format DEVS, appelés processeurs DEVS. Un processeur DEVS est défini par un ensemble de variables d'état, un ensemble de variables d'entrée, un ensemble de variables de sortie, et une durée durant laquelle le processeur est en sommeil. La dynamique d'un processeur DEVS est explicitée au moyen de 4 fonctions : (i) la fonction de transition interne, qui permet la détermination des variables d'états, (ii) la fonction de transition externe, qui permet la prise en compte des informations transmises par un autre processeur DEVS à un instant donné (échange évènementiel entre processeurs), (iii) la fonction de gestion des conflits, qui décide de l'action à conduire dans le cas où des informations arrivent alors que celui-ci est en sommeil, et enfin (iv) la fonction de sortie, qui va transmettre des informations à l'attention d'autres processeurs. Ce formalisme permet d'appréhender un processeur DEVS comme une boîte noire multivalente constituée de ports d'entrée et de sortie multiples. Chaque port d'entrée correspond à une fonction de transition externe mise à disposition des autres composants DEVS. Les variables d'états d'un processeur peuvent par conséquent évoluer selon des rythmes différents suivant le port d'entrée activé.

L'insertion d'un programme dans VLE requiert une transformation préalable du programme en un processeur DEVS. Deux options sont possibles (Figure 07) : soit réécrire le programme au moyen du langage VLE, soit le rattacher à un processeur DEVS qui sert alors d'élément de liaison. Cette dernière option (i) nécessite toutefois l'adéquation de la logique de fonctionnement interne du programme avec celle d'un processeur DEVS, à l'exemple de l'intégration numérique par un programme externe, et (ii) demande l'adjonction d'une interface de liaison au programme afin de le relier à un processeur DEVS. En plus des quatre fonctions de gestion de la dynamique du processeur DEVS, cette interface requiert la définition d'autres fonctions demandant entre autres la description de l'état initial du processeur, l'initialisation de l'horloge interne au processeur, et la spécification du pas d'incrémentation de l'horloge pour la gestion du sommeil.

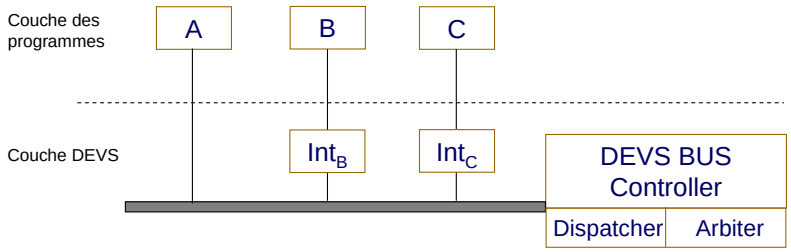


Figure 07 : Insertion de programmes au sein de la plateforme VLE. Le programme A correspond à un processeur DEVS directement implémenté au moyen du langage VLE. Les programmes B et C sont des programmes développés au moyen d'un autre langage. L'utilisation des interfaces Int_B et Int_C permet la « transformation » de B et C en processeurs DEVS. La mise en relation des processeurs DEVS s'effectue au moyen d'un bus piloté par un contrôleur (Kim et Kim, 1998).

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

☞ *Les termes du dictionnaire VLE correspondent soit à des programmes implémentés sous la forme de processeurs DEVS multivalents, soit à des programmes externes connectés à des processeurs DEVS de valence variable.*

R1 et S1

La mise en relation des processeurs DEVS s'effectue via un bus commun auquel tous sont connectés (Kim et Kim, 1998). Dans la mesure où le bus ne peut être utilisé que par un seul processeur à la fois, l'arbitrage des échanges est assuré par un contrôleur (Figure 07). Ce contrôleur permet également le fonctionnement en parallèle de processeurs DEVS par la diffusion simultanée d'une même information auprès de plusieurs processeurs (Quesnel et al., 2007).

L'utilisation du Bus, la multivalence des processeurs DEVS et le fonctionnement de chaque processeur selon un pas de temps propre permettent de construire des enchaînements de programmes qui varient suivant l'état de sommeil de chaque processeur et les informations qui interrompent le sommeil. Par construction, DEVS permet les dix-huit formes d'interrelations de la Figure 02 (zeigler, 1972).

La Figure 08 présente un enchaînement de processeurs DEVS, extrait de Quesnel et al. (2009). Dans cette figure, les processeurs sont représentés par des rectangles comportant des flèches pointant vers l'intérieur et d'autres vers l'extérieur, correspondant respectivement aux ports d'entrée et de sortie. Différentes configurations se présentent : les processeurs peuvent avoir un ou plusieurs ports d'entrée, et dans le cas où il y a plusieurs ports d'entrée, un port de sortie peut soit être commun à plusieurs ports d'entrée, soit être associé à chaque port d'entrée, auquel cas il y a autant de ports d'entrée que de ports de sortie. Les traits réunissant les ports d'entrée/sortie des processeurs correspondent aux enchaînements d'exécution des programmes. On y retrouve les formes d'interrelations que sont la relation d'ordre (image n°4, Figure 02), l'interaction (image n°7, Figure 02) et le parallélisme (image n°12, Figure 02).
L'exemple porte sur l'intervention de pompiers luttant contre des feux de forêt, déclenchés par un pyromane. Cet exemple est implémenté sous la forme d'un automate cellulaire. Dans ce graphe, le pyromane est représenté par le processeur 'Agent', et les forêts par les processeurs 'Forest11', 'Forest12', 'Forest13' et 'Forest14'. Le processeur 'Move' gère le déplacement de 'l'Agent' au sein de l'automate cellulaire. Les interrelations établies entre les processeurs Forest correspondent (i) au parallélisme, puisque la sortie d'un processeur Forest est connectée à l'entrée de plusieurs autres processeurs Forest, par exemple Forest22 alimente les processeurs Forest21 et Forest12, et (ii) à l'interaction, à l'exemple du processeur Forest21 qui alimente le processeur Forest 22. La mise en place de l'interrelation, du type action, entre 'l'Agent' et un processeur 'Forest' s'effectue en fonction de la position de 'l'Agent' au sein de l'automate cellulaire et de la présence ou non d'un processeur 'Forest'. Dans ce graphe, 'l'Agent' est connecté avec 'Forest12'.

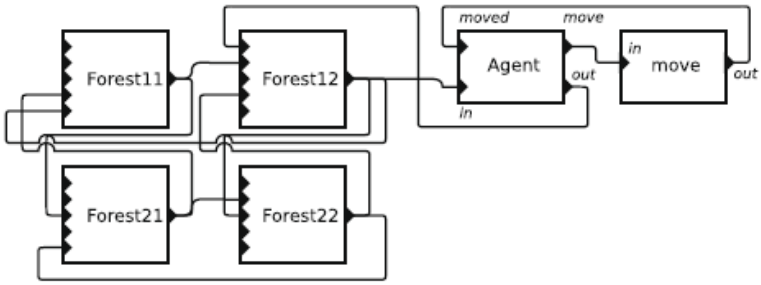


Figure 08 : Exemple d'enchaînement de processeurs DEVS (Quesnel et al., 2009).

☞ *En terme sémantique, R1 de DEVS donne accès aux différentes variantes sémantiques combinant l'action, l'interaction, la rétroaction et le parallélisme.*

R2 et S2

VLE ne propose pas de mécanisme d'assistance de mise en coïncidence des entrées/sorties des processeurs DEVS. L'opération doit être conduite par l'utilisateur.

R3 et S3

La transmission des informations entre deux processeurs DEVS s'effectue via le bus, qui sert uniquement de véhicule. Cette situation correspond au mode de transmission « Data Coupling ».

☞ *En terme sémantique, R3 de DEVS correspond à un niveau de couplage faible* (Tableau 1).

2.3 Open Modelling Interface (OpenMI)

OpenMI (http://www.openmi.org/) est initialement un standard d'échange de données environnementales entre systèmes hétérogènes, établi dans le cadre du programme européen LIFE (L'Instrument Financier pour l'Environnement). Ce standard est accompagné aujourd'hui d'un support générique de construction de plateforme, également appelé OpenMI.

Constitution du dictionnaire

Le rattachement d'un programme au support générique OpenMI s'effectue par adjonction d'une interface de liaison au programme pris en l'état (Gregersen et al., 2007). Ce mode opératoire ne nécessite donc pas la modification du code source du programme. OpenMi n'a aucune exigence particulière concernant la valence des programmes. Ceux-ci peuvent être aussi bien monovalents que multivalents.

☞ *Les termes du dictionnaire OpenMI sont des programmes mono ou multivalents.*

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

R1 et S1

Le standard OpenMI permet l'enchaînement de programmes selon une arborescence où chacun est à la fois serveur du programme appelant et client des programmes appelés (Moore et Tindall, 2005). Le parcours de cet arbre s'effectue tout d'abord en profondeur puis en largeur. La Figure 09 présente un enchaînement de quatre programmes. A est la racine de l'arbre, B est une branche qui supporte les feuilles C et D. L'exécution du programme A entraîne l'émission d'une requête, représentée par une flèche en trait continu, auprès de B (étape 1), puis B auprès de C (étape 2). L'octroi d'une réponse, représenté par une flèche en pointillé, de C à B (étape 3) permet l'envoi d'une requête auprès de D (étape 4). L'obtention de réponses de D vers B (étape 5) puis de B vers A (étape 6) permet de boucler la chaîne. Cette méthode permet de construire des arbres à géométrie variable : une branche peut échanger à plusieurs reprises avec plusieurs autres branches. Cette structure arborescente correspond à la mise en place de l'interaction entre paires de programmes (image n°7, Figure 02), du fait que le programme client attend des informations du serveur pour terminer son exécution (Gijsbers et al., 2005).

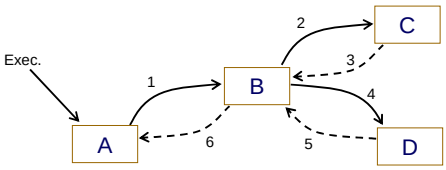


Figure 09 : Enchaînement en cascade permis par OpenMI. Chaque programme est serveur d'un précédent et client de suivants (Verweij et al., 2007).

☞ *En terme sémantique, R1 d'OpenMI permet de produire l'interaction.*

R2 et S2

Le standard OpenMI (Figure 10) est une métadonnée qui permet de caractériser les entrées/sorties des programmes (Gijsbers et al., 2005). La métadonnée comporte différentes classes : (i) le type d'information véhiculée par l'indication d'une description littérale de l'information, du type de données (tableaux, etc.), et de l'unité dans laquelle l'information est exprimée (What en Figure 10), (ii) la date ou la durée de l'information (When en Figure 10), (iii) son positionnement géographique (Where en Figure 10) et (iv) des modalités particulières de transformation (How en Figure 10).

☞ *L'accès à la sémantique de R2 d'OpenMI s'effectue par l'utilisation d'une métadonnée.*

R3 et S3

La transmission des informations entre deux programmes s'effectue via les interfaces de liaison des programmes à la plateforme. Cette situation correspond au mode de transmission « Stamp Coupling ».

☞ *En terme sémantique, R3 d'OpenMI correspond au niveau de couplage faible (Tableau 1).*

tel-00458265, version 1 - 19 Feb 2010

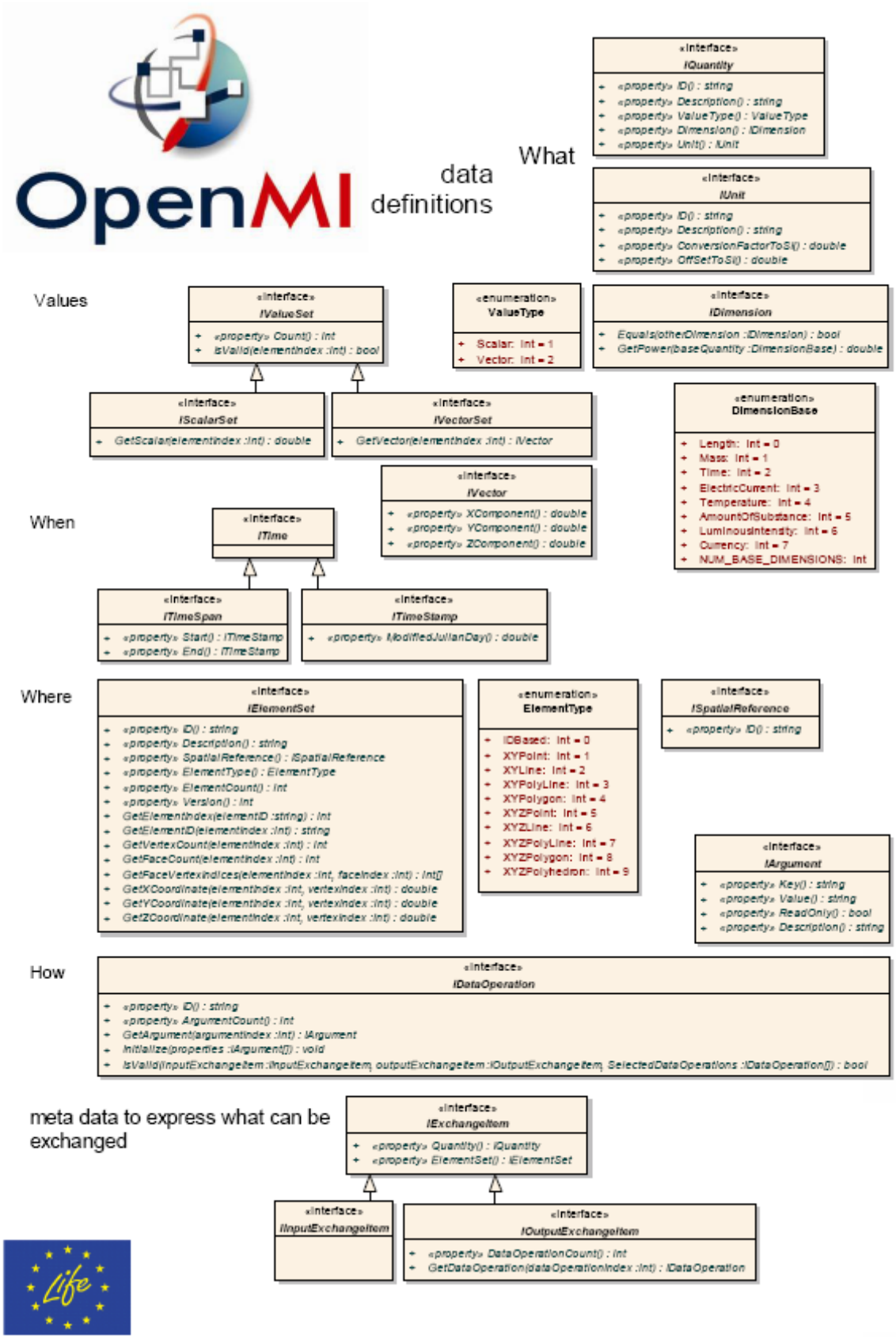


Figure 10 : Standard d'échange OpenMI version 1.0 (Gijsbers et al., 2005).

tel-00458265, version 1 - 19 Feb 2010

2.4 Récapitulatif des caractéristiques des supports génériques

Dictionnaire

MODCOM, VLE et OpenMI permettent d'appréhender l'insertion d'un programme sous la forme d'un rattachement à un support. Ce rattachement s'effectue par l'adjonction d'une interface de liaison qui permet de lever la dépendance de l'opération d'insertion vis-à-vis de la technologie nécessaire au fonctionnement externe du programme. L'interface permet par exemple d'accéder à des programmes situés sur des machines distantes via un réseau informatique.
Les trois supports proposés par MODCOM, VLE et OpenMI se distinguent toutefois par leur modalité de rattachement. Dans OpenMI, le rattachement du programme s'effectue en l'état, sans nécessiter de modification. Le programme est donc considéré comme indépendant du support générique. Par contre, l'assemblage peut nécessiter qu'un programme mette à disposition du programme suivant des données non disponibles en sortie. VLE et MODCOM assurent l'intégration numérique d'équations différentielles en plus de l'association de programmes. Le rattachement requiert donc que le code source du programme à rattacher soit structuré de façon à permettre à un programme extérieur, le support générique en l'occurrence, d'effectuer l'intégration numérique.
Le rattachement d'un programme peut par conséquent requérir deux natures d'interventions sur le code source. La première est l'adaptation des entrées/sorties du programme, que nous appelons typage dans ce document. C'est l'option retenue par OpenMI. La seconde opération est le reformatage, correspondant à une restructuration du code source. Cette opération est nécessaire pour le rattachement d'un programme à MODCOM ou à VLE, dans le cas par exemple où la structure du programme ne permet pas de déléguer l'intégration numérique au support générique.

☞ *Les dictionnaires des supports génériques MODCOM et VLE comportent des termes qui sont soit construits dans le langage de programmation de la plateforme (endogène), soit accolés au moyen d'une interface spécifique (exogène). Dans OpenMI, le dictionnaire est uniquement constitué de termes exogènes. La valence des termes de MODCOM est de 1 ; pour VLE et OpenMI, celle-ci peut prendre différentes valeurs. Pour VLE, la valeur varie selon l'origine du terme (endogène vs exogène). L'insertion d'un terme dans le dictionnaire d'OpenMI requiert au maximum un typage. Pour les dictionnaires de MODCOM et de VLE, elle peut nécessiter un typage et un reformatage.*

R1 et S1

Trois types de séquence d'appel sont employés dans les supports génériques. Chacun impose une valence aux programmes à rattacher, et donne accès à une ou plusieurs formes d'interrelation entre programmes. MODCOM permet de constituer une suite ordonnée de composants. En version originale, MODCOM n'autorise que les composants monovalents, et ne permet donc pas l'interaction entre composants. L'exécution de chaque programme est alors dirigée par MODCOM selon un mode centralisé.
OpenMI permet de constituer une arborescence de programmes. Celle-ci est établie selon un mode client/serveur, c'est-à-dire en faisant interagir les programmes se succédant. L'exécution

de l'arborescence est gérée par OpenMI de façon décentralisée : le programme situé en amont initie l'exécution de l'arborescence. L'arbre est alors parcouru en profondeur puis en largeur. Ce fonctionnement hiérarchisé confère à chaque programme la responsabilité d'exécution des programmes qui lui sont directement connectés, ainsi que de l'obtention des données nécessaires au programme exécuteur.
VLE permet de constituer des enchaînements complexes de type treillis orienté (Figure 08). Par définition, les processeurs DEVS sont multivalents. La communication au moyen d'un bus partagé par tous les processeurs autorise toutes les formes d'interrelation présentées en Figure 02 (parallélisme, etc.). La gestion de l'exécution de l'enchaînement est décentralisée dans la mesure où les programmes communiquent « directement » les uns avec les autres via le bus.

☞ *R1 diffère selon le support générique considéré : MODCOM permet uniquement l'action, OpenMI l'interaction, et VLE l'action, l'interaction, la rétroaction et le parallélisme.*

R2 et S2

Les supports MODCOM et VLE ont une vocation générique et ne proposent pas d'assistance à la mise en coïncidence des entrées/sorties entre les programmes. Le standard d'échange de données proposé par OpenMI met à disposition une métadonnée relative à des informations environnementales.

☞ *MODCOM et VLE ne disposent pas de R2. Cette opération est déléguée aux plateformes logicielles qui utilisent MODCOM et VLE comme langage d'assemblage. Quant à OpenMI, il propose pour R2 une métadonnée établie par les Sciences du Vivant (SV). La métadonnée comporte des propriétés relatives à l'information (What), au positionnement géographique (Where), à la temporalité (When) et son traitement (How).*

R3 et S3

Deux modes de transmission des données sont proposés par les supports génériques. VLE et OpenMI fonctionnent selon un mode décentralisé, où chaque programme assure l'échange des données. Cette option confère aux deux supports un niveau de couplage faible, en cohérence avec les recommandations de Pressman (1992). Dans MODCOM, par contre, la gestion centralisée de l'ensemble « exécution des programmes et transmission de données » confère à ce support un niveau « moyen++ ».

☞ *Pour S3, OpenMI et VLE présentent un niveau de couplage faible, et MODCOM Moyen++.*

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

Résumé

MODCOM et VLE proposent une fonctionnalité d'intégration numérique. MODCOM considère le temps, et VLE le temps, l'espace et l'état. L'opération numérique s'intercale entre deux exécutions de programmes. A ce titre, elle correspond à un programme tiers qui effectue des opérations numériques. Vis-à-vis de la trame d'analyse DRS, ce mode opératoire correspond à l'émission par les supports génériques de termes supplémentaires, qui complètent le Dictionnaire (Tableau 3).

Tableau 3 : Synthèse de description des supports génériques MODCOM, OpenMI et VLE.

		MODCOM	OpenMI	VLE
Dictionnaire	Valence du terme	1	n ≥ 1	n ≥ 1
	Origine du terme	Endogène/exogène	Exogène	Endogène/exogène
	Modalités d'insertion	Typage et reformatage si exogène	Typage	Typage et reformatage si exogène
	Terme dévolu à l'intégration numérique	Temps	Non défini	Temps, Espace, Etat
Sémantique	S1	Action	Interaction	Action, interaction, rétroaction, parallélisme
	S2	Non défini	Métadonnées	Non défini
	S3	Moyen++	Faible	Faible

3 Les plateformes agronomiques

Vis-à-vis de l'agronome, la plateforme recouvre un sens plus large que celui de l'assemblage proprement dit de programmes. A l'objectif final de produire des logiciels, la plateforme agronomique ajoute également la fonction de rassembler et de faire collaborer des équipes d'origine disciplinaire et géographique différentes. De ce fait, les chercheurs amenés à contribuer à la plateforme y amènent leurs méthodes, outils et concepts/paradigmes qu'il s'agit de faire coïncider. Les connaissances pragmatiques, inhérentes à ces méthodes, outils et concepts, sont donc importantes dans ces constructions logicielles.

A titre de « dispositif d'étude », nous avons choisi trois plateformes agronomiques, DSSAT (Jones et al., 2003), APES (Agricultural Production and Externalities Simulator ; Donatelli et al., 2010) et SEAMLESS-IF (van Ittersum et al., 2008). Dans le cadre de mon activité professionnelle, j'ai eu à utiliser DSSAT et ai participé au projet SEAMLESS, au sein duquel APES et SEAMLESS-IF ont été développés. Dans ce projet, j'ai contribué à la construction d'APES pour la partie sol. DSSAT occupe une place particulière parmi les plateformes agronomiques dans la mesure où l'histoire de sa construction et de ses évolutions se déroule sur plus de vingt ans. DSSAT résulte d'une construction informatique ad hoc. APES et SEAMLESS-IF, au contraire, reposent sur les supports génériques MODCOM et OpenMI respectivement. DSSAT et APES partagent l'objet de simuler le fonctionnement d'une parcelle agricole, et permettent de simuler l'effet cumulé de successions culturales sur une même parcelle et sur plusieurs années. Pour DSSAT, chaque type de culture (maïs, blé, etc.) est représenté par un programme spécifique, alors que pour APES, toutes les cultures sont représentées par un même programme, mais dont la valeur des paramètres de simulation afférents à la culture diffère. APES, par contre, permet de simuler les cultures associées, c'est-à-dire plusieurs cultures présentes au même moment sur la parcelle et en interaction, ce que ne permet pas DSSAT. Enfin, SEAMLESS-IF vise à associer APES à des programmes

développés par d'autres disciplines, économiques, socio-économiques et politiques, en l'occurrence. En lien avec l'assemblage de programmes pluridisciplinaires, SEAMLESS-IF considère plusieurs échelles spatiales, depuis la parcelle cultivée jusqu'au territoire national. Ces trois plateformes présentent l'intérêt d'une gradation en termes de proximité disciplinaire, forte dans DSSAT, APES vis-à-vis des sciences du vivant (SV), plus faible dans SEAMLESS-IF associant SV, économie et politique. Une première version des plateformes APES et SEAMLESS-IF a été diffusée en 2009.

L'objet de ce paragraphe est de décrire le fonctionnement des plateformes agronomiques et de produire une analyse sémantique sur la base de la trame d'analyse DRS proposée dans §1.3.

3.1 Les programmes agronomiques en question

Les programmes considérés dans les plateformes APES, DSSAT et SEAMLESS-IF rassemblent des équations numériques aux dérivées partielles. Le programme requiert en entrée des données de différentes natures – données observées (climat, etc.), constantes (physiques, mathématiques etc.) et paramètres – et produit des données de sortie. Ces programmes comportent deux types de paramètres : les paramètres de pilotage qui décident de l'algorithme, et les paramètres de calage qui sont insérés dans les équations numériques. Le rôle de ces derniers est d'ajuster les données de sortie produites par le programme en regard de données observées.

Le programme est caractérisé au moyen de deux notions sémantiques. La première est le sens littéral du programme. Le sens littéral du programme correspond à son objet, la simulation des flux hydriques dans le sol par exemple. La seconde notion est la fonction sémantique du programme, qui correspond à la relation de transformation d'un jeu de valeurs des données d'entrée en un jeu de valeurs de données de sortie (Ward et Zedan, 2007). Il s'agit donc d'une relation logico-mathématique. A titre d'exemple, l'interception de la lumière par un couvert végétal peut être simulée en regard de diverses formes de représentations du couvert (Martin et al, 2007b ; à partir des travaux de Clouvel et al., 2008a et Dauzat et al., 2008). Ainsi, changer la valeur des paramètres de calage du programme pour un jeu de données d'entrée modifie sa fonction sémantique, et non son sens littéral. Changer la valeur des paramètres de pilotage altère la fonction sémantique mais peut aussi modifier son sens littéral. Par exemple, soit un programme de simulation d'objets situés sur une parcelle agricole et un paramètre p qui, dans le cas où il est égal à 1, force la simulation à considérer le sol et la culture, et dans le cas où il est égal à 2, à n'impliquer que le sol. Suivant la valeur de p, respectivement 1 et 2, le sens littéral du programme est « simuler une culture agricole » ou « simuler le fonctionnement du sol d'une parcelle agricole », ce qui est différent sur le plan sémantique.

L'implémentation de ces programmes est assurée au moyen de deux familles de langages. La première, utilisée classiquement par les informaticiens, s'appuie sur une logique particulière (logique prédicative ou du 2ème ordre) et s'inscrit dans des paradigmes conceptuels de l'informatique (procédural vs objet, impératif vs déclaratif, etc.). La seconde famille est celle des langages dits « d'Aide à la Modélisation » (LAM), comme Stella® ou SIMILE (Muetzelfeldt et Massheder, 2003) par exemple. Les LAM sont élaborés à partir d'un langage de la première famille (surcouche), mais selon une école de pensée particulière à l'exemple de la théorie des systèmes (von Bertalanffy, 2002). Le choix du langage définit donc les limites de formulation du programme, et dans le cas des LAM, impose en sus un mode de pensée.

tel-00458265, version 1 - 19 Feb 2010

☞ *Un terme correspond à un programme numérique aux dérivées partielles. Ce programme est caractérisé (i) par un sens littéral, c'est à dire l'objet naturel qu'il représente, et (ii) par une fonction sémantique. La fonction sémantique correspond à la relation qui transforme un jeu de données d'entrée en un jeu de données de sortie. La formulation de la fonction sémantique s'effectue au moyen d'un langage redevable d'un mode de pensée de la discipline informatique (langage et paradigmes conceptuels), voire d'une autre école de pensée (théorie des systèmes, système multi-agents, etc.) dans le cas de l'utilisation d'un LAM.*

3.2 Description des plateformes agronomiques choisies

3.2.1 Une construction ad hoc, la plateforme DSSAT

La plateforme DSSAT (Jones et al., 2003) est la première plateforme élaborée en agronomie. Son développement a été initié dans le cadre du projet IBSNAT (1993) dont l'objectif était de disposer d'un support de transfert de technologies dans le domaine du fonctionnement des cultures. L'ambition de DSSAT était de produire un outil capable de réunir des programmes de simulation des cultures existants, et ceux en cours de développement, afin d'accéder aux propriétés de succession de cultures. De nombreuses équipes ont participé, et participent encore au projet, parmi lesquelles l'USDA/ARS, les universités du Michigan, de Floride, de Géorgie et d'Hawaii et l'IFDC aux Etats-Unis, l'université de Guelph au Canada, le CSIRO en Australie, l'école d'agriculture d'Edinbourg en Angleterre, le CIAT en Colombie, l'ICRISAT en Inde etc.

3.2.1.1 Les versions successives

La première version de la plateforme diffusée en 1989 (v2.1) rassemblait les programmes numériques de simulation CERES pour le maïs (Jones et Kiniry, 1986) et le blé (Ritchie et Otter, 1985), SOYGRO pour le soja (Wilkerson et al., 1983) et PNUTGRO pour l'arachide (Boote et al., 1986). En termes de fonctionnalité, la plateforme permettait (i) de manipuler chaque programme au moyen d'une interface commune, et (ii) de partager les outils de gestion des données (saisie des données d'entrée, analyse des données produites, etc.).
Tous les programmes « culture » étaient développés par des agronomes au moyen du langage FORTRAN et fonctionnaient sur le système d'exploitation MS-DOS. Les programmes différaient toutefois sur le jeu de données requis en entrée, et la façon dont les données d'entrée et de sortie étaient agencées au sein des fichiers. Pour accéder aux fonctionnalités de DSSAT (interface commune de visualisation des données de sortie produites par les programmes, etc.), une arborescence commune des fichiers et une organisation des données au sein des fichiers ont été définies. La construction de la plateforme a requis (i) le développement de l'interface d'utilisation de la plateforme (shell), produit au moyen du langage C, et (ii) l'adaptation au sein des programmes de la partie des codes sources ouvrant l'accès aux fichiers de données. L'insertion d'un programme dans la plateforme s'effectuait par la simple copie du programme en version exécutable dans un répertoire défini.

La version suivante (v3.0) de la plateforme a été diffusée en 1994 (Tsuji, 1994). Cette version, développée à partir de la précédente v2.1, (i) étend les fonctionnalités initiales à de nouveaux programmes de simulation de cultures, portant à douze le nombre de cultures prises

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

tel-00458265, version 1 - 19 Feb 2010

en compte, et (ii) comporte de nouvelles applications permettant par exemple de construire des jeux de données climatiques sur la base de prévisions. Afin de faciliter la gestion des données entre les programmes, un standard de dénomination et de description des données au sein des fichiers a été défini (Jones, 1994). Le standard de dénomination correspond à une codification des types de donnée (variété cultivée, etc.). Le standard de description correspond à une métadonnée. La métadonnée comporte le nom de la variable formulé dans son sens littéral, le code associé à la variable dans les fichiers d'entrée/sortie, l'abréviation de la variable à utiliser en entête des fichiers de sortie, et le format d'écriture de la donnée. Le format indique le nombre de caractères 'Espace' qui précède la donnée, le type de la donnée (Character = C, Real = R, Integer = I), le nombre de caractères de sortie de la variable, et, dans le cas où la donnée est de type réel, le nombre de décimales. Par exemple, le code associé à la variable 'Irrigation level' est 'LNIR', son abréviation pour les fichiers de sortie 'MI' et son format '0 I 2'.
Cette version ouvre également une nouvelle fonctionnalité, relative à la succession de cultures sur un même site dans l'objectif de simuler l'évolution de la fertilité du sol au cours du temps et selon les successions adoptées. Cette fonctionnalité impose à un programme quelconque que l'état du sol en début de simulation corresponde à celui obtenu en fin de simulation par le programme exécuté précédemment. Si la réalisation parait simple, l'usage a montré deux failles. La première est liée à l'existence de données manquantes dans les sorties de certains programmes en regard des données d'entrée nécessaires pour l'exécution du programme qui le succède dans l'enchaînement. L'autre faille est relative aux différences de valeurs de données, de programmes différents, ayant pourtant un même nom et représentant le même phénomène biophysique (Jones, 2003).

Les modifications apportées à la version 3.5 diffusée en 1998 ont eu pour objet d'apporter une solution temporaire aux failles soulevées par la succession de cultures. La solution proposée dans cette version a consisté en la définition d'un jeu de données d'entrée et de sortie minimum commun à tous les programmes en place (Hunt, 1998 ; Jones, 1998).

Le développement de DSSAT est aujourd'hui pris en charge par le consortium ICASA (International Consortium for Agricultural Systems Applications ; http://www.icasa.net). Dans la version 4.0 (Hoogenboom, 2003), d'autres programmes de simulation de culture ont été rajoutés, portant à 20 le nombre total de cultures considérées par cette plateforme. Cette version, diffusée en 2003, apporte une solution définitive aux failles soulevées par la succession de cultures. Elle consiste à adopter (i) une structure et un mode de fonctionnement commun à tous les programmes (Porter, 2000 ; Jones, 2001) ainsi que (ii) un sous-programme sol unique pour tous les programmes (Gijsman, 2002 ; Jones, 2003). La Figure 11 montre la structure commune adoptée.
Concernant la structure, les programmes ont été découpés en modules selon deux niveaux emboités. Les modules du premier niveau, appelés modules primaires, correspondent aux thématiques disciplinaires générales considérées par les programmes à insérer dans la plateforme. Les modules du second niveau, que j'appellerai modules secondaires, correspondent aux sous-thématiques disciplinaires. Le module primaire 'Soil' par exemple est décomposé en modules secondaires 'Soil temperature', 'Soil water' et 'Soil Nitrogen & carbon', qui comportent respectivement les sous-programmes relatifs à la température du sol, aux flux hydriques, d'azote et de carbone dans le sol. Les modules secondaires comportent donc des fragments du code source du programme original, et en particulier les équations numériques. Chaque module est constitué de 4 fonctions (Initialisation, calcul de la variation des variables d'état, intégration des variables d'état, et mémorisation des variables dans des

tel-00458265, version 1 - 19 Feb 2010

fichiers de sortie) identifiées par Reynolds and Acock (1997) dans le cadre de la proposition d'une architecture modulaire générique des modèles de simulation de culture.
Concernant le fonctionnement du programme, les modules primaires comportent la séquence d'appel des modules secondaires, et le 'Land Unit Model' (figure 11) définit celle des modules primaires. Pour chaque pas de temps du programme principal (Main Program en figure 11), la routine principale exécute trois fois le Land Unit Model qui se charge alors de la séquence d'appel des modules primaires. La première fois, elle appelle la fonction de calcul de variation des variables d'état de tous les modules, la seconde fois la fonction d'intégration et la dernière fois la fonction de mémorisation. L'échange des données entre les modules s'effectue via des fichiers de données. Chaque module est par conséquent autonome dans son fonctionnement. Un tel mode opératoire autorise l'échange d'un module par un autre.
Pour un même jeu de données d'entrée, les deux versions du même programme, initial et doté du module sol commun, produisent des jeux de données de sortie de valeurs sensiblement différentes. Afin de respecter la fonction sémantique, les paramètres des équations numériques de tous les programmes de simulation ont du être recalculés. Un outil d'analyse de sensibilité a également été introduit, afin d'évaluer la stabilité des programmes vis-à-vis des nouvelles valeurs de paramètre attribuées.

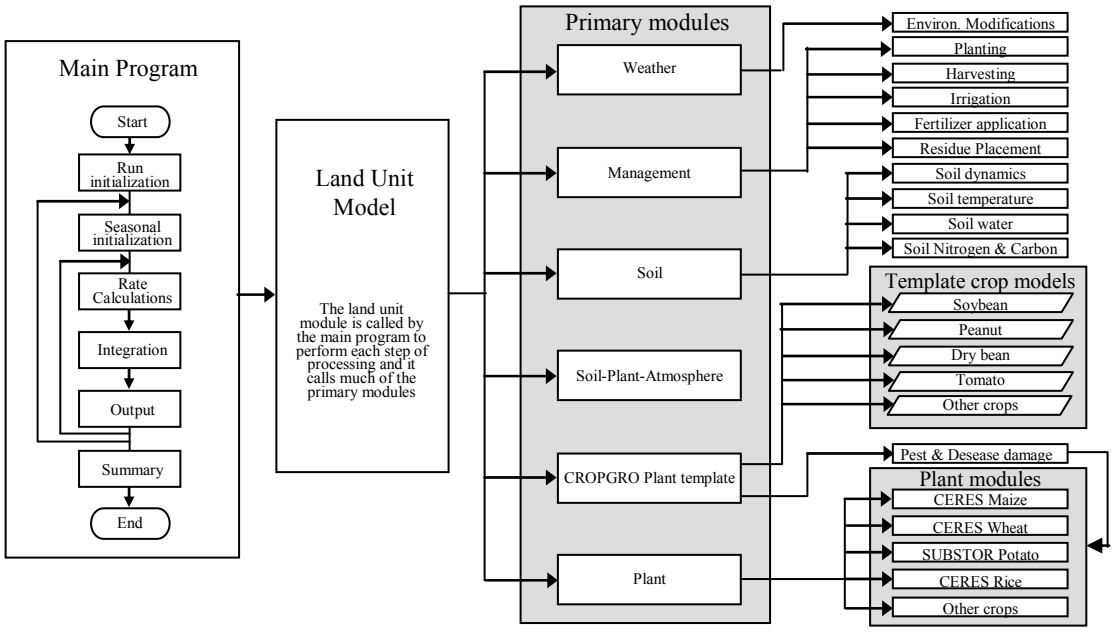


Figure 11 : Architecture d'un programme de simulation d'une culture dans DSSAT (Jones, 2003).

Concernant enfin la mise à jour « technique » de la plateforme en regard de l'avancée des technologies informatiques, les opérations ont consisté (i) à la migration de la plateforme depuis le système d'exploitation MS-DOS vers celui de WINDOWS, (ii) et à l'implémentation d'une nouvelle méthode de gestion des données. De façon à mémoriser les conditions d'obtention des données observées et simulées, une métadonnée a été adjointe à chaque jeu de données (McNair et al., 2004). Cette métadonnée comporte des informations concernant (i) le propriétaire du jeu de données, (ii) l'essai agronomique mis en place (dates de début et de fin, localisation géographique, et culture considérée), (iii) les données mesurées (climat, variété cultivée, la densité de semis, les opérations culturales produites telles que les irrigations et fertilisations, les données mesurées), et (iv) le fichier de données (format,

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

références bibliographiques si données publiées). De fait, le passage de la v3.5 à la v4 .0 a demandé la réécriture du code source (i) de la plateforme, (ii) de tous les programmes de simulation et (iii) de toutes les applications. Elle a été conduite par les équipes d'agronomes en partenariat avec un service informatique.

3.2.1.2 DSSAT dans la trame d'analyse DRS

Depuis la première version diffusée, l'évolution de l'architecture logicielle de DSSAT s'est effectuée au coup par coup en fonction de l'histoire des fonctionnalités requises et des problèmes rencontrés. Initialement conçue et développée par des agronomes, l'émergence de nouvelles technologies informatiques a conduit les concepteurs à recourir aux services d'informaticiens pour effectuer la mise à jour de la plateforme. Celle-ci a demandé un exercice de réécriture des programmes et des applications, et entrainé la conception d'une nouvelle interface d'utilisation. L'objet de la version 2.1 est de réunir les programmes conçus indépendamment au sein d'une structure d'accueil partagée. Dans les versions suivantes de DSSAT, la construction de chaînes de programmes a été permise en dotant le support DSSAT d'un langage d'assemblage des programmes. L'évolution de la plateforme est revisitée dans ce paragraphe selon la trame d'analyse DRS décrite au §1.3.

Le dictionnaire

Le nombre de programmes inclus dans DSSAT a augmenté dans le temps. De cinq dans la version 2.1, il est passé à 12 dans la version 3.0, à 16 dans la version 3.5 et à 20 dans la version 4.0. La méthode d'insertion d'un programme dans la plateforme a évolué entre les versions initiale (2.1) et actuelle (4.0). Dans la version 2.1, les données ne sont pas partagées par les programmes. L'insertion consiste donc à incorporer le programme en l'état, modulo la mise en conformité des données d'entrée/sortie, d'après un standard défini pour la plateforme. Ce mode opératoire correspond au typage (définition donnée au § 2.4). Dans la version 3.0, l'insertion d'une nouvelle fonctionnalité, à savoir la mise en relation des programmes pour la succession culturale, implique la nécessité d'échanges de données entre programmes. Un standard de dénomination des données a été défini au niveau de la plateforme. Ce standard correspond à la définition d'un vocabulaire de données, dont le sens littéral serait partagé par tous. Cette approche présuppose que la spécification d'un vocabulaire de référence soit suffisante pour permettre la cohérence de l'échange de données entre programmes. Le typage des programmes a été revisité de façon à permettre la mise en correspondance des entrées/sorties des programmes vis-à-vis de ce standard.

Dans la version 3.5, un jeu de données commun aux programmes est adopté. Le typage s'effectue alors en regard des autres programmes.

Dans la version 4.0, une architecture logicielle commune à tous les programmes est adoptée. L'architecture est constituée (i) d'un ensemble de modules différenciés par champs disciplinaires et agencés selon une arborescence à deux niveaux d'inclusion, et (ii) d'une interface de manipulation des modules commune. L'interface comporte les quatre fonctions d'initialisation, calcul de la variation des variables d'état, intégration des variables d'état, et mémorisation des variables dans des fichiers de sortie. Ce découpage conférant une valence de 4 (quadrivalence) au programme correspond à un reformatage du code source. L'imposition d'un module sol commun contraint à restructurer les entrées/sorties et

correspond à un typage. L'insertion d'un programme requiert désormais le typage et le reformatage du programme selon un schéma imposé.

☞ *Dans DSSAT, un terme correspond à un programme. Le nombre de termes augmente entre la version 2.1 et la version 4.0, passant de cinq à vingt. Dans la version 2.1, le typage du terme s'effectue d'après l'organisation des fichiers de données dans la plateforme. Dans les versions 3.0 et 3.5, il s'effectue respectivement selon un standard d'échange de données et selon les autres termes. Dans la version 4.0, l'insertion d'un terme requiert (i) un reformatage selon une architecture imposée, (ii) un typage, et (iii) l'échange d'une partie du terme (module sol) au profit d'une partie commune. Cette architecture implique le changement de la valence du terme, de 1 à 4 (Tableau 4).*

R1 et S1

Dans la version 2.1, l'assemblage des programmes pour la succession des cultures n'est pas prévu. Dans les versions 3.0, 3.5 et 4.0 les programmes sont exécutés les uns après les autres, sans permettre d'autre forme d'interrelation. En regard des formes d'interrelations établies par Klir et Valach (1967), ce mode opératoire correspond à la mise en place de la forme d'interrelation 4 (Figure 02) entre les programmes, à savoir l'action d'un programme sur l'autre sans feedback. La séquence forme un ordre total.

Dans la version 4.0, l'exécution interne d'un programme correspond au pilotage de l'arborescence des modules par une routine principale. Pour chaque pas de temps, la routine principale effectue trois séquences d'appel des modules. Chaque séquence s'effectue (i) selon un parcours vertical de l'arborescence, de la routine principale vers les modules primaires puis vers les modules secondaires, et (ii) selon un parcours horizontal entre les modules secondaires, au niveau des modules primaires, et entre modules primaires au niveau de la routine principale. En regard des formes d'interrelations établies par Klir et Valach (1967), ce mode opératoire correspond à la mise en place (i) de l'action d'une brique sur l'autre (image 4, Figure 02), et (ii) de la rétroaction (image 2, Figure 02) pour l'exécution des trois séquences d'appel des modules.

☞ *La version 2.1 correspond à l'établissement d'un dictionnaire de termes. Pour les versions 3.0, 3.5 et 4.0, S1 est l'action. Pour la version 4.0, l'exécution des trois séquences correspond à la mise en place d'une rétroaction interne au terme (Tableau 4).*

R2 et S2

Pour des raisons pratiques, la réunion des programmes au sein du même support dans v2.1 a demandé la définition d'une structuration des fichiers commune aux différents programmes. La nécessité de faire communiquer les programmes dans la version 3.0 a contraint à passer d'une structure commune de fichiers à un jeu de données partagé. Pour ce faire, un standard de dénomination et d'organisation des données au sein des fichiers a été défini (Jones, 1994). Ce standard, correspondant à une métadonnée, a évolué selon les versions en fonction des besoins.

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

☞ *La R2 des versions 3.0, 3.5 et 4.0 est une métadonnée (Tableau 4).*

R3 et S3

Les diverses évolutions de la plateforme n'ont pas remis en question le mode de transmission des données. Que ce soit entre programmes ou entre modules, celui-ci s'effectue via les fichiers de données. DSSAT opère donc suivant la méthode « Common coupling » (Tableau 1). En outre, la méthode adoptée ne permet pas d'assurer que les données, recueillies par un module ou un programme, sont cohérentes en regard de l'exécution en cours. Il peut donc se produire que la donnée soit modifiée par un programme tiers (interface de manipulation de données de DSSAT ou programmes extérieure à la plateforme agronomique), et que sa valeur soit incohérente (pas de garde-fou).

☞ *En terme sémantique, la R3 des versions 3.0, 3.5 et 4.0 de DSSAT correspond au niveau de couplage Fort (Tableau 4).*

Question soulevée par l'interchangeabilité

Dans les versions 3.0 et 3.5, le problème d'échange de données du sol n'est pas résolu. La solution adoptée dans la version 4.0 a consisté à imposer à tous les programmes un sous-programme sol identique. Le fait que le module sol commun ne reproduise pas les mêmes données de sorties pour un même jeu d'entrée que les modules sols initiaux implique que la fonction sémantique du programme n'est pas préservée.

☞ *Dans la version 4.0, l'échange de la partie sol du terme altère sa fonction sémantique. Les termes constitutifs du dictionnaire de la v4.0 sont par conséquent différents des termes originaux.*

Tableau 4 : Evolution des versions successives de DSSAT selon la trame d'analyse DRS.

		V 2.1	V 3.0	V 3.5	V 4.0
Dictionnaire	Nb de termes	5	12	16	20
	Valence du terme	1	1	1	4
	Origine du terme	Exogène	Exogène	Exogène	Endogène
	Modalités d'insertion	Typage / Organisation plateforme	Typage / standard	Typage / autres termes	Typage + Reformatage + partie commune (sol)
	Terme dévolu à l'intégration numérique	Non	Non	Non	Non
Sémantique	S1	Non défini	Action	Action	Action, rétroaction interne au terme
	S2	Non défini	Métadonnées	Métadonnées	Métadonnées
	S3	Non défini	Fort	Fort	Fort

tel-00458265, version 1 - 19 Feb 2010

3.2.2 APES, instance de MODCOM

Dans la mesure où APES utilise le support MODCOM décrit plus haut en § 2.1, la description présentée ici s'attache à mettre en évidence les solutions techniques adoptées pour construire APES. La plateforme agronomique APES (http://www.apesimulator.it), développée dans le cadre du projet européen SEAMLESS (2005-2009), permet de reproduire le fonctionnement d'une parcelle cultivée (Donatelli et al., 2006). Le principe prévalant à la construction d'APES est la réutilisation de constructions existantes. Le projet réunit de nombreuses équipes de recherche européennes, parmi lesquelles le CRA-ISCI, l'université de Milan (UNIMI), l'université catholique du sacré cœur (UNICATT) en Italie, l'INRA-EA, l'IRD, le CIRAD et l'UMR System pour la France, l'université de Wageningen pour les Pays-Bas, etc. Les équipes ont été retenues sur la base de leur reconnaissance scientifique dans différentes thématiques biophysiques (hydrologie du sol, physiologie de la plante, etc.) et des programmes qu'elles ont développés. L'hypothèse sous-jacente au projet APES est que l'assemblage de programmes unitaires, élaborés indépendamment, permet de construire un programme de simulation du fonctionnement d'une parcelle cultivée. Dans APES, ces programmes unitaires sont appelés composants logiciels. La modalité d'insertion du composant logiciel dans MODCOM est le rattachement par adjonction d'une interface logicielle de liaison (Figure 05).

Constitution du dictionnaire

Dans l'objectif de partager des outils (éditeurs, etc.) et des méthodes de développement, la cellule de coordination du projet APES a (i) défini une structure d'organisation du code source du composant et de l'interface logicielle de connexion à MODCOM, au moyen de patrons de conception (Gamma et al., 2001), et a (ii) demandé aux partenaires de réorganiser leur programme selon cette structure.
En ce qui concerne les composants logiciels, la structure a été établie à partir de trois patrons de conception. Le patron structurel « Façade » permet de définir une signature identique pour toutes les méthodes. Son adoption permet de manipuler n'importe quelle méthode sans se soucier des données spécifiques requises par celle-ci. Le patron structurel « Composite » permet la construction d'arborescences de méthodes en cascade, et de manipuler l'arborescence comme s'il s'agissait d'une seule méthode. Le patron comportemental « Stratégie » consiste à définir une mécanique unique d'encapsulation de l'algorithme au sein des méthodes, de façon à rendre les méthodes interchangeables et de les permuter dynamiquement.
Concernant l'interface logicielle, la structure a été élaborée à partir du patron de conception structurel « Adaptateur » qui permet d'abstraire le fonctionnement propre de chaque composant, et permet ainsi à la plateforme de manipuler tous les composants au moyen d'un langage unique.
La combinaison des patrons de conception retenue impose une répartition du code source du programme dans le composant et dans l'interface logicielle. En premier, toutes les variables sont définies dans l'interface. Ces variables peuvent être de type entier, réel ou chaîne de caractères. En second lieu, les équations numériques sont situées au sein des méthodes (stratégies), qui sont regroupées dans le composant. Le rôle de l'interface logicielle est d'exécuter les méthodes, implémentées dans le composant, qui modifient les données contenues dans l'interface. Cette structuration du code source est redevable d'un programme de type procédural pour lequel (i) les données sont localisées dans une mémoire globale et (ii)

les procédures, correspondant aux méthodes du composant, sont appliquées tour à tour sur les données.

L'utilisation combinée de MODCOM et de la structure imposée implique que les opérations d'intégration numérique des variables, préalablement effectuées au sein des programmes, soient externalisées au profit de MODCOM. Cette externalisation, nécessitant l'incorporation des variables à intégrer dans le jeu d'entrées/sorties du composant, correspond à un typage. Ainsi, l'insertion d'un programme dans APES demande l'opération de reformatage du code source selon la structure de patrons de conception, ainsi que celle de typage.

Deux versions d'APES ont été produites. La première, sortie en 2006, est présentée en Figure 12. Dans cette version, onze composants sont insérés dans la plateforme (Figure 12a). Certains fournissent des données en entrée, tels que les composants 'Weather' et 'Management' qui informent respectivement des conditions climatiques et des interventions techniques sur la parcelle (irrigation, fertilisation, etc.). Les autres composants simulent des phénomènes biophysiques (hydrologie du sol, etc.).

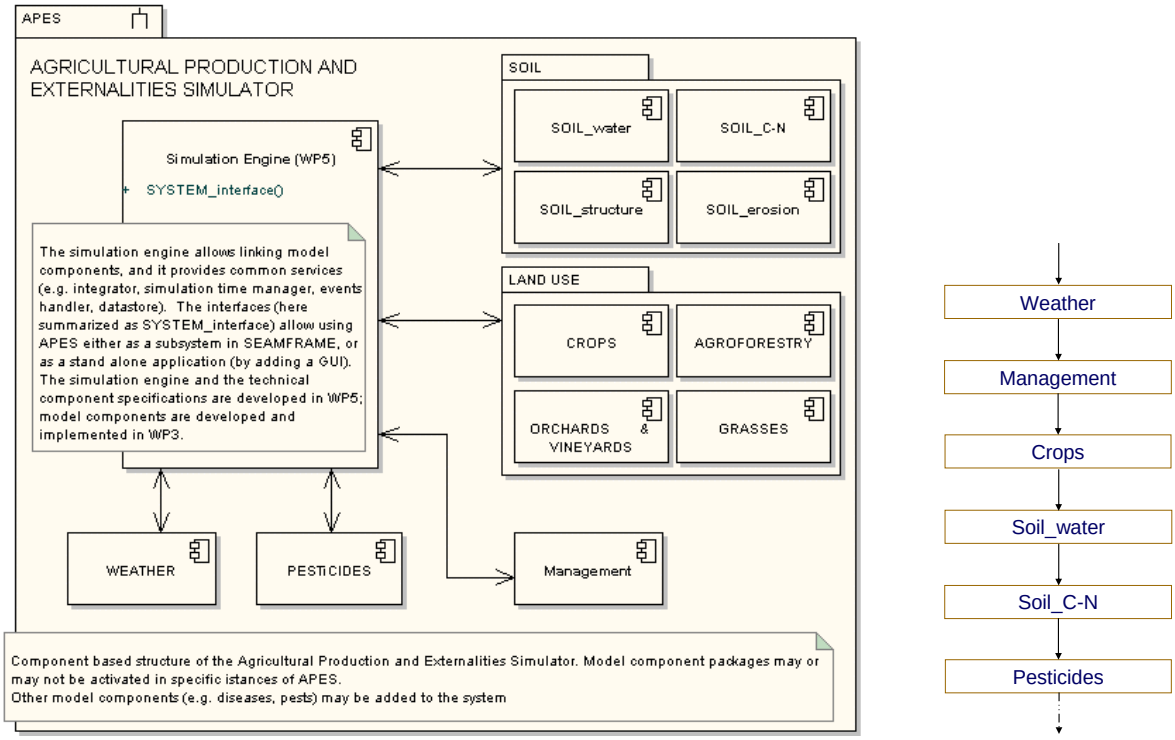


Figure 12 : Etat de la plateforme APES en 2006. (a) Composants logiciels (Donatelli et al., 2005). (b) exemple de cycle d'exécution.

La deuxième version, sortie en 2008, est présentée en Figure 13. La Figure 13a montre que le nombre de composants insérés dans la plateforme a doublé par rapport à celui de la version précédente, de onze à vingt-trois composants. Cette augmentation du nombre de composants résulte tout d'abord d'un enrichissement de la plateforme par intégration de composants supplémentaires. Parmi ces nouveaux composants, certains correspondent à la simulation de nouveaux phénomènes biophysiques, à l'exemple de 'Tree' pour le fonctionnement d'une parcelle agro-forestière (association de cultures et d'arbres). D'autres constituent des

tel-00458265, version 1 - 19 Feb 2010

doublons en termes de phénomènes simulés, à l'exemple de « Soilwater » et de « Soilwater2 ».

L'augmentation du nombre de composants résulte également de la scission de composants en plusieurs composants élémentaires dans le but de partager des sous-programmes avec d'autres composants (réutilisation). Par exemple 'Soil_water' a été désagrégé sous la forme de trois composants indépendants afin de mettre en commun avec 'Soilwater2' le sous-programme 'Soil température'. Le composant 'Crop' a également été scindé en deux : 'Crop' correspond à la partie aérienne de la culture et 'Root_crop' à son système racinaire.
L'augmentation du nombre de composants s'explique enfin par l'apparition de composants dédiés à la gestion des ressources (eau, lumière, etc.) partagées par plusieurs composants. Par exemple, l'insertion dans un même enchaînement de 'Crop' et de 'Tree' nécessite de partager les ressources mises à disposition par 'Weather', pour l'accès à la lumière d'après l'espace aérien occupé par chacun, et par 'Soilwater', pour l'accès à l'eau disponible dans le sol d'après l'espace souterrain occupé par chaque système racinaire. Ces répartitions sont effectuées respectivement par les composants 'Light interception' et 'Water arbitration', développés à partir de sous-programmes du composant 'Crops' de 2006.

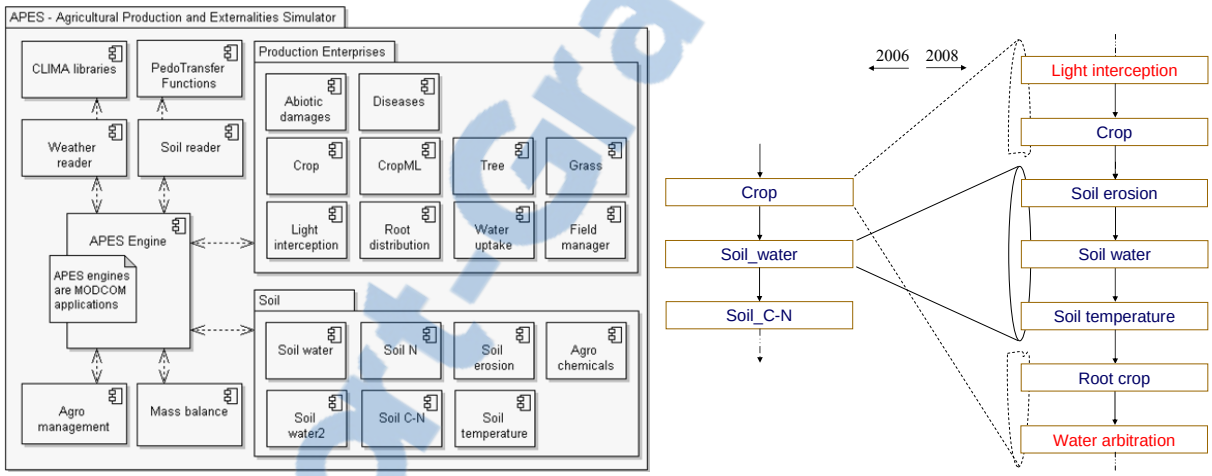


Figure 13 : Etat de la plateforme APES en 2008. (a) composants logiciels d'après (Donatelli, 2010). (b)Evolutions portées sur le cycle d'exécution de la Figure 12b. Les composants écrits en rouge sont dédiés au partage de ressources.

☞ *Les termes du dictionnaire APES sont les composants logiciels. L'insertion d'un terme requiert les opérations de typage et de reformatage. Pour le reformatage, la combinaison de patrons de conception adoptée correspond au paradigme de programmation procédural. L'accroissement du nombre de termes dans la version 2008 correspond (i) à l'ajout de programmes supplémentaires, (ii) à la décomposition des termes de la version 2006, afin de partager les sous-programmes avec les nouveaux termes, et (iii) à l'ajout de programmes dédiés à la gestion de ressources partagées par les termes (Tableau 5).*

R1 et S1

APES utilise la méthode de mise en interrelation des composants proposée par MODCOM. Un exemple d'enchaînement de composants de la version 2006 est présenté en Figure 12b.

tel-00458265, version 1 - 19 Feb 2010

Dans l'objectif de reproduire le fonctionnement d'une parcelle cultivée, l'enchaînement est constitué de 'Weather' et 'Management', qui informent du climat et des interventions techniques, 'Crops', qui calcule l'état de la culture, 'Soil_water', qui calcule l'état hydrique du sol, 'Soil_C-N', qui calcule les déplacements d'azote et de carbone dans le sol, et enfin 'Pesticides', qui calcule les déplacements des molécules chimiques (produits phytosanitaires) dans le sol.

☞ *S1 d'APES hérite de MODCOM et correspond donc à l'action (Tableau 5).*

R2 et S2

MODCOM ne propose pas de méthode d'assistance à la mise en correspondance des entrées/sorties entre programmes. Pour cette opération dans APES, la cellule de coordination a opté pour la technique des métadonnées. La métadonnée comporte six informations : le nom de la donnée, une description littérale du contenu, l'unité d'expression de la donnée, et les valeurs minimale, maximale et par défaut (Athanasiadis et al., 2006 ; Rizzoli et al., 2008). La mise en correspondance consiste à vérifier que les informations entre les entrées et les sorties à associer concordent.

La Figure 13b présente l'évolution de la partie de l'enchaînement, présenté en Figure 12b, relative au segment ['Crops' - 'Soil_water'] mise en place dans la version 2008. La Figure montre que si les trois composants issus de 'Soil_water' de 2006 se succèdent, ceux issus du découpage de 'Crops' ne sont pas disposés selon cette configuration : les composants issus de 'Crops' qui concernent la partie aérienne de la culture sont situés en amont du trio sol (érosion water et température), et ceux qui concernent la partie souterraine en aval. L'enchaînement résultant est constitué de deux groupes thématiques, à savoir la plante puis le sol, puis en un enchaînement des composants dans chaque groupe, d'après un schéma unique dans lequel le calcul de la disponibilité d'une ressource précède celui de la consommation effective. La Figure 13b montre également que les composants dédiés au partage des ressources (en rouge) forment les extrémités du découpage de 'Crops' de 2006. Ainsi, 'light_Interception', chargé de la répartition de la lumière, précède le composant 'Crop' qui calcule l'état de la plante, alors que 'Water_arbitration', qui gère la répartition de l'eau du sol, suit le composant 'Root_crop', qui calcule la croissance du système racinaire. Le découpage des composants en composants élémentaires a par conséquent impliqué une réorganisation des composants.

☞ *R2 d'APES utilise les métadonnées (Tableau 5).*

Quand R2 impacte sur les termes du dictionnaire et sur R1 de MODCOM

La vérification de la cohérence entre la séquence d'appel des composants (R1) et l'alimentation en données d'entrée (R2) s'effectue en superposant le graphe d'enchaînement d'exécution des composants avec celui explicitant le besoin de données. La Figure 14 présente un exemple de superposition pour 2006 et pour 2008. La Figure 14a montre le besoin croisé des composants 'Crops' et 'Soil_water' de 2006 : 'Crops' a besoin en entrée de données de 'Soil_water', et réciproquement. La Figure 14b montre qu'en 2008, la situation est étendue à d'autres composants. Ainsi, les composants sont en interaction pour l'accès aux données.

tel-00458265, version 1 - 19 Feb 2010

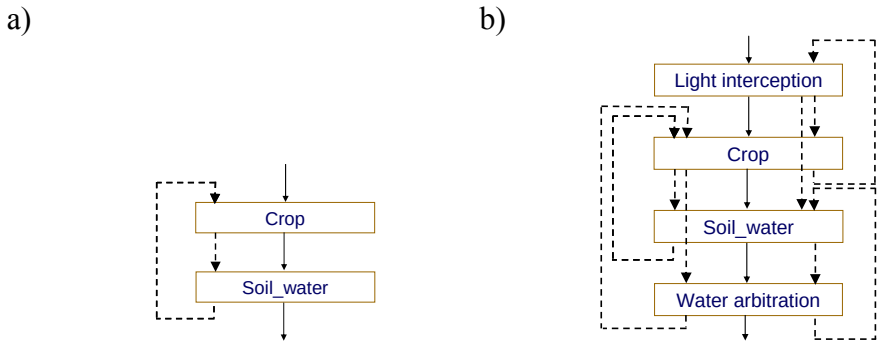


Figure 14 : Exemple de superposition de l'enchaînement d'appel et du besoin en données d'entrée des composants 'Crops' et 'Soil_water' pour 2006 (a) et 2008 (b). (b) est issu de (van Evert et al., 2007). Les flèches en trait continu correspondent à la séquence d'appel des programmes. Les flèches en pointillé expriment le besoin en entrée de données de la part du composant à l'extrémité de la flèche.

Pour la version 2006, la Figure 12b présente une relation d'ordre total : les composants sont exécutés de façon séquentielle par MODCOM, sans autre possibilité. Or, comme le montre la Figure 14a d'un point de vue fonctionnel (flèches en pointillé), les composants sont en interaction vis-à-vis du besoin en données d'entrée (image n°7, Figure 02). L'astuce adoptée pour contourner le problème a consisté en une décomposition de chaque composant. L'interface logicielle connectant le composant à MODCOM est constituée de quatre méthodes. La première méthode, appelée StartRun, a pour objet d'initialiser le composant. L'objet de la seconde, appelée BeforeIntegration, est de récupérer des données d'entrée et de calculer un premier jeu de données de sortie. La troisième méthode, appelée GetRates, a pour objet de récupérer un second jeu de données afin de calculer les variations d'état du composant. Ces variations sont alors transmises à MODCOM, qui effectue l'intégration numérique et calcule l'état du composant. L'objet de la dernière méthode, appelée AfterIntegration, est de procéder à des initialisations ou des opérations particulières (sauvegarde, etc.).
L'exécution de l'enchaînement de composants, présentée en Figure 12b et 14a, nécessite trois exécutions de la chaîne. La première boucle appelle la méthode BeforeIntegration de l'interface logicielle de tous les composants, la seconde la méthode GetRates, et la troisième AfterIntegration. Vis-à-vis du nombre de variables concerné par l'imbrication, cette astuce a été suffisante pour assurer l'échange de données entre les composants en 2006.
Dans la version 2008, la même configuration se rencontre, mais le nombre d'interactions est accru du fait de l'insertion de nouveaux composants. La Figure 13b présente une relation d'ordre total, et la Figure 14b montre plusieurs interactions pour l'accès aux données d'entrée. L'astuce adoptée en 2006 s'avère également suffisante pour résoudre la situation de 2008, ce qui n'était pas évident de prime abord.

La mise en place de l'interaction a requis une modification de la structure logicielle des composants. Le découpage en quatre fonctions correspond à la modification de la valence du composant, à savoir le passage de monovalent à quadrivalent (= l'astuce). La seconde modification concerne le support générique MODCOM. La mise en place de l'interaction à consisté en la transformation d'un support dédié à l'enchaînement de composants monovalents ordonnés selon un ordre total en un enchaînement de composants quadrivalents, toujours selon un ordre total. L'exécution en boucle de l'enchaînement correspond à la mise en place de la rétroaction (image 2, Figure 02) de la séquence complète de composants.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

☞ *La superposition de R1 et de R2 met en évidence la nécessité d'instaurer l'interaction entre les termes. Cette opération a consisté (i) à porter à quatre la valence des termes, et (ii) introduire dans R1 la rétroaction appliquée à toute la séquence de termes (Tableau 5).*

Mise en place de l'interchangeabilité des composants : le cas du doublon « Soilwater » et « Soilwater2 »

La construction d'un programme avec APES s'effectue par juxtaposition de composants. L'insertion du composant 'Soilwater2' en 2008 a donné lieu à une nouvelle fonctionnalité offerte par APES, à savoir l'échange de composants. Dans le cas présent, l'échange concerne les composants 'Soil_water' et 'Soilwater2', dont le sens littéral est identique (la simulation de flux hydriques dans un sol), de même que le nom des variables d'entrée et de sortie, mais dont la fonction sémantique est différente. Afin de caractériser les flux, le sol est discrétisé en un nombre fini de couches. De façon à garantir l'interchangeabilité vis-à-vis des autres composants, notamment l'accès aux informations par couche de sol, c'est APES qui impose la discrétisation du sol, et donc la dimension des vecteurs de données d'entrée et de sortie.
En matière de métadonnées, les données intitulées 'teneur en eau d'une couche de sol' et 'teneur en eau d'une couche de sol à saturation' sont respectivement exprimées en m^3 d'eau/m^3 de sol pour 'Soil_water' et en kg d'eau/kg de sol pour 'Soilwater2'. De façon à garantir l'interchangeabilité, les entrées/sorties de 'Soilwater2' ont du être adaptées pour être connectées aux autres composants d'APES sur la base des entrées/sorties de 'Soil_water'. Pour ce faire, des fonctions de conversion des données ont été adjointes aux entrées/sorties de 'Soilwater2' (voir § 4.2 pour plus de précision). Dans la mesure où les composants 'Soil_water' et 'Soilwater2' ne disposent pas d'une même fonction sémantique, l'ensemble des paramètres des composants (plante etc..) a du être recalculé afin de respecter la fonction sémantique de l'assemblage. La modification des paramètres d'un composant implique la modification de sa fonction sémantique. De ce fait, l'opération d'échange effectuée ne respecte pas la fonction sémantique propre de chaque composant.

☞ L'échange du terme Soilwater par Soilwater2 a nécessité (i) le typage du terme Soilwater2 en regard du terme Soilwater, et (ii) le recalcul de tous les paramètres de l'enchaînement.

Tableau 5 : Evolution des versions successives d'APES selon la trame d'analyse DRS.

		Version 2006	Version 2008
Dictionnaire	Nb de termes	11	23
	Valence du terme	4	4
	Origine du terme	Exogène	Exogène
	Modalités d'insertion	Typage + Reformatage	Typage + reformatage
Sémantique	S1	Action, rétroaction de la séquence complète pour accéder à l'interaction	Action, rétroaction de la séquence complète pour accéder à l'interaction
	S2	Métadonnées	Métadonnées

tel-00458265, version 1 - 19 Feb 2010

3.2.3 SEAMLESS-IF, instance d'OpenMI

Dans la mesure où SEAMLESS-IF utilise le support OpenMI décrit plus haut en § 2.3, la description présentée ici s'attache à mettre en évidence les solutions techniques adoptées pour construire SEAMLESS-IF. La plateforme agronomique SEAMLESS-IF a été développée dans le cadre du projet européen SEAMLESS (2005-2009) dans l'objectif de permettre l'estimation de l'impact de décisions politiques européennes actuelles sur l'agriculture et l'environnement pour les cinquante années à venir (van Ittersum et al., 2008). La plateforme rassemble des programmes établis par plusieurs disciplines (APES pour l'agronomie, FSSIM et CAPRI pour l'économie aux échelles de l'exploitation agricole et de la région, GTAP pour l'analyse politique nationale, etc.) et des bases de données (FADN : Farm Accountancy Data Network, ESDB : European Soil Database, EICD : European Interpolated Climate Data, CAPREG : CAPsis REGlementation dataset). Les programmes et bases de données sont élaborés au moyen de technologies et de langages différents (Janssen et al., 2009). L'enjeu du projet est de mettre en relation ces programmes et bases de données, traitant d'échelles et de connaissances disciplinaires différentes, pour un objectif donné.
Dans un esprit de réutilisation d'éléments existants, la plateforme SEAMLESS-IF a adopté le standard OpenMI. Toutefois, afin d'introduire des données de natures différentes (économiques, etc.) à celles prévues par le standard OpenMI (figure 10), une extension nommée OpenMI+ a été produite par les partenaires du projet SEAMLESS. Dans cette extension, la métadonnée standard a été enrichie par ces données supplémentaires (économie, politiques, etc.) de façon à concilier les entrées/sorties des programmes entre eux et avec les bases de données.

Constitution du dictionnaire

L'architecture de SEAMLESS-IF est constituée de trois couches applicatives (Figure 15). La couche « end-user » comporte les programmes numériques. On y retrouve les programmes APES, FSSIM (Li et al., 2007) et CAPRI (Junker et al., 2007). Ces programmes sont monovalents. La couche « applications » comporte les applications dédiées à la gestion des applications de la couche End-user. On y retrouve par exemple le langage de programmation par contraintes GAMS (http://www.gams.com/) utilisé par FSSIM, MODCOM utilisé par APES, de même que les bases de données (KM, etc.). La dernière couche « SeamFrame » comporte l'application SOFA (SEAMLESS OpenMI+ Framwork Architecture), qui supporte le fonctionnement de l'ensemble. SOFA est le support générique qui implémente OpenMI+, élaboré à partir du support OpenMI.

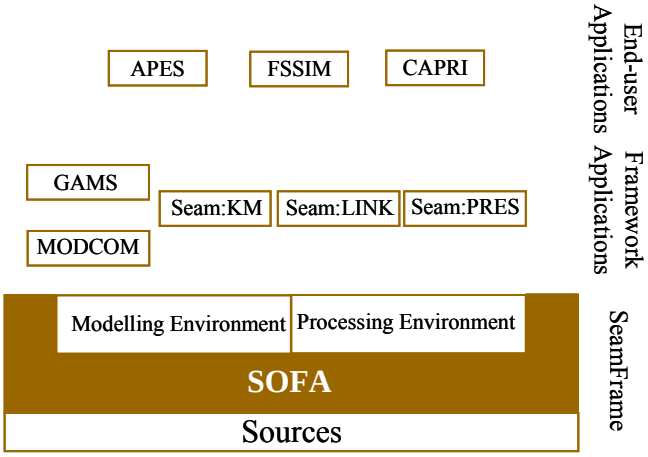


Figure 15 : Les trois couches applicatives de la plateforme SEAMLESS-IF (van Ittersum et al., 2006). La première couche comporte les programmes APES (Agricultural Production and Externalities Simulator), FSSIM (Farm Systems SIMulator), et CAPRI (Common Agricultural Policy Regional Impact Analysis model). La seconde comporte entre autres les programmes GAMS (General Algebraic Modelling System) et MODCOM (Modelling using Component Object Model). La dernière couche comporte SOFA (SEAMLESS OpenMI+ Framework Architecture).

☞ *Le dictionnaire SEAMLESS-IF organise les termes en trois collections. La collection « End-user » comporte les programmes agronomiques. La collection « Application » comporte les programmes dédiés à la gestion des programmes de la collection « End-user ». La collection Seamframe comporte le support générique OpenMI adapté à SEAMLESS-IF. La valence des programmes est de 1 (Tableau 6).*

R1 et S1

SEAMLESS-IF utilise la méthode d'interrelation proposée par OpenMI, à savoir l'interaction. L'exécution d'un programme de la couche End-User demande que l'application, qui la gère au niveau de la couche Framework Applications, soit exécutée par SOFA. En fin d'exécution, le programme de la couche End-User envoie un accusé de fin d'exécution au programme de la couche Framework Applications, qui procède de même avec SOFA. Ce principe de fonctionnement a été conservé pour les deux versions de SEAMLESS-IF produites.

L'enchaînement des programmes de la couche End-User de la première version, sortie en 2006, est présenté en Figure 16a (van Ittersum et al., 2006). L'enchaînement des programmes correspond à une grande boucle, dans laquelle l'interface de SEAMLESS-IF, appelée « Indicator Calculator », exécute CAPRI, CAPRI exécute FSSIM et FSSIM APES. Dans cette grande boucle, deux programmes sont apparus. Il y a d'une part EXPAMOD (Pérez Domínguez et al., 2008), qui est inséré entre CAPRI et FSSIM, et d'autre part FSSIM-AM, qui est inséré entre FSSIM (renommé FSSIM-MP) et APES. La fonction d'EXPAMOD est d'assurer le changement d'échelle entre les modèles économiques CAPRI et FSSIM. La fonction de FSSIM-AM est d'assurer le changement d'échelle spatio-temporelle, de l'exploitation FSSIM à la parcelle APES, et la transition disciplinaire économie FSSIM - agronomie APES. La transition disciplinaire s'effectue par le calcul d'indicateurs, correspondant à une traduction de données agronomiques en données faisant sens pour les

tel-00458265, version 1 - 19 Feb 2010

économistes (Ewert et al., 2009). Les doubles flèches reliant les programmes de changement d'échelle à ceux situés en aval de la Figure 16a indiquent que l'opération de changement d'échelle s'effectue par l'intégration des résultats produits par plusieurs exécutions des programmes situés en aval. Ainsi, à l'intérieur de la grande boucle, initiée par « Indicator Calculator », deux boucles sont imbriquées : celle qui est initiée par EXPAMOD, qui exécute plusieurs fois FSSIM-MP, et celle initiée par FSSIM-MP, qui exécute plusieurs fois APES.
La seconde version a été diffusée en 2008 (Figure 16b). Deux modifications principales ont été apportées à la plateforme agronomique (van Ittersum et al., 2008). D'une part, il y a eu un échange de modèle : CAPRI et l'Indicator Calculator ont été réunis et remplacés par SEAMCAP (CAPRI for SEAMLESS). Il s'agit d'une version de CAPRI adaptée aux objectifs du projet SEAMLESS, dans laquelle diverses fonctionnalités ont été incorporées. D'autre part, la chaîne a été enrichie par de nouveaux programmes, comme par exemple que GTAP (Hertel, 1997), ceux-ci étant insérés sur des maillons différents.

Dans la version 2006, le principe de rétroaction des programmes a été réutilisé au niveau de la couche End-User. Dans la version 2008, ce principe n'est plus tout à fait utilisé : SEAMCAP communique directement avec FSSIM-MP sans passer par EXPAMOD, et l'exécution des programmes « Agricultural Employment », « Territorial Models » et « Developing country model » s'effectue sans rétroaction.

a) b)

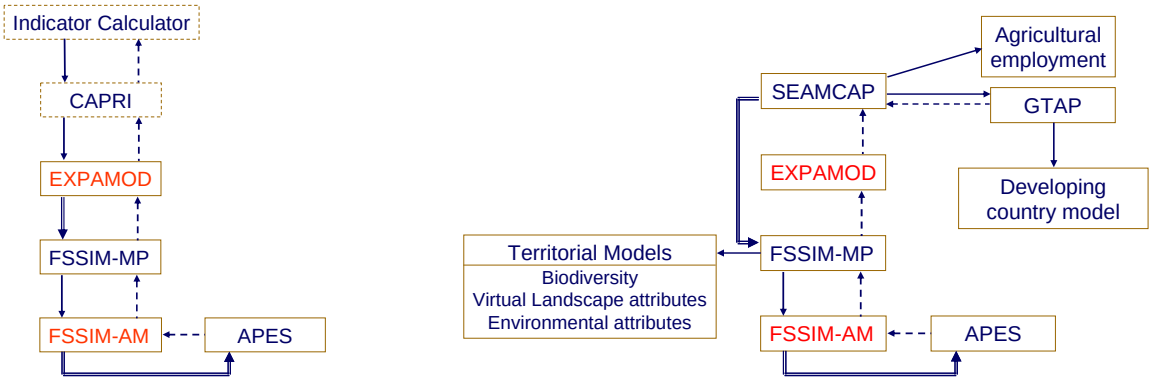


Figure 16 : Chaîne d'exécution des programmes de la couche End-User instaurée dans la plateforme SEAMLESS-IF en 2006 (a) et 2008 (b) issues respectivement de (van Ittersum et al., 2006) et (van Ittersum et al., 2008). Les programmes notés en rouge sur la figure (a) sont des programmes intermédiaires créés spécifiquement pour assurer le changement d'échelle (EXPAMOD et FSSIM-AM) et la transition disciplinaire économie – agronomie (FSSIM-AM).

☞ *R1 de SEAMLESS-IF utilise les modalités d'assemblage permises par OpenMI (l'interaction).*

R2 et S2

Par construction, la plateforme OpenMI met à disposition une métadonnée pour établir la correspondance des entrées/sorties des programmes. Initialement, la cellule de coordination du projet SEAMLESS-IF avait étendu le standard OpenMI, spécifique aux données

tel-00458265, version 1 - 19 Feb 2010

environnementales, de façon à accepter les données économiques (OpenMI+). Par la suite, la cellule de coordination du projet s'est orientée vers l'emploi d'une ontologie (van Ittersum et al., 2008) développée au moyen du logiciel Protégé (http://protege.stanford.edu/). L'ontologie est consultable sur internet à l'adresse suivante : (http://delivered.seamless-ip.org:8060/browser/zul/main.zhtml). L'ontologie a permis de structurer la base de données réunissant l'ensemble des données nécessaires au fonctionnement des programmes. Cette base de données comporte entre autres les données relatives (i) aux cultures agricoles, (ii) aux productions agricoles, (iii) à la structure des exploitations agricoles, (iv) aux conditions pédologiques et climatiques, (v) aux techniques de gestion agricoles, et (vi) aux politiques agricoles (Janssen et al., 2009). Dans un premier temps, l'ontologie a été élaborée à partir des informations localisées dans les métadonnées associées aux bases de données (Janssen et al., 2009). La construction de typologies des informations pour les échelles d'observation considérées par les programmes (parcelle pour APES, exploitation pour FSSIM, et région pour CAPSIS) a donné lieu à l'identification des premiers concepts de l'ontologie. La nomenclature de découpage des unités territoriales européennes NUTS-2 (EC, 2008) a servi de support pour la structuration de l'information spatiale. Par la suite, l'ontologie a été enrichie des concepts et relations nécessaires à l'établissement de la liaison entre les programmes. Considérés indépendamment, CAPRI, FSSIM et APES utilisent les concepts de 'Crop' et de 'Product', respectivement « culture » et « production ». Dans la mesure où les modalités de leur utilisation diffèrent dans la plateforme (objectif et échelle), l'ajout de concepts spécifiques à leur utilisation permet de les différencier. Par exemple, la figure 17 montre les relations entre les concepts 'Crop', 'Product', 'ProductType', 'CropGroup' et 'ProductGroup', ainsi que les propriétés des concepts. Les concepts 'Crop' et 'Product' ont été attribués à APES et FSSIM, et renommés 'CropGroup' et 'ProductGroup' pour CAPRI. La relation entre les concepts 'Crop' et 'CropGroup' et entre 'Product' et 'ProductGroup' est le lien IsPartOf, et correspond à la composition. En dernier lieu, l'ontologie a été complétée des concepts de gestion de l'application SEAMLESS-IF (projet, problème, contexte, etc.). Fin 2008, l'ontologie comportait 379 concepts et 487 relations (Janssen et al., 2009).

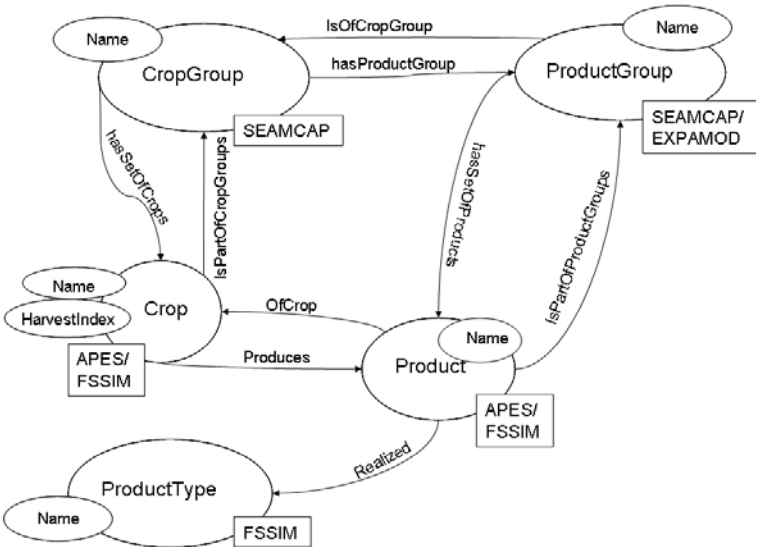


Figure 17 : Les concepts 'Crop', 'Product' et leurs dérivés insérés dans l'ontologie de SEAMLESS-IF (Janssen et al., 2009). Les concepts, propriétés de concepts et relations sont respectivement représentés par des grandes ellipses, petites ellipses et flèches. Les programmes afférents aux concepts sont indiqués dans les rectangles.

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

☞ *Dans la version 2006, la règle R2 mise à disposition par OpenMI (la métadonnée) a été enrichie selon les besoins de SEAMLESS-IF (standard OpenMI+). Dans la version 2008, R2 utilise les ontologies (Tableau 6).*

R3 et S3

L'échange de données entre les programmes s'effectue via des fichiers de données. SEAMLESS-IF opère donc suivant la méthode « Common coupling ».

☞ *En terme sémantique, la R3 de SEAMLESS-IF correspond au niveau de couplage Fort (Tableau 6).*

Tableau 6 : Synthèse des versions de SEAMLESS-IF selon la trame d'analyse DRS.

		Version 2006	Version 2008
Dictionnaire	Valence du terme	1	1
	Origine du terme	Exogène	Exogène
	Modalités d'insertion	Typage	Typage
Sémantique	S2	Métadonnées	Ontologies
	S3	Couplage Fort	Couplage Fort

3.3 Etude comparative des plateformes agronomiques

3.3.1 Architecture et sémantique

Dictionnaire

L'insertion d'un terme dans le dictionnaire de DSSAT a évolué d'une version à l'autre (Tableau 4). Dans la version 2.1, l'insertion requiert le typage du terme en regard de la structuration des données dans les fichiers. La volonté de faire succéder les termes au moyen d'une règle R1 s'est traduite (i) dans la version 3.0 par la standardisation des données situées dans les fichiers, et (ii) dans la version 3.5 par la définition d'un standard d'échange de données entre les termes. Ce mode opératoire n'a pas permis de lever les deux failles relatives à la mise en relation des termes : données manquantes, et différence de valeur de données ayant pourtant un même nom. Les limites de cette construction vis-à-vis des fonctionnalités requises ont remis le mode opératoire en question, et conduit à proposer dans la version 4.0 une architecture logicielle commune à tous les termes de la plateforme agronomique. Le référentiel de typage des termes a donc évolué au fil des versions. Il s'est tout d'abord situé au niveau de la plateforme (v3.0), puis des termes (v3.5), et enfin (v4.0) d'un terme « normalisé » (l'architecture logicielle). Ce terme normalisé est quadrivalent.

L'insertion d'un terme dans APES s'effectue par rattachement à MODCOM (exogène). L'insertion requiert (i) le typage du terme, de façon à permettre l'intégration numérique par MODCOM, et (ii) le reformatage du code source du programme selon une architecture imposée (patrons de conception). L'architecture impose la quadrivalence du terme. Entre les versions 2006 et 2008, l'évolution du dictionnaire concerne uniquement le nombre de termes, porté de onze à vingt-trois (Tableau 5).

Pour SEAMLESS-IF, l'insertion d'un terme dans le dictionnaire s'effectue selon la technique proposée par OpenMI, à savoir le rattachement du programme au moyen d'une interface de liaison. Le typage d'un terme est effectué lorsque les données produites par celui-ci ne sont pas disponibles en sortie. Entre les versions 2006 et 2008, le dictionnaire a été enrichi de nouveaux termes (Tableau 6). Les termes de SEAMLESS-IF sont monovalents alors qu'OpenMI permet la multivalence.

R1 et S1

Pour les versions 3.0, 3.5 et 4.0 de DSSAT, la sémantique (S1) de la règle de juxtaposition des termes R1 est l'action. L'architecture logicielle de la version 4.0 comporte deux S1 : (i) l'action pour la succession d'exécution des modules primaires et secondaires, et (ii) la rétroaction pour l'exécution en boucle de cette séquence.
L'adoption d'un support générique permet aux plateformes SEAMLESS-IF et APES de bénéficier de règles de juxtaposition des programmes. SEAMLESS-IF réutilise le S1 proposé par OpenMI (rétroaction) sans modification, et reste ainsi dans le cadre strict de la réutilisation. APES réutilise la règle R1 de MODCOM. De façon à permettre l'interaction entre les termes, une règle supplémentaire a été introduite dans MODCOM. Cette règle permet la rétroaction de la séquence complète d'appel des termes. La construction effectuée en 2008 correspond à une extension de celle de 2006. Elle présente un plus grand nombre de termes inter-reliés, et par voie de conséquence pouvait laisser supposer un degré d'imbrication plus élevé des termes, et donc de valence (Figure 14b). La quadrivalence a pourtant été suffisante pour assurer la cohérence de la construction.

R2 et S2

Dans DSSAT, la métadonnée est le support de mise en coïncidence des données échangées entre les termes. Une des difficultés non résolues dans les versions 3.0 et 3.5 concerne la différence de valeur attribuée à des données portant le même nom. Pour lever ce problème, la solution adoptée dans la version 4.0 a été d'adopter un sous-programme sol identique (la partie critique), restreignant alors l'usage des métadonnées aux parties non critiques.

Deux méthodes de mise en coïncidence des entrées/sorties sont employées. La première repose sur l'adjonction d'une métadonnée qui complémente la donnée requise en entrée du terme ou proposée en sortie. Cette approche part du postulat selon lequel l'ensemble des informations contenues dans chaque métadonnée suffit à qualifier complètement la donnée, et permet une correspondance cohérente entre les entrées/sorties. Dans APES, la métadonnée est constituée de cinq types d'informations (description littérale, unité d'expression, valeurs limites et par défaut). En théorie, l'utilisation de métadonnées permet d'assurer la mise en coïncidence des entrées/sorties des termes. En pratique cependant, l'identité des données est vérifiée par échanges discursifs entre les chercheurs SV, concepteurs des termes. En effet, le remplissage des métadonnées ayant été effectué pour chaque terme indépendamment des autres, la signification des terminologies varie d'une discipline à l'autre, et n'est donc pas partagée par tous. La difficulté de l'opération réside alors dans la mise en coïncidence de terminologies partagées.
Dans OpenMI, la métadonnée comporte quatre classes d'informations (What, When, Where, How) définies pour des données environnementales. Une classe comportant plus d'informations qu'un type (unitaire), les informations contenues dans la métadonnée OpenMI

tel-00458265, version 1 - 19 Feb 2010

sont par conséquent plus nombreuses que celles contenues dans la métadonnée d'APES. Ce faisant, pour mettre en coïncidence les terminologies, la métadonnée définie dans OpenMI offre plus d'opportunités (figure 10). Toutefois, dans la construction de SEAMLESS-IF version 2006, la métadonnée initiale a du être mise à jour en regard des données économiques à considérer (OpenMI+). Malgré cet enrichissement, la mise en coïncidence des métadonnées donne lieu à des problèmes sémantiques. Le sens de certains mots diffère en effet selon la discipline politique, économie ou agronomie. Ainsi 'Crop' dans CAPSIS signifie la culture comme production d'une filière agricole régionale, dans FSSIM c'est le type de production végétale au sein d'une exploitation, alors que pour APES il s'agit de la plante cultivée sur une parcelle. En raison de la variabilité de signification d'une terminologie selon les disciplines, OpenMI+ n'est pas suffisamment puissant pour permettre de lever l'ambiguïté sémantique.
Afin de lever cette ambigüité, la version 2008 de SEAMLESS-IF a opté pour la seconde méthode de mise en coïncidence des entrées/sorties, à savoir la construction d'une ontologie. Cette option présente l'avantage de proposer un langage permettant de spécifier les relations entre concepts. Dans SEAMLESS-IF, la construction de l'ontologie relève d'une approche « top-down », qui consiste à partir du général pour aller au particulier. Elle est établie à partir d'informations localisées dans les métadonnées associées aux bases de données, et enrichie selon les besoins d'échange de données entre termes. Dans la mesure où certains termes utilisaient des concepts ayant la même dénomination mais avec une signification différente, ceux-ci ont du être revisités. L'ontologie a ainsi permis de lever l'ambiguïté sémantique entre termes en définissant un positionnement relatif de la terminologie. En ce qui concerne les cultures par exemple, le concept 'Crop' de CAPSIS a été transformé en 'CropGroup' et une relation de composition établie entre 'Crop' et 'CropGroup' (Figure 17). En revanche, l'attribution d'un même concept 'Crop' à l'attention de disciplines différentes (agronomie et économie) renvoie à la façon de concevoir l'ontologie. Ainsi, alors qu'il s'agit de concepts de nature différente, le 'Crop' agronomique d'APES est inclus dans le 'CropGroup' politico-économique de CAPSIS. Ce qui n'est pas cohérent sur le plan sémantique. Cette situation résulte de la difficulté d'exprimer les variantes sémantiques d'un concept, celui de 'Crop' en l'occurrence, au moyen d'une simple relation formelle logico-mathématique.

L'énoncé de l'ordre d'appel des termes s'effectue selon le besoin relatif de données en entrée. Pour APES, celle-ci a été définie au moyen des métadonnées. Cette méthode a permis de définir la valence des termes, valence qui a été conservée en 2008. Pour SEAMLESS-IF, l'ordre d'appel a également été établi au moyen des métadonnées. La juxtaposition des entrées/sorties des termes a cependant mis en évidence deux incohérences sémantiques. La première concerne le couple [CAPRI → FSSIM]. Ces deux termes traitent d'économie mais à des échelles spatiales différentes, la région et l'exploitation agricole respectivement. La solution adoptée a consisté à insérer le terme EXPAMOD entre ces deux termes. Ce terme, élaboré spécialement à cette intention, effectue le changement d'échelle par intégration numérique des résultats fournis par plusieurs exécutions de FSSIM.
La seconde incohérence sémantique concerne le couple [FSSIM → APES], où APES est le terme redevable de la discipline agronomique à l'échelle de la parcelle agricole, et FSSIM celui de la discipline économique à l'échelle de l'exploitation agricole. Dans ce cas, l'incohérence a été levée par l'introduction du terme FSSIM-AM. FSSIM-AM, inséré entre FSSIM et APES, assure deux types de transition : d'une part le changement d'échelle spatial, et d'autre part la transition disciplinaire économie-agronomie. La transition disciplinaire est résolue par la construction d'indicateurs qui traduit des données signifiantes pour la discipline de départ et absconses pour la discipline d'arrivée en données signifiantes pour la discipline d'arrivée. Le changement d'échelle a été résolu selon le même schéma qu'EXPAMOD, à savoir l'intégration de résultats de plusieurs exécutions du terme APES. La technique adoptée

tel-00458265, version 1 - 19 Feb 2010

pour le changement d'échelle est la même pour EXPAMOD ou FSSIM-AM. Leur développement a cependant nécessité l'implémentation de fonctions d'intégration numérique, ces fonctions étant mises à dispositions par le support MODCOM.

R3 et S3

Pour toutes les versions de DSSAT, S3 correspond au niveau de couplage Fort. APES hérite de S3 du support générique MODCOM, correspondant au niveau de couplage Moyen++. SEAMLESS-IF n'utilise pas le mécanisme d'échange proposé par le support générique OpenMI. Pour SEAMLESS-IF, l'échange de données s'effectue via un fichier intermédiaire. Cette méthode correspond au niveau de couplage Fort.

3.3.2 Impact du sens littéral des termes

Dans les paragraphes précédents, les termes ont été considérés comme des briques indépendamment de leur sens littéral (concept de boite noire). Dans ce paragraphe, les constructions sont discutées en regard du sens littéral des termes, toujours selon la trame d'analyse DRS.

Constitution du dictionnaire

Le sens littéral des termes varie selon les plateformes agronomiques. Les termes d'APES sont des programmes de simulation de phénomènes biophysiques, ceux de DSSAT sont des programmes de simulation du fonctionnement d'une culture agricole mono-spécifique, et enfin ceux de SEAMLESS-IF sont des programmes redevables de disciplines SV plurielles (APES pour l'agronomie, FSSIM et CAPRI pour l'économie, etc.). L'objet d'APES étant de produire un programme de simulation du fonctionnement d'une culture, il existe donc une identité sémantique entre la construction agronomique APES et le terme DSSAT. De plus, les équations biophysiques utilisées dans les deux constructions sont de même nature. Il est donc intéressant de les comparer sur le plan sémantique.

Les versions 2006 et 2008 d'APES permettent de simuler le fonctionnement d'une culture agricole respectivement mono-spécifique et plurispécifique (« Grass » et « Tree » par exemple). L'identité sémantique entre les versions d'APES et le terme DSSAT varie donc selon les versions : elle est totale pour celle de 2006, et partielle (ou inclusive) pour celle de 2008. L'histoire de DSSAT a conduit dans la version 4.0 à disposer d'une architecture logicielle commune (notée ALC dans la suite du document) à tous les termes.

Les termes d'ALC sont les modules primaires et secondaires de l'architecture logicielle (Figure 11). Ces modules sont quadrivalents (Tableau 4). Les modules secondaires sont associés au module primaire selon une relation hiérarchique d'inclusion. Deux S1 sont utilisées : (i) l'action pour l'enchaînement des modules et (ii) la rétroaction appliquée à l'enchaînement des modules pour exécuter toutes les quatre méthodes (rétroaction interne au terme, Tableau 4). La règle R2 est la métadonnée. La transmission des données s'effectuant via les fichiers de données, S3 correspond au niveau de couplage Fort.

tel-00458265, version 1 - 19 Feb 2010

La comparaison des descriptions d'ALC de DSSAT avec les versions 2006 et 2008 d'APES (Tableau 5) montre que dans les deux cas les termes sont quadrivalents. La décomposition consécutive à la quadrivalence est effectuée selon le même schéma : Initialisation, calcul de la variation des variables d'état, intégration des variables d'état, et gestion des données. De plus, les S1 des versions d'APES et d'ALC sont strictement identiques : action et rétroaction sur l'ensemble de la séquence. Les S2 utilisent des méthodes similaires (les métadonnées), qui ne sont cependant pas appréhendées selon la même logique : collection de données pour APES et utilisation de codes standards pour ALC. En revanche, les S3 diffèrent : niveau de couplage Fort pour ALC et Moyen++ pour APES. Vu les identités terminologiques et syntaxiques, il s'en suit qu'ALC et APES devraient théoriquement être en mesure de représenter les mêmes cultures agricoles. En particulier, ALC devrait pouvoir permettre l'implémentation d'une culture plurispécifique. Or, si on compare les dictionnaires, on constate que les termes d'APES sont indépendants, tandis que les termes primaires et secondaires du dictionnaire d'ALC sont liés par une relation d'inclusion. Cette relation correspond à la mise en place d'une interrelation, de type action, des termes primaires sur les termes secondaires. Cette interrelation se superpose à S1, et de fait, contraint les constructions possibles.

L'indépendance des termes du dictionnaire APES a permis d'accéder à la culture plurispécifique dans la version 2008. La mise en place a consisté (i) à insérer d'autres termes dans l'enchaînement («Tree » et « Grass » par exemple), (ii) à décomposer les termes existants pour le partage de ressources (« water arbitration » et « light interception », Figure 13b), et (iii) à réordonner l'enchaînement en fonction de S2. Cette opération a été rendue possible grâce à l'indépendance des termes les uns par rapport aux autres. Ainsi par construction, ce ne sont pas seulement deux cultures qui peuvent être considérées simultanément dans un enchaînement d'APES, mais un nombre « infini ». Pour ce faire, il suffit d'insérer autant de termes « culture » dans l'enchaînement que nécessaire.

Pour ALC, l'obtention de la pluri-spécificité à partir de l'organisation actuelle requerrait des travaux conséquents. En référence à l'expérience APES, l'opération nécessiterait (i) l'ajout de composants primaires et secondaires dédiés au partage des ressources, (ii) la décomposition de certains composants secondaires, et (iii) leur réorganisation. De façon à permettre l'interaction entre cultures, des modules primaires additionnels doivent être créés. L'obtention de la pluri-spécificité implique pour ALC la revisite de l'ensemble de la construction. En conclusion, l'adoption d'une relation hiérarchique entre modules, du type de celle adoptée dans DSSAT, contraint très fortement les possibilités de représentation de systèmes agricoles complexes.

R1 et S1

Nous avons vu en §1.2 que la règle R1 permet d'assembler des termes au moyen des relations série et parallèle. R1 dispose d'une sémantique S1, qui peut être l'action, la rétroaction, l'interaction, le parallélisme, ou une combinaison de ces sémantiques. Plus loin dans l'analyse des plateformes agronomiques (§ 3.1) nous avons vu qu'un terme est caractérisé (i) par un sens littéral, c'est à dire l'objet naturel qu'il représente, et (ii) par une fonction sémantique. L'assemblage de deux termes au moyen d'une règle R1 s'interprète au niveau sémantique comme l'assemblage du sens littéral des termes au moyen de la sémantique des règles. Au niveau syntaxique, la résultante de l'assemblage est la composition des termes, et au niveau sémantique la sémantique de la composition. Ainsi, le choix des relations délimite les possibilités d'assemblage, et par conséquent de formulations sémantiques.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

tel-00458265, version 1 - 19 Feb 2010

A titre d'illustration, l'adoption de l'action comme unique forme élémentaire (image 4, Figure 02) circonscrit le dialogue entre termes à un lien monodirectionnel (monologue) et ne donne donc accès qu'à la relation causale. C'est l'option adoptée dans MODCOM et dans toutes les versions de DSSAT. L'absence de dialogue réduit les possibilités d'usage pour les sciences du vivant. En agronomie, par exemple, l'association de cultures (ou culture plurispécifique) rassemble des cultures présentes en même temps et sur un même site. La spécificité demandée à un outil implémentant ce type de fonctionnement est de permettre le partage des ressources naturelles (lumière, eau, etc.) entre les cultures. En termes informatiques, cette fonctionnalité nécessite l'interaction. Afin d'accéder à cette fonctionnalité, la solution adoptée dans APES a consisté à modifier la valence des termes, de monovalent à quadrivalent, et à multiplier le nombre de cycles d'exécution de la séquence d'appel de MODCOM (rétroaction de la séquence complète). Cette solution autorise l'établissement de l'interaction entre termes à partir d'une action. Le dialogue est toutefois limité à une seule interaction entre paires de termes.

OpenMI utilise l'interaction comme unique forme élémentaire (image 7, Figure 02). Par construction, il permet d'établir un dialogue illimité entre les termes. Bien que ne mettant pas à disposition une fonction de résolution numérique, OpenMI donne accès aux systèmes complexes, à savoir un ensemble d'éléments en interaction (von Bertalanffy, 2002).

L'idéal en matière de représentation de système complexe est de disposer d'un support d'assemblage permettant l'utilisation des quatre formes sémantiques élémentaires d'association (Klir et Valach, 1967). C'est l'option retenue par VLE, avec DEVS comme support théorique. Cependant, pour qu'un terme, connecté à un processeur DEVS via une interface de connexion, puisse accéder à toutes ces possibilités de liaison, le terme doit être multivalent, et formulé très précisément au format d'un processeur DEVS. En effet, ce n'est pas parce qu'un programme est connecté à un processeur DEVS que ce programme devient multivalent. Les diverses formes auxquelles un terme peut accéder sont donc liées à la modalité d'insertion du terme, à savoir une connexion simple à un processeur DEVS ou une transcription dans le langage du processeur DEVS.

R2 et S2

R2 a pour objet de mettre en correspondance les entrées/sorties des termes. APES, DSSAT et la version 2006 de SEAMLESS-IF utilise la méthode des métadonnées, et la version 2008 de SEAMLESS-IF les ontologies. Ces constructions agronomiques réunissent des termes conçus par des équipes différentes. Bien qu'utilisant des méthodes similaires, la problématique de mise en coïncidence varie d'une construction à l'autre.

SEAMLESS-IF rassemble des termes conçus par différentes disciplines : agronomie pour APES, économie à l'échelle de l'exploitation agricole pour FSSIM, et économie à l'échelle régionale pour CAPRI, etc. La problématique posée par la R2 de SEAMLESS-IF est donc d'accorder les entrées/sorties de termes dont les différences entre les sens littéraux sont induites par les disciplines. Cette problématique est homologue à la mise en coïncidence de langages différents. En termes de résolution, les métadonnées ont permis, dans la version 2006, d'identifier la nécessité d'insertion de programmes médiateurs entre APES et FSSIM, et entre FSSIM et CAPRI. L'utilisation des ontologies dans la version 2008 a permis de « lever » les ambiguïtés sémantiques entre les entrées/sorties des termes.

tel-00458265, version 1 - 19 Feb 2010

APES rassemble des termes dont le sens littéral est redevable d'une même discipline (l'agronomie) mais relevant de différentes thématiques. Par exemple, le terme « Soil-water » reproduit le fonctionnement de l'eau dans le sol, « Crops » le développement d'une culture agricole, etc. La problématique pour la R2 d'APES est donc d'accorder les vocabulaires de domaines thématiques appartenant à une même discipline scientifique. Dans le cas présent, les métadonnées ont permis d'identifier les correspondances entre vocabulaires. La correspondance est validée par les spécialistes des différents domaines thématiques, à l'issue de discussions.
DSSAT rassemble les termes disposant d'un sens littéral identique i.e. la « simulation du fonctionnement d'une culture agricole ». La méthode de construction adoptée pour chaque terme repose sur un même mode de représentation de l'espace cultivé : plante-sol-atmosphère. La différence entre les termes porte sur la plante cultivée considérée (maïs, blé, etc.). La problématique posée à DSSAT est donc de faire coïncider des vocabulaires de spécialistes d'un même domaine thématique. Dans les versions 3.0 et 3.5, l'adoption d'un standard correspond à définir un langage commun à toutes les équipes. Cette méthode a permis de mettre en évidence l'existence de données portant un même nom et représentant le même phénomène biophysique mais ayant des valeurs différentes (la faille n°2). L'existence d'une telle faille révèle que, même entre équipes de recherche impliquées dans un même domaine thématique, des variations sémantiques du vocabulaire existent. Pour résoudre cette « faille », la solution adoptée a consisté à imposer un sous-programme sol unique. A défaut de pouvoir mettre en correspondance les vocabulaires, ce mode opératoire correspond à uniformiser les modes de représentation de l'espace cultivé.

Les problématiques de mise en coïncidence des entrées/sorties des termes sont différentes selon la plateforme agronomique considérée. Pour SEAMLESS-IF, il s'agit d'établir la correspondance de langages scientifiques. Pour APES, il s'agit d'établir la correspondance de vocabulaires d'un même langage scientifique mais portant sur des domaines scientifiques différents, et pour DSSAT la correspondance de vocabulaires de domaines scientifiques identiques. D'un point de vue opérationnel, la mise en coïncidence des entrées/sorties a été effectuée pour SEAMLESS-IF, APES et DSSAT respectivement au moyen des ontologies, des métadonnées et par le couple (métadonnées, uniformisation). Le choix de la méthode s'effectuerait donc en fonction de la distance disciplinaire et entre domaines thématiques considérés.

Dans les exemples livrés, les métadonnées servent à renseigner les données en regard du programme qui les utilise ou les produit. Elles sont établies par le collectif scientifique concepteur du programme. La difficulté à établir une relation entre les entrées et les sorties au moyen des métadonnées réside dans le sens implicite inhérent à chaque programme. Or c'est le sens implicite qui limite le sens communiqué entre entrées et sorties. Les ontologies quant-à-elles sont élaborées au sein de la communauté scientifique ayant contribué à la construction des plateformes agronomiques. Les ontologies sont élaborées en regard d'un vocabulaire « partagé » par la communauté. Les exemples ci-dessus montrent que le vocabulaire établi ne suffit pas à la mise en correspondance des entrées/sorties des programmes.

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

R3 et S3

Pressman (1992) présente les diverses modalités de transmission des données du point de vue technique, et qualifie chacune au moyen d'un niveau de couplage. Le choix de la modalité par un support ou une plateforme a également une signification. Dans le cas des couplages Faible et Faible++, la transmission des données entre termes s'effectue sans que la plateforme interfère. Autrement dit, la plateforme véhicule d'une manière neutre les données entre termes indépendants. C'est l'option adoptée par OpenMI (et VLE).
Le niveau de couplage Moyen correspond à l'influence des données d'un terme P_A sur le fonctionnement d'un terme P_B. Cela implique que P_A produit une action, qui altère le comportement de P_B, tel que le définit Klir et Valach (1967 ; schéma 4, Figure 02). L'indépendance des programmes est donc partiellement rompue.
Le niveau de couplage Moyen++, choisi par MODCOM, correspond au formatage des données échangées entre les termes par une interface extérieure. Le véhicule qui transporte les données interfère alors sur le transfert des données. Les programmes sont dépendants du véhicule vis-à-vis de la transmission de données. Les niveaux de couplages « moyen » et « Moyen++ » impliquent donc l'établissement d'une dépendance fonctionnelle des termes, soit entre eux dans le cas du couplage « Moyen », soit avec le support de couplage dans le cas du niveau de couplage Moyen++.
Les niveaux de couplages Fort et Fort++ correspondent au partage d'un espace de données. Cet espace est global (ou public) à tous les termes dans le cas du couplage Fort, et local à un terme (ou privé) dans le cas du Fort++. Ce mode opératoire implique la mise en place d'une dépendance structurelle des termes. Dans le cas de DSSAT, les termes sont structurellement dépendants de la plateforme.

Ainsi, l'adoption d'une modalité de transmission, par un support générique ou une plateforme, a une influence sémantique sur l'association des termes. La rupture de l'indépendance fonctionnelle ou structurelle d'un terme correspond alors à une perte d'autonomie et à sa subordination à un élément extérieur (terme ou support).

3.3.3 Conclusion partielle

Le tableau 7 récapitule les caractéristiques de la dernière version des plateformes logicielles DSSAT, APES et SEAMLESS-IF. Dans les paragraphes précédents, nous avons montré que la dépendance des termes du dictionnaire limite l'aptitude à représenter les systèmes complexes (DSSAT vs. la culture plurispécifique, par exemple). Dans la même logique, le choix de R3 influe sur la dépendance fonctionnelle des termes en regard de l'échange de données et du comportement des termes.
Nous avons également montré que dans R1, le choix des méthodes de mise en interrelation des termes contraint l'aptitude à représenter les systèmes complexes (MODCOM vs. interaction et parallélisme, par exemple).

Concernant R2 enfin, l'analyse des plateformes agronomiques met en évidence l'impact du sens implicite dans l'assemblage au niveau de la mise en correspondances des entrées/sorties. Par rapport aux métadonnées, les ontologies utilisées dans les exemples permettent de lever certaines ambigüités sémantiques au sein d'un champ disciplinaire. Toutefois, les vocabulaires construits ne permettent de lever les problèmes de polysémie liés à l'interdisciplinarité (concept 'Crop' par exemple).

Tableau 7 : Principales caractéristiques des plateformes logicielles agronomiques utilisées

		DSSAT	APES	SEAMLESS-IF
Support générique		Aucun	MODCOM	OpenMI
Dictionnaire	Valence	1 dans la v2.1 4 dans la v4.0	4	1
	Modalités d'insertion	Typage + Reformatage + partie commune (sol)	Typage + Reformatage	Typage
Sémantique	S1	Action entre les termes, Rétroaction interne au terme	Action, Rétroaction de la séquence complète pour accéder à l'Interaction	idem OpenMI (Rétroaction)
	S2	Métadonnées pour la mise en coïncidence de terminologies redevable d'un même domaine thématique	Métadonnées pour la mise en coïncidence de terminologies de domaines thématiques différents	Ontologies pour la mise en coïncidence de terminologies de disciplines différentes
	S3	Fort	Moyen++	Fort

4 La constitution du dictionnaire

Au §2.4, nous avons vu que l'insertion d'un terme dans un support générique peut s'effectuer soit (i) en développant le terme au moyen du langage proposé par le support générique, à l'exemple de MODCOM et VLE, soit (ii) en rattachant le terme au support générique, grâce à une interface de liaison, à l'exemple de MODCOM, VLE et OpenMI. Pour les plateformes agronomiques APES, DSSAT et SEAMLESS-IF, la méthode d'insertion adoptée est le rattachement. Selon les cas, le rattachement requiert le typage ou le reformatage du terme. Dans ce paragraphe, nous nous proposons d'étudier ces deux mécanismes.

4.1 Reformatage d'un terme d'après un langage imposé

Le reformatage du code source d'un terme s'effectue en regard d'un objectif particulier. Pour l'insertion dans MODCOM et VLE, l'objet du reformatage est d'accéder à la fonctionnalité d'intégration numérique. Dans VLE en sus, le formatage a pour objet d'accéder à la multivalence. Dans DSSAT, l'objet du reformatage est d'échanger le module sol du terme par un module sol commun à tous les termes. Dans APES enfin, l'adoption d'une architecture logicielle commune permet de partager les outils de développement. Dans ces quatre cas, l'opération de reformatage correspond à une réorganisation des équations numériques insérées dans le code source, selon une structure imposée. Les équations numériques représentent le fondement de la fonction sémantique. Leur réorganisation relative remet donc en cause la préservation de la fonction sémantique du terme.

4.1.1 L'expérience APES

Pour rappel, l'architecture logicielle des composants d'APES résulte de la combinaison des patrons de conception Façade, Composite, Stratégie et Adaptateur. L'architecture est quadrivalente. Les données sont localisées dans une mémoire globale, et le code source est réparti dans quatre fonctions principales (StartRun, BeforeIntegration, GetRates,

tel-00458265, version 1 - 19 Feb 2010

AfterIntegration). Cette structuration logicielle est redevable du paradigme procédural (§ 3.2.2).

Transcription

Pour le développement d'APES, deux cas de reformatage se sont présentés. Dans le premier cas, le langage de construction du programme agronomique était redevable du même paradigme (procédural) que celui retenu par l'architecture logicielle APES. Cette technique de reformatage a été adoptée par exemple pour construire le terme « Crops » à partir du programme de simulation de cultures LINTUL, développé en FORTRAN. Du point de vue technique, l'opération de reformatage a consisté (i) en la réunion des données manipulées par le programme au sein de l'interface logicielle du terme, et (ii) en la répartition du code source (équations numériques) dans les quatre fonctions de l'architecture logicielle. La contrainte imposée par cette répartition porte sur la conservation de l'ordre de calcul dans le but de préserver la fonction sémantique.

Dans le second cas, le langage utilisé pour construire le programme agronomique est différent du paradigme procédural. La transcription correspond alors à un exercice de migration d'un programme écrit dans un langage vers un autre (Mens et Van Gorp, 2006). L'exemple choisi est celui du programme sol Kamel (Figure 18) développé au moyen de l'outil d'aide à la modélisation SIMILE (Muetzelfeldt et Massheder, 2003), et transcrit en un terme appelé 'SoilWater2' (Martin et al., 2007a; Mohtar et al., 2008 ; Braudeau et al., 2009). SIMILE est un langage de type objet établi sur la base de la théorie des systèmes. C'est un langage d'aide à la modélisation (LAM, §3.1) qui met à disposition du programmeur (i) des classes d'éléments à sens littéral prédéterminé (variables informatives, compartiments d'énergie et containers d'inclusion d'éléments), et (ii) des classes de liens permettant la mise en relation des éléments (flux d'énergie et d'information). La construction graphique du programme s'effectue par instanciation de ces deux ensembles de classes. Les éléments instanciés pouvant être aisément transformés en tableaux d'éléments (par un simple clic), y compris les containers d'inclusion, une relation binaire entre deux éléments peut se transformer en un produit cartésien.

Du point de vue technique, la transcription d'un programme d'un paradigme objet en procédural requiert la reformulation du programme dans son ensemble. Elle correspond à transformer un ensemble d'objets inter-reliés (relation d'inclusion, relation associative, etc.), et disposant de méthodes propres, en une séquence de fonctions appliquées à un jeu de variables. A cette occasion, (i) les éléments SIMILE ont été convertis sous la forme de variables insérées dans l'interface, et (ii) les méthodes propres aux éléments SIMILE (initialisation, calcul pour chaque pas de temps, etc.) sont devenues des fonctions numériques de l'architecture logicielle appliquées à des variables. Les liens orientés entre paires d'éléments, et entre tableaux d'éléments, ont servi de support à l'identification de l'ordre d'appel des fonctions numériques. Les fonctions numériques ont ensuite été réparties dans les quatre fonctions principales de l'architecture logicielle, tout en respectant la fonction sémantique. D'autre part, une méthode d'intégration numérique, préexistante dans SIMILE, a du être implémentée dans le composant « Soil-water » afin de préserver la fonction sémantique. Cette fonctionnalité est requise pour traiter des pas de temps inférieurs à celui proposé par MODCOM pour APES (quotidien). Ce travail de transcription, réalisé en regard de la théorie des catégories et conduit par l'auteur du mémoire, est présenté en annexe 1. La

tel-00458265, version 1 - 19 Feb 2010

théorie des catégories a permis de mettre en évidence les classes d'éléments et les relations entre classes d'éléments n'ayant pas de correspondance (Martin et al., 2008a).

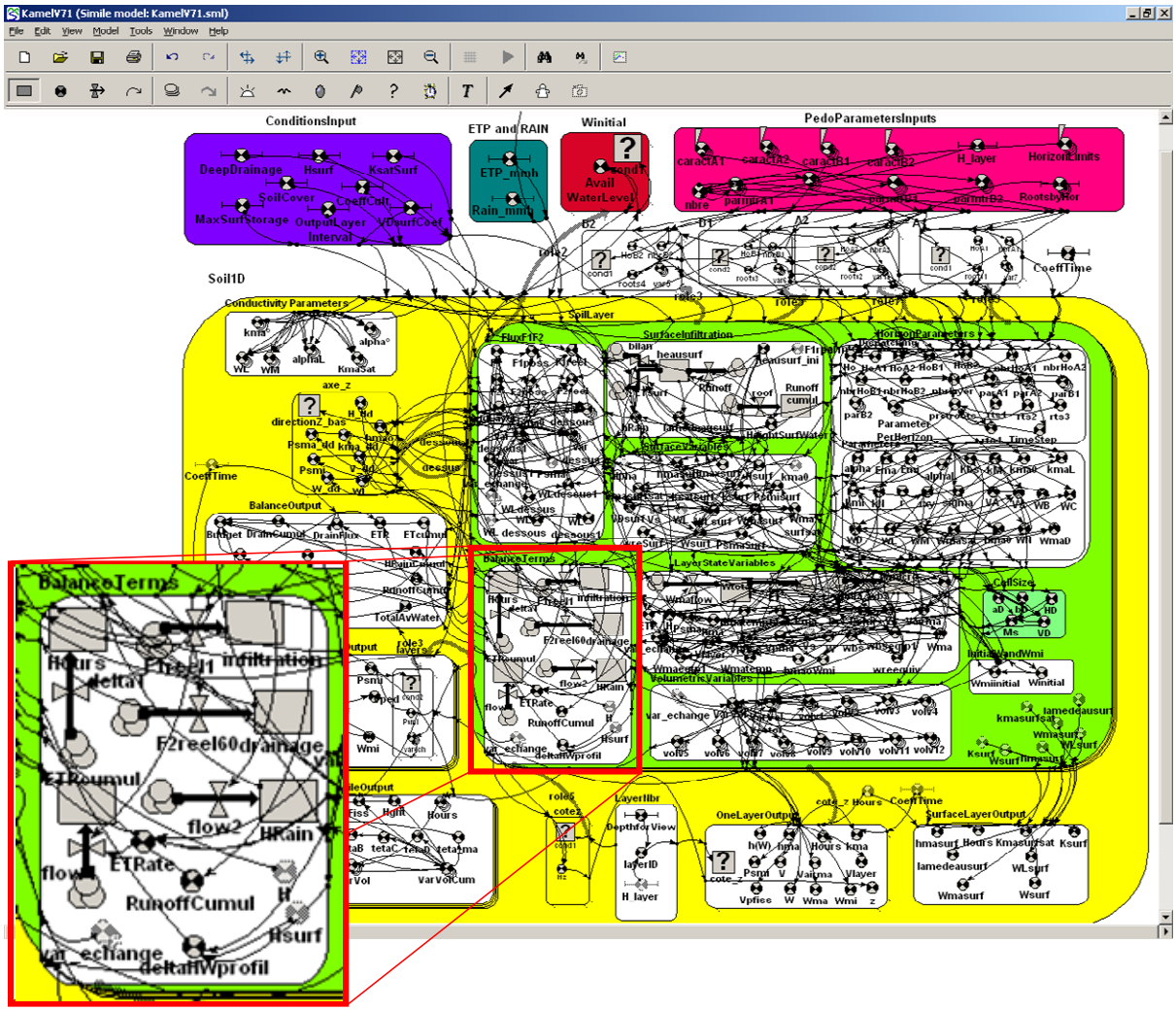


Figure 18 : Programme KAMEL, formulé dans le langage SIMILE, utilisé pour produire le composant Soilwater2.

Dans ce travail de transcription, deux informations existantes dans le programme agronomique SIMILE n'ont pu être préservées. La première concerne le container d'inclusion d'éléments de SIMILE, pour lequel APES ne dispose pas d'équivalent. La relation d'inclusion entre éléments n'existant plus, leur organisation relative disparait, de même que le sens qu'elle véhicule. La transcription a ainsi eu pour effet de transformer une organisation d'éléments liés sous SIMILE en une collection de variables indépendantes dans l'architecture logicielle d'APES. ***Dans la trame d'analyse DRS, cette perte correspond à une altération du dictionnaire (l'élément inclusif) et aussi une perte de R1 (l'inclusion)***.

La seconde information qui n'a pu être préservée concerne le sens littéral des éléments de SIMILE. En effet, l'utilisation de la théorie des systèmes amène à différencier la variable informative du compartiment d'énergie, dont le sens littéral est respectivement « détenir une information » et « contenir de l'énergie ». Par opposition dans APES, le rôle fonctionnel

tel-00458265, version 1 - 19 Feb 2010

d'une variable n'est pas pris en compte. Un terme APES ne comporte donc plus que des variables. La transcription en variables indifférenciées APES de variables informatives et de compartiments d'énergie de SIMILE se traduit par une perte de sens littéral. ***Dans la trame d'analyse DRS, cette perte affecte le dictionnaire.***

Les programmes initiaux et transcrits ont bien les mêmes sens littéraux et fonctions sémantiques. Toutefois, la transcription s'est accompagnée d'une perte de sens dans le contenu des programmes.

Actualisation

Une première version du composant 'Soilwater2' a été produite en 2007. Fin 2008, un certain nombre d'améliorations ont été apportées par le concepteur SV, sous SIMILE. Ce travail, effectué au cours de l'année, a conduit à un nombre important de modifications non relevées. La première tâche accomplie pour actualiser le composant sous format APES a donc consisté à identifier les transformations par comparaison des codes source SIMILE des versions 2007 et 2008. L'exercice a permis de montrer que les modifications réalisées étaient uniquement redevables d'agrégations locales de code. Le code source de 2007 est par conséquent resté intact dans la version de 2009 (Martin et al., 2009a ; Martin et al., 2009b).

La mise à jour du composant a donc consisté à repérer dans le code source de 'Soilwater2' les points d'accrétion des fragments de code agrégés au programme SIMILE, puis à introduire ces fragments dans le code d'APES. L'opération de mise à jour n'était pas triviale dans la mesure où l'organisation des éléments de SIMILE (relation d'inclusion) n'existe plus dans APES. A titre d'exemple, les éléments monodimensionnels nouvellement insérés dans le programme SIMILE, et inclus dans les tableaux de containers, ont dû (i) être transformés en variables multidimensionnelles dans APES, et (ii) être connectés dans les équations numériques. Cette connexion a demandé l'application du produit cartésien sans recours à l'automatisme du « clic » de SIMILE. L'insertion des équations numériques, et le respect des produits cartésiens, a été effectuée en suivant les liens orientés entre paire d'éléments. Si les modifications avaient porté sur le code initial, alors le programme aurait dû être entièrement retranscrit. ***Par rapport à la trame d'analyse DRS, cet exemple illustre l'effet de l'altération du dictionnaire sur l'actualisation du programme.***

4.1.2 Discussion

Le reformatage du code source est l'option adoptée par les supports génériques MODCOM et VLE, qui assurent la résolution numérique. Ce mode opératoire n'a aucune répercussion sur le sens littéral du programme final, puisqu'il s'agit de la sous-traitance d'une opération initialement conduite au sein du programme initial. Par contre, les plateformes agronomiques DSSAT (version 4.0) et APES imposent une structure au code source des termes. Cette option revient à imposer un langage de formulation du code source. Le langage agit alors comme un filtre qui n'autorise que l'insertion des programmes à même d'en respecter la syntaxe. Ce langage imposé peut donc être réducteur en termes de possibilité d'accrétion et de développement futur par rapport à celui dans lequel les programmes sont initialement formulés.
A titre d'exemple, DSSAT v4.0 requiert la décomposition du terme sous forme de modules répartis sur deux niveaux emboités, ainsi que le découpage des modules en quatre fonctions

tel-00458265, version 1 - 19 Feb 2010

principales. Dans le même ordre d'idée, APES impose un langage de développement de l'environnement .Net (C#, …), ainsi que la répartition du code source (interface logicielle vs. composant). Tout programme qui ne peut pas être reformulé au moyen du langage imposé ne peut donc pas être inséré dans la plateforme.
En outre, prescrire une structuration de code source correspond à imposer un langage d'expression. Les travaux de Sapir-Whorf (Détrie et al., 2001) dans le cadre de la relativité linguistique montrent que deux langues ne sont jamais suffisamment semblables pour représenter la même réalité. Ainsi, même si la fonction sémantique n'est pas modifiée, la traduction du programme dans le nouveau langage est susceptible d'altérer le sens du contenu du programme. Par exemple, la perte de l'organisation relative des éléments SIMILE du programme KAMEL a modifié le sens littéral du contenu du programme en induisant une indépendance entre éléments. En regard de la théorie des systèmes dans laquelle s'inscrit le chercheur SV concepteur du programme, cette indépendance est aberrante.
De plus, la transcription du code source a des répercussions sur la poursuite du développement informatique du programme. Dans l'exemple de 'Soilwater2' inséré dans APES, la transcription implique à la fois l'adoption d'un langage de formulation (langage de programmation informatique vs. Langage d'Aide à la Modélisation), et un changement de paradigme de conception (procédural vs. objet). Confronté à la question, le chercheur SV peut adopter deux postures. Soit il n'est pas en mesure de reformuler son programme au format imposé, auquel cas il doit recourir aux services d'un informaticien pour en assurer la transcription. Soit il réécrit le programme. Dans le premier cas, l'opération aboutit à l'existence de 2 programmes : l'original qui reflète la pensée du chercheur du vivant, et le programme transcrit (le terme) qui présente une fonction sémantique similaire à celle du programme original, mais dans la limite des possibilités du langage de transcription. En corollaire, du fait de sa transcription en un langage abscond pour le chercheur SV, le terme acquiert alors le statut de « boite noire » aux yeux du chercheur SV. Cela pose en outre la question de l'actualisation des programmes au cours du temps, qui s'impose par nature à tout produit de recherche.
Dans le second cas, le chercheur SV doit s'approprier le langage et l'organisation du code imposés. Adopter un langage de formulation du code source imposé contraint alors à organiser différemment les informations contenues, et par voie de conséquence sa pensée. Une phase d'apprentissage doit donc être conduite par le chercheur SV afin d'adapter son mode de pensée au langage, aux patrons de conception, ou encore à la structure logicielle modulaire (le cas de DSSAT) imposés.

4.2 Typage d'un terme et Pragmatique

Le typage a été défini dans le paragraphe 2.4 comme l'adaptation des entrées/sorties du programme. Différentes modalités de typage ont été présentées. Pour SEAMLESS-IF, le typage correspond à la fourniture de données de sorties supplémentaires, à l'attention des autres programmes (§ 3.2.3). Pour APES, deux types de typage ont été évoqués (§ 3.2.2) : (i) la mise à disposition des variables à l'attention de MODCOM afin que celui-ci puisse effectuer l'intégration numérique, et (ii) la mise à disposition d'un même jeu d'entrée/sortie du terme « Soilwater2 » en regard de celui du terme « Soil-water ». Pour DSSAT (§ 3.2.1), les versions 2.0, 3.0 et 3.5 requièrent l'adaptation des entrées/sorties des termes en regard respectivement de leur organisation au sein des fichiers de la plateforme agronomique, du standard de données défini au niveau de la plateforme agronomique, et du standard de données défini en regard des autres termes. Dans la version 4.0 de DSSAT enfin, le typage correspond à l'échange dans ALC du module sol par un module commun à tous les termes.

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

4.2.1 Les deux formes de typages

Deux formes de typage se distinguent dans ces plateformes agronomiques. La première forme correspond à la fourniture de données de sorties supplémentaires. C'est le cas pour SEAMLESS-IF, pour le (i) d'APES (§ ci-dessus), et pour les versions 2.1, 3.0 et 3.5 de DSSAT. Cette opération est une intervention technique sur le code source du terme, qui n'a aucune répercussion sur son fonctionnement ni sur son sens littéral. La seconde forme de typage correspond à la transformation des données de sorties. C'est le cas pour la version 4.0 de DSSAT et pour le (ii) d'APES (§ ci-dessus). Dans le cas d'APES et pour le doublon « Soilwater » et « Soilwater2 », bien que la désignation des variables soit identique, on observe des différences d'unités dans l'expression des données au sein des métadonnées (m^3 d'eau/m^3 de sol pour 'Soil_water' et en kg d'eau/kg de sol pour 'Soilwater2', page 37). A titre de rappel, les fonctions de conversion des données d'entrée/sortie ont dû être rajoutées dans l'interface logicielle. Celles-ci utilisent des données internes au composant. La substitution du terme « Soilwater » par « Soilwater2 » dans une séquence a également demandé le recalcul des paramètres des autres termes de la séquence. Suite à l'échange des modules sol dans DSSAT (version 4.0), on n'a pas de référence sur l'utilisation éventuelle de fonctions de conversion. Par contre, on sait que les paramètres du terme ont été recalculés. Un outil d'analyse de sensibilité permet d'ailleurs d'en contrôler les effets sur l'ensemble des données de sortie produites par le terme.

4.2.2 Discussion

Ainsi, pour ces deux plateformes agronomiques, bien que les termes échangés (composants pour APES ou modules sol pour ALC) disposent d'un même sens littéral, leur fonction sémantique diffère, et modifie celle des termes auxquels ils sont associés. D'un point de vue pratique, cet échange contraint à recalculer les paramètres des équations des termes précédemment en place dans la chaîne, en vue du respect de la fonction sémantique de l'assemblage. D'un point de vue théorique, le calcul des paramètres, effectué en regard d'une valeur imposée par un terme, introduit une relation d'interdépendance artificielle entre les termes. Assumer le fait de changer la valeur des paramètres revient également à remettre en question la fonction sémantique des autres données, qui elle appartient aux concepteurs SV (domaine de validité, cohérence régionale etc.). Ce mode opératoire « à l'aveugle » requiert donc obligatoirement l'aval des concepteurs.

D'autre part, utiliser une fonction de conversion des entrées/sorties équivaut à adjoindre une fonction de transcription des données. Comme il a été mentionné en § 4.1, l'exercice de transcription consiste à adopter un autre langage d'expression. Du point du sens littéral, cette conversion correspond à un alignement du vocabulaire entre les deux composants. Si on prend l'exemple de la « teneur en eau », cette variable correspond effectivement au même concept hydrologique dans les termes Soilwater et Soilwater2, à savoir la quantité d'eau contenue dans de sol à un moment donné. Par contre, ce n'est plus le cas pour la variable intitulée « teneur en eau à saturation », définie pour les deux termes comme la limite maximum de teneur en eau du sol, et dont les valeurs diffèrent d'un terme à l'autre. Dans le cas de 'Soilwater', cette limite correspond à la capacité maximale de rétention en eau du sol (capacité au champ) alors que pour 'Soilwater2', elle correspond à la quantité maximale d'eau que le sol peut comporter à un instant donné, la quantité maximale d'eau étant supérieure à la capacité au champ. Ainsi, bien que le concept de saturation soit conforme à chaque terme, il diffère d'un terme à l'autre. Les variables représentent donc un concept hydrologique

différent. Dans le processus de transcription, il a donc été nécessaire d'identifier un équivalent sémantique à « teneur en eau à saturation », à partir des variables de « Soilwater2 », qui approche au mieux ce concept hydrologique spécifique à « Soilwater ». La métadonnée est donc valide par rapport au programme, mais c'est le paradigme des sciences du vivant adopté qui fait la différence. ***Ainsi, bien que le sens littéral des variables soit identique, leur sens implicite diffère.***

5 Conclusion de la partie A

L'analyse présentée ci-dessus montre que les problèmes de sémantique dans l'assemblage de termes au sein d'une plateforme agronomique émergent au niveau du vocabulaire (degré de valence des termes) ainsi que lors de la mise en coïncidence des entrées/sorties. Les problèmes de pragmatique émergent en lien avec les solutions techniques d'assemblage adoptées : formes d'interrelation permises, méthode de couplage, ainsi que pour la mise en coïncidence des entrées/sorties des termes.

La trame DRS présentée au §1.3 permet d'analyser les supports génériques (§ 2) et les plateformes agronomiques (§3). L'analyse des supports génériques permet de discuter de l'effet des règles R1 et R3 adoptées en regard de la représentation de systèmes complexes. L'analyse des plateformes agronomiques permet de détailler la règle R2, pour la mise en coïncidence des entrées/sorties entre les termes. Dans DSSAT, l'option retenue a consisté à éluder la question des connaissances pragmatiques en imposant un terme particulier, le module sol commun. Dans APES, un langage commun de formulation des termes a été imposé. Ce faisant, des termes ont été retranscrits, avec pour effet une perte de sens littéral des éléments constitutifs du terme. Par ailleurs, l'exemple de l'échange de composants sol dans APES et le traitement de la pluridisciplinarité dans SEAMLESS-IF illustrent les carences des R2 adoptés vis-à-vis du sens explicite des entrées/sorties des termes.

Dans cette étude, nous avons étudié les aspects morphologiques, syntaxiques et sémantiques de l'assemblage des programmes. Dans la suite de l'étude, nous cherchons à accéder à l'aspect pragmatique, et en particulier au sens implicite des entrées/sorties des termes. L'hypothèse sous-jacente à cette approche est que disposer du sens implicite permettrait (i) d'automatiser l'opération de mise en coïncidences des entrées/sorties et (ii) de réutiliser le principe de métadonnée et d'ontologie.

tel-00458265, version 1 - 19 Feb 2010

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

tel-00458265, version 1 - 19 Feb 2010

Chapitre B
Accès au sens implicite

La question que nous abordons dans ce chapitre est celle de l'identification d'une méthode formelle permettant de décrire un terme (ou programme informatique) et d'accéder aux connaissances pragmatiques (Sabah, 1988). Les connaissances pragmatiques permettent d'accéder à deux formes de sens. Le "sens complet" résulte de l'interprétation du sens littéral (de l'énoncé) à partir des « connaissances du monde ». Ce sens considère « comment l'information apportée par l'énoncé s'intègre dans les connaissances que l'on a sur le monde de référence ». La deuxième forme de sens, la "signification" résulte de l'interprétation du sens littéral (de l'énoncé) à partir de la situation d'énonciation. Ce sens considère « dans quelle mesure le contexte d'énonciation influe sur le sens de la phrase énoncée, et comment il convient de modifier le sens littéral pour "calculer" une signification dépendante de la situation ». Le sens implicite correspond à la "signification" telle qu'énoncée par Sabah (1988).

Dans cette étude, nous nous intéresserons aux connaissances pragmatiques des programmes et des plateformes agronomiques. L'étude des formes du sens est un objet appartenant à la linguistique. L'agronomie est une science naturelle basée sur l'observation. Avant d'être transformées en un jeu d'équations numériques, ces observations sont énoncées au moyen du langage naturel. La concordance de ces deux faits conduit à considérer les travaux conduits en linguistique pour proposer un modèle d'accès aux connaissances pragmatiques d'un terme. ***Dans l'acception que l'écriture d'un terme constitue un énoncé, nous proposons d'utiliser la linguistique pour le décrire.***

1 Bibliographie linguistique

1.1 La question de la sémantique dans la production d'un énoncé

La linguistique pragmatique distingue trois niveaux graduels d'analyse du sens d'un énoncé (Sabah, 1988; Armengaud, 2007). Le *niveau syntaxique* s'adresse aux mots du point de vue morphologique et à leurs agencements relatifs d'après des règles de grammaire pour constituer une phrase, c'est-à-dire l'élément minimal de l'énoncé (Benveniste, 1966). Le *niveau sémantique* traite du sens littéral des mots, et des conditions de vérités induites par leur combinaison en vue de produire un énoncé. Ce niveau considère la cohérence de l'association des règles d'inférence reliant un ensemble de prémisses à la conclusion. Le *niveau pragmatique* enfin appréhende le sens communiqué des mots et de l'énoncé dans le contexte dans lequel il a été exposé. Le niveau pragmatique établit la relation entre ce que l'énoncé signifie intrinsèquement et ce que son emploi signifie effectivement ou véhicule (Récanati, 2007).

Le contexte comporte les éléments descriptifs de la situation de locution nécessaires à la juste compréhension de ce qui est dit (Armengaud, 2007). Deux approches de la notion de contexte sont considérées. La première, décrite par Sarfati (2002) et Armengaud (2007), appréhende le contexte selon une vue « informelle et qualitative ». Selon ce point de vue, les auteurs

distinguent (i) le contexte circonstanciel (identité des interlocuteurs, l'environnement physique, le lieu et le temps où la locution est produite), (ii) le contexte situationnel (éléments relatifs à l'environnement culturel de l'action définissant les critères locaux de validité de la locution), (iii) le contexte interactionnel (séquence d'enchaînement des locutions dans une séquence inter-discursive), et (iv) le contexte présuppositionnel (croyances et valeurs communes aux locuteurs). Sarfati (2002) rassemble ces contextes en deux groupes : le contexte « restreint » comportant les circonstances et les éléments tangibles qui environnent l'action et réunissant les contextes circonstanciel et interactionnel, et le contexte « étendu » comportant les données à priori intangibles de la situation et réunissant les contextes situationnel et présuppositionnel.
La seconde approche, proposée par Stalnaker (1970), considère l'évolution concomitante du contexte avec le discours : « Il est l'effet des actes de langages antérieurs et la cause des actes de langages ultérieurs ». Cette approche repose sur la notion de monde possible dans lequel le contexte étend « des alternatives d'univers au-delà du monde réel ». Sa description rassemble les informations supposées partagées par tous les protagonistes : l'objet de l'action, les faits qui apparaissent au cours de son déroulement, les relations causales entre situations locales apparues durant le déroulement de l'action, etc. (Stalnaker, 1998). D'après l'auteur, cette approche permettrait de disposer d'une version unifiée de la notion de contexte

Vis-à-vis de la problématique d'assemblage, le caractère pluridisciplinaire et pluri-équipe implique de considérer le contexte d'élaboration du terme. Dans la suite du document, un terme est représenté par un couple (contexte d'élaboration, énoncé du terme).

1.2 L'interrogation dans la production d'un énoncé

Le langage naturel résulte d'une longue et lente évolution de la pensée humaine établie d'après la perception que les hommes ont de leur environnement (hypothèse de Sapir-Whorf d'après Détrie et al., 2001). Afin d'être en accord avec les pluralités ambiantes de perception du monde, le langage naturel est en perpétuelle évolution, qui se traduit par des variations phonologiques, lexicales et syntaxiques (Gougenheim, 1949 ; Martin, 2002). Pour notre objet de thèse, produire une description à partir du langage naturel inféoderait la description à un mode de pensée particulier. De façon constante par contre, le langage naturel permet trois opérations fondamentales : décrire le monde, interroger sur le monde et changer le monde (Benveniste, 1966). L'interrogation, située entre l'assertion et l'intimation, occupe une place prépondérante dans les relations humaines (Kerbrat-Orecchioni, 1991). En effet, « l'interrogation suppose toujours une ignorance de la part du locuteur (réelle ou feinte) et une tension vers un état de connaissance » (Martin, 1985). ***L'interrogation est par conséquent l'opération fondamentale qui permet de décrire le terme***.

Trois types d'interrogation sont permis par la langue française, global pour obtenir les conditions de vérité (réponse vrai/faux), alternatif pour proposer un choix de réponses (lequel, etc.) et partiel pour renseigner sur l'inconnu. Le type partiel porte sur un membre précis de l'énoncé et requiert en réponse un ensemble de mots formant une seule unité, le groupe fonctionnel. En mode direct, l'interrogation partielle se construit en remplaçant le groupe fonctionnel inconnu par un mot (i) dont la sémantique est partagée par tous les interlocuteurs et (ii) qui appelle une réponse spécifique (Kerbrat-Orecchioni, 2001). Par exemple, à la question « Où es tu ? », le groupe fonctionnel énoncé dans la réponse peut être « à la maison ».

tel-00458265, version 1 - 19 Feb 2010

Sept mots, pronoms ou adverbes, sont employés en Français, qui suffisent à questionner le monde (Gosselin, 1990). Le pronom « Qui » interroge sur l'identité, la détermination d'une personne (TLFi, 2009). Cette personne est de nature animée, fondamentalement un humain ou assimilée à l'humain (Gougenheim, 1950). Le pronom « Quoi » est porteur du genre inanimé (TLFi, 2009) et « interroge sur la nature, la détermination de quelque chose » (Larousse, 2003). Le pronom « Que» interroge sur quelque chose situé en complément d'objet direct ou attribut (Larousse, 2003). Les adverbes « Où », « Quand », « Comment », et « Pourquoi » requièrent respectivement en réponse un complément circonstanciel de localisation, de temporalité, de manière ou de raison (Gosselin, 1990). Vis-à-vis d'une phrase (ou action), les pronoms Qui, Quoi, Que interrogent respectivement sur les actants, le verbe, et les compléments d'objet. Touratier (2001) différencie le complément d'objet qui complémente le verbe, appelé alors complément essentiel, de celui qui informe des circonstances ou complément circonstanciel. Le verbe et le complément essentiel constituent l'Acte (Mélis, 1983).
L'adjonction d'une préposition (à, pour, de, depuis, par, jusque, vers, etc.) au mot interrogatif permet de faire varier la portée de la question. Par exemple, à une question initiée avec l'adverbe « quand », la réponse attendue est un instant précis. L'adjonction de la préposition « depuis » transforme alors le statut de la réponse : bien que ce soit un instant qui soit demandé, elle révèle une prolongation dans le temps de l'action (Maingueneau, 1994).

Selon Martin (2002), la langue « repose sur le principe de classification et de hiérarchie ». Les types de réponse possible à une interrogation amènent donc à proposer une structure d'organisation pour la description d'un terme.

1.3 L'interrogation et la structure de réponse associée

Dans la littérature, la structure d'organisation du groupe fonctionnel correspondant au « Où » est donnée par Hadermann (1993), dans le cadre d'une analyse morphosyntaxique. Celle correspondant au « Pourquoi » est donnée par Korzen (1986). La littérature fournit les structures du groupe fonctionnel correspondant à la manière (El Hasnaoui, 2008). Pour le « Qui », c'est la notion de personne qui est discutée (Jacquesson, 2008) ainsi que le concept d' « Actant » (Lazard, 1994). Pour le « Quand », c'est le concept de temporalité qui est étudié selon différentes postures (Maingueneau, 1994 ; Fillmore, 1997 ; Declerck, 1997 ; Grenier, 2001 et Mascherin, 2008). Pour le « Que » et le « Quoi » enfin, c'est le verbe qui est l'objet de l'analyse (Tesnière, 1988 ; François, 1989 ; Dubois et Dubois-Charlier, 1997 ; Sénéchal et Willems, 2007).

1.3.1 Où et la Localisation

L'étude morphosyntaxique de l'adverbe « Où » présente une structure hiérarchique établie à partir de deux concepts : la nature du verbe qui accompagne le Où dans la question et la possibilité de changement de lieu (Hadermann, 1993). La possibilité de changement de lieu est représentée par les nœuds [± déplacement] dans l'arborescence (Figure 19). S'il s'agit d'un verbe de localisation, la réponse apportée à la question est un complément du verbe (complément locatif). Dans le cas où le verbe est statique, comme par exemple le verbe 'demeurer', la réponse fournie correspond à un lieu permanent. Dans le cas d'un verbe de déplacement, comme par exemple le verbe 'aller', la réponse correspond à un lieu initial, médian ou final. Dans ce dernier cas, la distinction de la position (initial vs médian vs final)

tel-00458265, version 1 - 19 Feb 2010

est généralement explicitée au moyen d'une préposition (de, par, jusque, …). Dans le cas où le verbe n'est pas de localisation, la réponse est un élément scénique qui s'adjoint au verbe (adjoint locatif), par exemple 'travailler à domicile'.
D'autre part, il est des questions initiées par Où qui comportent déjà une localisation : « Où demeure-t-il à Bruxelles ? Où travaille-t-il le mieux à Nice ? ». Cette formulation, appelée double localisation, « est utilisée pour obtenir davantage de renseignements concernant le lieu qui se trouve dans la phrase interrogative. Dans les interrogations qui contiennent un verbe locatif, « Où » a la fonction de complément et l'autre locatif est adjoint. Lorsque le verbe n'est pas locatif, « où » est évidemment adjoint, tout comme l'autre locatif » (Hadermann, 1993).
(Hadermann, 1993) s'appuie sur les travaux de (Vandeloise, 1985) pour traiter de l'aspect sémantique porté par le « où ». Vandeloise (1985) distingue le toponyme, lieu géographiquement repérable ('Montpellier', par exemple), de l'objet topographique, massif et immobile qui occupe une partie de l'espace ('le pont' par exemple) et de l'élément topologique, qui permet un repérage spatial plus abstrait ('chez Pierre' par exemple). Des éléments topologiques, Vandeloise (1985) dégage cinq groupes de « traits universels qui jouent un rôle dans l'analyse des mots spatiaux » : (i) les directions déterminées par la symétrie du corps humain, (ii) les concepts de physique populaire (axe vertical, relation porteur/porté, relation contenant/contenu), (iii) l'accès physique et l'accès à la perception, (iv) la rencontre potentielle et (v) les orientations générales. Selon (Hadermann, 1993), le toponyme et l'objet topographique ont une valeur locative intrinsèque, et l'élément topologique une valeur locative extrinsèque.

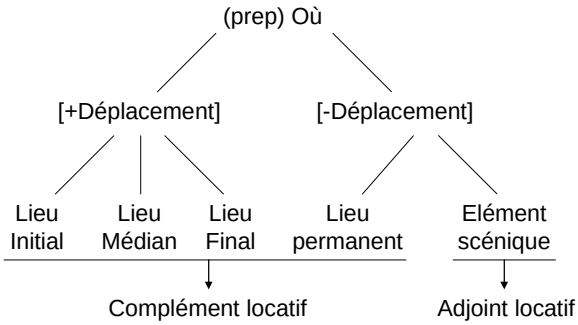


Figure 19 : Etude morphosyntaxique du Où (Hadermann, 1993).

1.3.2 Pourquoi et la Raison

Pourquoi « sert à interroger sur la cause ou sur la finalité d'une action ou d'un fait » (TLFi, 2009). Korzen (1985) distingue deux modalités d'utilisation de l'adverbe 'Pourquoi', selon que l'énoncé introduit par cet adverbe comporte une présupposition forte ou faible. Dans la première modalité, la présupposition, dite forte, correspond au cas où le locuteur et l'allocutaire présupposent ensemble de l'existence d'une cause. L'interrogation porte ainsi sur la raison qui a conduit à produire l'action. Dans ce cas, l'adverbe, appelé 'pourquoi normal', peut être remplacé par la variante marquée « Pour quelle raison ».
Dans la seconde modalité, la présupposition dite faible, correspond au cas où la cause n'existe pas dans l'univers commun au locuteur et à l'allocutaire. L'adverbe est alors un mot outil, de nature abstraite, qui a pour objet d'inciter l'allocutaire à donner une explication des raisons de l'action. Dans ce cas, l'adverbe est appelé 'pourquoi incolore'. Cette forme de 'Pourquoi' se retrouve par exemple dans la phrase 'Pourquoi n'est-il pas venu ?'. Dans cette phrase, bien

tel-00458265, version 1 - 19 Feb 2010

que le locuteur et l'allocutaire savent tous les deux que la tierce personne n'est pas venue, le premier demande une explication de l'absence.

D'après (Korzen, 1985), une des particularités du 'Pourquoi incolore' est de disposer de la capacité de modifier l'acte illocutoire. Ainsi dans l'exemple 'Pourquoi rentres-tu déjeuner ?', la réponse attendue porte d'une part sur l'explication des raisons du retour pour déjeuner et d'autre part sur l'acte illocutoire qui s'en suivra, c'est-à-dire de déjeuner. Quant au 'pourquoi normal', il est le seul à fonctionner comme point de départ (ou foyer) d'une question partielle. L'utilisation du 'Pourquoi normal' conduit à deux types de réponses, soit la cause proprement dite, soit un enchaînement de causalités (ou chaîne causale).

Par ailleurs, selon le TLFi (2009), la chaîne causale est constituée (i) d'une cause première, qui agit par elle-même indépendamment de toute autre cause, (ii) de causes secondes, qui agissent sous la dépendance d'une autre cause, et (iii) d'une cause finale, ou finalité, qui a pour origine la fin recherchée.

1.3.3 Comment et la Manière

Comment « interroge sur la manière dont une action se déroule ou un fait se présente » (TLFi, 2009). La manière est un « Aspect particulier, forme particulière que revêt un processus, une action ou un état » (TLFi, 2009). La manière est un groupe fonctionnel complexe qui intervient selon différentes formes au sein de la phrase. La formalisation retenue par El Hasnaoui (2008) utilise la notion d'incidence.

En grammaire, l'incidence est définie comme *la relation de ce qui est dit à ce qui est parlé* (Ilinski, 2003). Ce principe, énoncé par (Guillaume, 1973), met en rapport au sein de la phrase le support, qui est ce dont on parle, de l'apport, qui est ce qui est dit. Ce principe formalise d'une manière prédicative le mécanisme établissant les liens entre les mots au sein de la phrase. Le principe différencie l'incidence interne, dont la fonction est d'identifier le support, de l'incidence externe, l'apport. Le support considéré est généralement le substantif, comme par exemple 'homme'. L'incidence externe au premier degré s'adresse aux mots en relation directe avec le support, comme par exemple 'un homme beau', où beau est incident au support 'homme'. L'incidence externe au premier degré considère notamment les adjectifs et verbes conjugués. L'incidence externe de second degré opère à l'endroit où un mécanisme d'incidence est déjà en place. Dans « 'Pierre marche vite', 'vite' n'est pas incident à 'Pierre' et n'est pas non plus, comme on (pourrait) le supposait à tort incident à 'marche' ; 'vite' est incident à l'incidence de 'marche' à 'Pierre', c'est-à-dire une incidence en cours » d'après (Guillaume, 1974). Ce second degré s'applique généralement aux adverbes, même si des situations particulières sont rencontrées. Ainsi « dans l'exemple 'un homme fort beau', 'fort' n'est pas incident à 'homme' et n'est pas non plus incident à 'beau' expressément, mais est incident à l'incidence de 'beau' à 'homme'; de sorte que l'incidence du mot 'fort' est une incidence à une incidence » (El Hasnaoui, 2008). Comme le montre l'exemple 'Pierre marche vite', la mise au point de ce principe est complexifiée de par la considération simultanée des niveaux syntaxiques et sémantiques (Ilinski, 2003 ; El Hasnaoui, 2008).

Dans son analyse morphosyntaxique du circonstant de manière, El Hasnaoui (2008) différencie deux constructions syntaxiques selon leur degré d'incidence. Dans la première construction, le circonstant est incident au verbe. Les termes adoptés sont alors soit de l'ordre du faire, comme par exemple 'il rentra par curiosité', soit de l'ordre du moyen utilisé pour

tel-00458265, version 1 - 19 Feb 2010

faire, comme ‘enfoncer un clou avec un marteau’ par exemple. Cette dernière forme correspond à l’instrumentation.
Dans la seconde construction, le circonstant est incident au sujet. Cette situation s’applique lorsque le circonstant est construit au moyen de la préposition *en + gérondif*, comme par exemple ‘il travaille en chantant’, ou de la préposition *par + infinitif*, comme dans ‘j’ai fini par dire oui’. « Deux actes sont alors menés de front par le même sujet, sans lien logique entre les deux. Il s’agit d’une relation de simultanéité » (El Hasnaoui, 2008).

Par ailleurs, la manière est également étudiée par les sociolinguistes dans le cadre de l’apprentissage. Charlot, (1997) par exemple établit une typologie du savoir en utilisant deux notions discriminantes : (i) l’entendement du savoir, qui est soit tacite soit explicite, et (ii) la valeur de vérité du savoir, objective ou subjective. La combinaison de ces deux concepts permet de différencier (i) l’information, à savoir une donnée objective et explicite, (ii) du savoir, une donnée objective du fait d’une édification méthodologique et tacite dans la mesure où elle est issue d’une expérience personnelle, (iii) de la Γonnaissance (initié par Γ pour différencier ce concept de celui de Connaissance adopté dans la thèse), définie comme une donnée issue de l’expérience personnelle du sujet et empreinte de valeurs affectives.

1.3.4 Qui et l’Actant

Selon Tesnière (1988), un ‘Actant’ est un substantif subordonné du verbe qui, à un titre quelconque, participe à ‘l’Action’ (le procès). Tesnière distingue le ‘prime actant’ qui fait l’action, du ‘second actant’ qui subit l’action, du ‘tiers actant’, qui en bénéficie ou au détriment duquel se fait l’action. Tesnière associe le concept d’’actance’ au rôle syntaxique dans la phrase : le ‘prime actant’ est le sujet, le ‘second actant’ est l’objet et le ‘tiers actant’ est le complément d’objet indirect.

En introduction à son étude, Lazard (1994) rappelle les différents ‘rôles sémantiques’ joués par ‘l’Actant’ et usuellement adoptés par les linguistes : « l’agent qui accomplit l’action – le patient qui la subit – le destinataire ou attributaire d’un don, d’une parole – l’instrument, au moyen duquel elle est accomplie – etc ». Cette conception de ‘l’Actant’ se distingue de celle adoptée par Tesnière (1988) par la différenciation du rôle sémantique de la fonction syntaxique. Ainsi, dans la phrase ‘la pomme est mangée’, ‘la pomme’ (l’Actant) est le sujet de la phrase. ‘La pomme’ n’en est pas l’agent, mais le patient.
Par ailleurs, Lazard (1994) analyse les constructions syntaxiques comportant zéro, un, deux ou trois actants (sans actant, uni-actancielle, bi-actancielle et tri-actancielle). La phrase ‘Il mange la pomme’ par exemple est constituée des deux actants ‘il’ et ‘pomme’. Dans son analyse, l’auteur oppose l’agent, celui qui agit (agent, destinataire, etc.), à l’objet, « correspondant au patient dans les phrases d’action et à ceux qui sont traités de même dans les autres types de phrases ». L’agent est préférentiellement un humain ou un animé, apte à exercer une action sur l’objet. L’objet peut être défini ou non, humain ou non, etc. Dans les constructions tri actancielles, comme par exemple, ‘il lui donne la pomme’, ‘lui’ est considéré comme un agent périphérique (complément de direction). Ce formalisme permet d’appréhender une construction tri-actancielle sous une forme bi-actancielle. Dans les constructions sans actant (‘il pleut’ par exemple), l’actant (‘il’ dans l’exemple) ne peut être explicité par aucun substantif, et est dépourvu de contenu sémantique. Il est alors qualifié de « vide » et la construction est décrite sous la forme d’une uni-actancielle.

tel-00458265, version 1 - 19 Feb 2010

Lazard (1994) enfin, dans une analyse portant sur plusieurs langues, identifie cinq structures syntaxiques pour la formulation d'une construction bi-actancielle. Par structure, l'auteur entend l'équivalence grammaticale des actants entre une phrase bi-actancielle, par exemple 'Quintus tua Marcus', et une uni-actancielle 'Marcus mourut'. Pour formaliser la structure, l'auteur utilise les variables X, Y et Z définies respectivement comme l'agent, le patient et le membre unique d'une construction uni-actancielle (exemple 'il dort'). Dans la structure accusative, Z a la même forme que le X de la phrase bi-actancielle : X = Z (exemple de Quintus et Marcus ci dessus). Dans la structure ergative, le patient est traité comme actant unique : Y = Z. La structure neutre correspond au cas où chacun des deux actants a le même traitement que l'actant unique : X = Z et Y = Z. Dans la structure mixte, chacun des deux actants a une des propriétés grammaticales de l'actant unique : X ≈ Z et Y ≈ Z (pas égalité mais équivalence). Dans la structure disjointe enfin, aucun des deux actants ne partage de propriété grammaticale avec l'actant unique : X ≠ Z et Y ≠ Z.

Jacquesson (2008) analyse la perception de la notion de personne dans diverses langues. En premier lieu, l'auteur considère la personne selon un angle fonctionnel. Il distingue alors la politesse signalée par l'emploi d'un mot particulier (vous, madame, …), de l'honorifique (majesté…), de la familiarité (tu…) du tabou (mots réservés dans certaines langues à des communautés castées). De ce fait, l'adoption d'un terme a des effets sur la relation entre deux personnes, dont le langage et le comportement.
En second lieu, il traite des catégories nominales que sont le genre et le nombre. Concernant le genre, la majorité des langues indo-européennes disposent du triplet (masculin, féminin, neutre). Ce n'est pas le cas de toutes, l'arménien par exemple n'en dispose que d'une seule, le français deux. D'autres langues en comportent encore plus à l'exemple du 'mullukmulluk' (Australie) qui en comporte quatre : masculin, féminin, arbres et objets inanimés, nourritures et quelques parties du corps. Il observe enfin que, dans certaines langues, les pronoms au singulier perdent leur genre au pluriel. C'est le cas par exemple en anglais où le pluriel des pronoms 'he', 'she' et 'it' est 'they'. Concernant le nombre, diverses formes sont employées pour différencier le singulier du pluriel selon le nombre d'individus représentés et le lien que ceux-ci entretiennent. Si la personne représente un seul individu, il est qualifié de singulier. S'il représente un petit nombre d'individus formant une unité, une dénomination est octroyée en fonction du nombre : duel et triel pour deux ou trois individus respectivement. Dans les autres situations, quelque soit le nombre d'individus et leur lien, il est qualifié de pluriel. Le pluriel inclusif est différencié du pluriel exclusif selon que l'allocutaire est compris ou non dans le nombre.
En dernier lieu, Jacquesson (2008) étudie la personne d'un point de vue syntaxique. Pour la relation d'Actance entre personnes, l'auteur indique que l'ordre d'énoncé des actants s'effectue généralement selon leur importance sémantique. Cet ordre, défini par la langue, peut être décroissant, c'est à dire du « plus agentif » vers le « moins agentif » (exemple 'je te donne une pomme'), ou croissant. Pour le pronom, l'auteur met en évidence deux constructions contrastées selon les langues. Certaines langues utilisent deux séries distinctes d'indice pour désigner les personnes, une série pour le sujet (je, tu, il…) et une autre pour le non-sujet (me, te, …). D'autres langues par contre n'utilisent qu'une seule série. Dans ce cas, deux agentifs ('je te' par exemple) sont réunis en un seul morphème. Cette variation linguistique, appelée forme sagittale par Claude Hagège (2001), met en évidence l'existence d'une relation interpersonnelle lorsque plusieurs personnes sont réunies dans une phrase.

tel-00458265, version 1 - 19 Feb 2010

1.3.5 Quand et la Temporalité

La notion de temporalité est source de questionnement sémantique. Comme le souligne (Fillmore, 1997), la question est de savoir si c'est le temps qui se déplace ou si c'est nous qui nous déplaçons par rapport à lui. Par rapport à cette question, Mascherin (2008) distingue trois types de perception de la temporalité. En premier lieu, le « temps universel » s'écoule de façon linéaire, immuable et perpétuelle en dehors de l'existence de l'homme. En second lieu, le temps « conventionnel », partagé par une même communauté d'individus, est mesuré au moyen d'outils (calendrier, horloge, etc.) établis sur la base de phénomènes naturels observés (rotation de la terre, etc.). Le « temps individuel », enfin, correspond à une perception personnelle du temps conventionnel, comme par exemple « trouver le temps long ». Ces trois types de perception du temps physique considèrent la temporalité sous la forme d'une durée indépendante des évènements qui ont lieu. Elle se distingue du temps historique, « où le contenu de chaque instant dépend du contenu de chacun des instants qui l'ont précédé » (Grenier, 2001).

L'adverbe Quand « interroge sur le moment (passé, présent ou à venir) ou la période (heure, jour, date, époque...) où se situe une action ou un état » (TLFi, 2009). En anglais, Declerck (1997) distingue deux formes de temporalité (Figure 20) selon qu'elle s'inscrit ou non dans la durée. Celle s'inscrivant dans la durée est délimitée par un début et une fin. L'autre, sans durée, s'inscrit dans l'intemporalité. Par ailleurs, en français, Maingueneau (1994) utilise trois notions que sont le point, la durée et la répétition afin de différencier les points de vue selon lesquels le temps est considéré (les visées temporelles). Le point correspond à une mesure de temps, la durée à une portion discrète ou continue, et la répétition à la réitération. Maingueneau (1994) fait ensuite correspondre chacune des possibilités offertes à une question redevable d'une donnée temporelle : à (i) l'itération, il associe la question 'combien de fois', (ii) à la durée les questions 'depuis/pendant/dans combien de temps ?', et (iii) au point la question 'quand ?' (Figure 20). Par rapport à la construction sous-tendue par cette figure, l'auteur souligne l'absence de la combinaison du point et de la durée à laquelle il fait correspondre les questions 'depuis quand ?' et 'jusqu'à quand ?'.

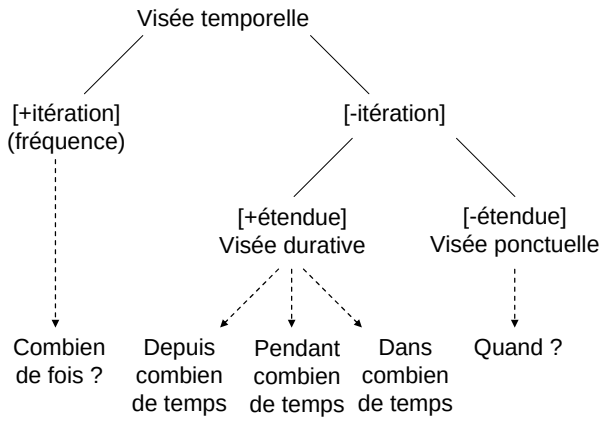


Figure 20 : Formes du temps selon Maingueneau (1994).

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

1.3.6 Quoi/Que et acte

L'acte est la réunion du verbe et du complément essentiel. Le verbe « exprime un procès, un état ou un devenir, variant en nombre, en personne et en temps et ayant pour fonction syntaxique de structurer les termes constitutifs de l'énoncé » (TLFi, 2009). Dans le cadre de la syntaxe structurale, (Tesnière, 1988) appréhende le verbe comme le descripteur d'un état ou d'une action. Le verbe d'état exprime une manière d'être caractérisée par une qualité ou une position comme par exemple 'être bon' ou 'être loin'. Le verbe d'action exprime une activité comme par exemple 'tomber' ou 'faire fureur'. D'après l'auteur, « l'opposition fondamentale n'est pas entre l'idée du verbe **être** et celle du verbe **avoir**, mais bien entre l'idée du verbe **être** et celle du verbe **faire** ». Il aurait d'ailleurs observé que, dans plusieurs langues, de nombreux verbes d'action sont constitués du verbe de base 'faire'.
Par ailleurs, Tesnière (1988) qualifie le verbe selon le nombre d'actants qui lui sont adjoints : la valence du verbe. Les verbes avalents ne requièrent pas la présence d'un actant, et leur valence est alors de zéro. Dans ce cas, le verbe est uniquement employé à la troisième personne du singulier. Il permet notamment de décrire un état météorologique ('il pleut' par exemple). Pour les autres verbes, la valence prend pour valeur 1 ('je cours'), 2 ('je mange une pomme') ou 3 ('je te donne une pomme').

Plusieurs classifications sont proposées pour les verbes. Ainsi, François (1989) distingue six indicateurs de discrimination des verbes : « [±**Agentif**] indique un contrôle de l'action par un animé, [±**Causatif**] l'action est conçue par le locuteur comme non autonome c'est-à-dire comme résultant d'un évènement autonome ou d'une action, [±**Transitionnel**] l'action est perçue comme la transition durant la durée de temps considérée (« t » de François, 1989), [±**Dynamique**] l'action est perçue comme se déroulant dans le temps, [±**Telique**] l'action est bornée, et [±**Momentané**] l'action est indivisible en phase successives ». L'auteur combine ensuite ces indicateurs au sein de typologies pour décrire les notions d'état, d'évènement, de processus, d'action, d'activité, de transition et de changement. Trois typologies distinctes sont produites (Figure 21). Chacune d'elle fait correspondre à un sous-ensemble de combinaisons d'indicateurs un sous-ensemble de notions. Dans ces typologies, une combinaison qui ne correspondrait pas à une des sept notions est notée ∅ (ensemble vide).

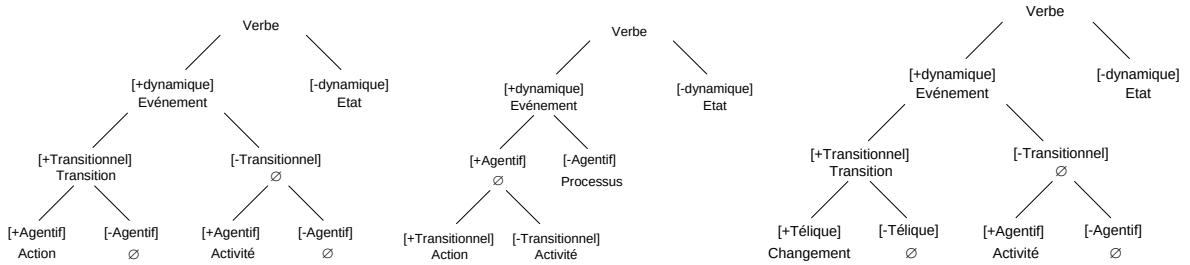


Figure 21 : Typologie du verbe selon (François, 1989).

Mélis (1983), quant à lui, considère que la classification des verbes doit s'effectuer selon leur emploi, et non sur la seule base d'éléments lexicaux. Cette posture l'amène à distinguer les verbes d'actions des verbes d'état au moyen de trois critères. Le premier critère consiste à identifier si le verbe à classer peut être suppléé par le verbe 'faire' ('il mange une pomme et il le fait avec appétit' par exemple). Le second critère utilise la périphrase 'être en train de', de façon préposée au verbe ('les légumes sont en train de cuire sur le feu' par exemple). Le

dernier critère est la possible combinaison du verbe avec les verbes ‘pouvoir’ ou ‘devoir’. Ce dernier critère correspond à l’expression de l’obligation, de l’intention ou de la suggestion. Cette décomposition l’amène à considérer le verbe ‘être’ comme un verbe d’action lorsque celui-ci est précédé d’un sujet humain, et que son utilisation correspond au dernier critère (‘Pierre est à la maison’ par exemple).

Enfin, Dubois et Dubois-Charlier (1997) proposent une classification syntaxico-sémantique des verbes en utilisant un automatisme. La base de données utilisée est celle du dictionnaire électronique des ‘Verbes Français’. Ce dictionnaire comporte 25610 entrées verbales représentant 12310 verbes différents, certains des verbes comme ‘passer’ comportant 61 entrées. L’hypothèse adoptée pour la catégorisation « repose sur l’adéquation entre les schèmes syntaxiques de la langue et l’interprétation sémantique qu’en font les locuteurs : à la différence syntaxique entre ‘venir à Paris’ et ‘venir de Paris’ correspond une différence sémantique entre la destination et l’origine ». Le dictionnaire comporte pour chaque entrée verbale : le sens donné par un synonyme, le domaine d’emploi en général (géologie, psychologie, etc.), la conjugaison et l’auxiliaire, la syntaxe (intransitif, transitif direct ou indirect, pronominal, nature des sujets, etc.), les dérivations (noms d’action, d’instrument, d’agent, etc.), des exemples d’emploi, etc.
Chaque entrée verbale est caractérisée au moyen d’opérateurs, selon son emploi syntaxique. Plus de 80 opérateurs sont utilisés « pouvant être combinés avec des notations portant sur la quantité, la qualité, la répétition et la forme de la proposition » (François et al., 2007).
La classification des entrées verbales est effectuée par réunion graduelle des opérateurs. Cinq niveaux de rassemblement ont été obtenus. Le premier niveau, appelé générique, est constitué de 14 classes référencées : communication (C), don et privation (D), entrée et sortie (E), frapper et toucher (F), états physiques et comportements (H), locatif (L), mouvement sur place (M), munir et démunir (N), verbes psychologiques (P), réalisation et mise en état (R), saisir, serrer et posséder (S), transformation et changement (T), union et réunion (U) et verbes auxiliaires (X). Le second niveau (sémantico-syntaxique), discrimine selon les oppositions ‘être vivant / non animé’ et ‘propre / figuré’. Le troisième niveau correspond au schème syntaxique et leur paradigme lexical. La considération des diverses formes syntaxiques puis des variantes syntaxiques et lexicales s’effectue aux niveaux quatre et cinq. A titre d’illustration, le Tableau 8 donne la classification du mot ‘aboyer’ selon cette méthode.

Tableau 8 : Description de l’entrée verbale ‘aboyer’, d’après (François, 2007). La colonne de droite, intitulée Opérateurs, comporte les opérateurs utilisés pour la classification du mot. Dans (François, 2007), les opérateurs ne sont pas décrits, à l’exception de l’opérateur ‘loq’ qui statut de la présence d’un complément nominal.

Catégorie	Niveau	Codage	Interprétation	Entrées	Opérateurs
Classe générique	5	C	« communication »	2039	dic ; ind ; mand ; f.son ; loq ; f.cri
Classe sémantico-syntaxique	4	C1	« s’exprimer par un son, une parole »	1059	f.son ; loq ; f.cri
Classe syntaxique	3	C1a	« émettre un cri », humain ou animal	232	Loq *(objet interne & en V-ant)* f.cri *(objet interne & en V-ant)*
Sous-type syntaxique	2	C1a/3	« émettre un type de parole »	75	Loq *(objet interne & en V-ant)* f.cri *(sujet hum)*
Variante syntaxique	1	[A16 T1300]	Type « cris »	27	f.cri *« animal »* (sujet hum)
Entrée **Aboyer** 02			« hurler, crier après » *on a ~ contre les voisins*	1	f.cri chien

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

tel-00458265, version 1 - 19 Feb 2010

Cette classification conduit à l'établissement d'une synonymie syntaxique des entrées verbales. Le degré de synonymie de deux verbes est spécifié par le niveau de regroupement des classes d'inclusion des verbes.
Dans le cadre de l'analyse des verbes locatifs trivalents, Sénéchal et al. (2007) relève que la construction proposée par Dubois et Dubois-Charlier (1997) ne tient pas compte des aspects sémantiques et, de ce fait, ne permet pas de lever les problèmes de polysémie : « les emplois d'un même verbe sont dissociés dès lors qu'ils ne présentent pas les mêmes schèmes syntaxiques ». Pour la prise en compte de l'aspect sémantique, les auteurs proposent de « distinguer trois cas de figure : (1) les emplois verbaux qui peuvent être décrits comme des alternances syntaxiques sans impact sur le sens, et qui ne mettent pas en cause la monosémie des entrées ('les moustiques pullulent dans le marais' et 'le marais pullule de moustiques' par exemple) ; (2) les emplois syntaxiques différents présentant des extensions de sens régulières et systématiques, et que nous qualifierons de polysémiques ('apporter un café à quelqu'un' qui étend le verbe de déplacement 'apporter' au don par exemple); (3) les emplois syntaxiques différents présentant des différences non systématiques et irrégulières dans lequel les auteurs voient (provisoirement) des cas d'homonymies ».

2 Vers une structure de graphes

Les auteurs précédemment cités ont produit des typologies relatives à diverses notions. L'objet de ce paragraphe est de récapituler et de transcrire les typologies déclarées sous forme de graphe.

2.1 Exploitation de la littérature et formulation de graphes

2.1.1 Où et la Localisation

A partir des travaux d'Hadermann (1993), on peut dissocier le complément de l'adjoint locatif. Dans le cas du complément locatif, l'auteur introduit la notion de déplacement selon la Figure 22a. Les règles syntaxiques énoncées permettent de formuler le changement locatif en termes d'itinéraire ('de Montpellier à Rome en passant par Paris' par exemple). Au sein de l'itinéraire, les différents lieux décrits par l'auteur (initial, médian et final) sont ordonnés (Figure 22b). L'adjoint locatif prend son importance dans le cas d'une localisation double. Dans ce cas, le groupe fonctionnel est constitué d'un premier lieu dont on parle, et d'un second lieu, référentiel du premier. Le lieu dont on parle peut être soit permanent, soit en déplacement. La relation entre le lieu dont on parle et le référentiel correspond alors une relation d'inclusion (Figure 23).
La discrimination faite par Vandeloise (1985) puis Hadermann (1993) en regard du caractère intrinsèque/extrinsèque du lieu considéré, permet d'établir une typologie de la sémantique du lieu selon qu'il s'agit d'un toponyme, d'un objet topographique, ou d'un élément topologique (Figure 24). Dès lors qu'il s'agit d'un élément topologique enfin, la relation au référentiel du lieu considéré peut être décrite en utilisant les « cinq traits universels » identifiés par Vandeloise (1985). Ces traits correspondent à une relation de voisinage qui situe le lieu considéré (référé) relativement au référentiel (Figure 25).

tel-00458265, version 1 - 19 Feb 2010

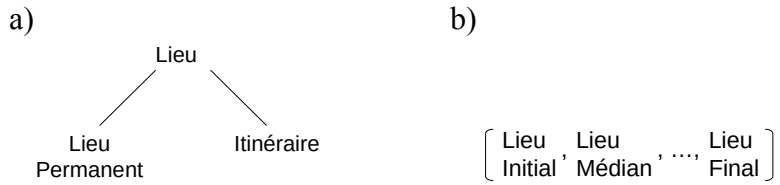


Figure 22 : Typologie du lieu (a) et itinéraire (b).

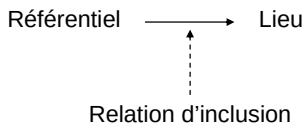


Figure 23 : Relation entre le lieu et son référentiel dans le cas d'une double localisation.

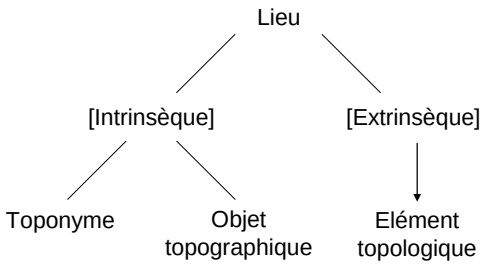


Figure 24 : Sémantique de la localisation.

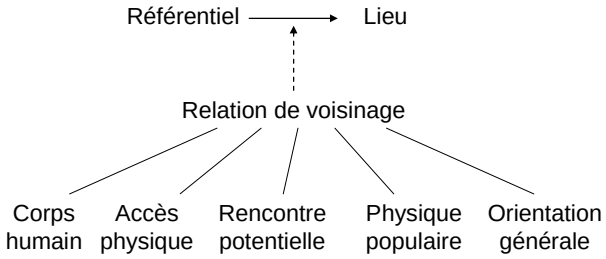


Figure 25 : Relation de voisinage pour accéder au lieu dans le cas où celui-ci est un élément topologique.

2.1.2 Pourquoi et la Raison

Le 'pourquoi normal' a pour objet de produire un questionnement de type partiel, c'est-à-dire sur un membre précis de l'énoncé au même titre que les autres mots interrogatifs (Korzen, 1985). La réponse fournie est soit une cause, soit une chaîne causale (Figure 26a). D'après le TLFi (2009), les éléments constitutifs de la chaîne causale sont la cause première, les causes secondes et la finalité (Figure 26b). L'objet du pourquoi incolore est d'obtenir une explication des raisons de l'action en vue de les comprendre. La réponse peut également avoir des répercutions sur l'acte illocutoire (Figure 27).

tel-00458265, version 1 - 19 Feb 2010

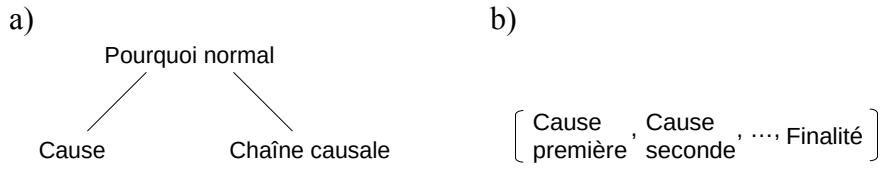


Figure 26 : Typologie des réponses obtenues par l'usage du 'pourquoi normal' (a) et constitution d'une chaîne causale (b).

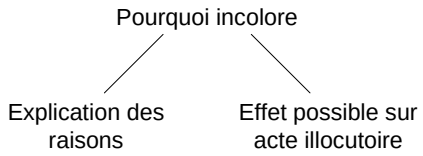


Figure 27 : Typologie des réponses obtenues par l'usage du 'pourquoi incolore'.

2.1.3 Comment et la Manière

Pour son analyse de la 'Manière', El Hasnaoui (2008) utilise le principe d'incidence pour distinguer les deux modalités d'usage du 'Comment'. Lorsque le circonstant est incident au verbe, les termes utilisés sont soit de l'ordre du faire, i.e. la « façon » (de faire) adoptée pour conduire l'action indiquée par le verbe, soit de l'ordre du moyen (ou instrument) utilisé pour faire (Figure 28).
Par ailleurs, en sociolinguistique, Charlot (1997) différencie plusieurs formes du savoir à partir de la valeur de sa vérité (objectif vs. subjectif) et de son entendement (tacite vs. explicite). La combinaison de ces critères permet de dresser une typologie des formes du savoir (Figure 29) distinguant information, savoir et Γonnaissance.

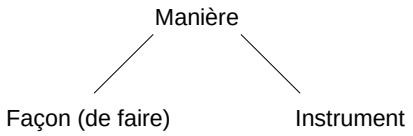


Figure 28 : Typologie de la manière lorsque le circonstant est incident au verbe. D'après (El Hasnaoui, 2008).

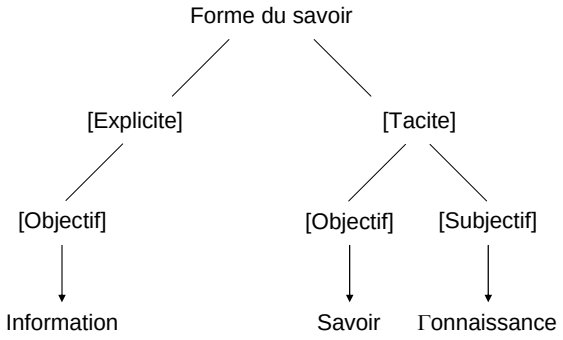


Figure 29 : Typologie des formes du savoir, d'après (Charlot, 1997).

tel-00458265, version 1 - 19 Feb 2010

2.1.4 Qui et l'Actant

D'après Lazard (1994), les linguistes distinguent plusieurs types d'actants au sein d'une construction syntaxique : l'agent, le patient, le destinataire, l'instrument, etc. Pour sa part, il distingue les deux rôles sémantiques majeurs d'agent et d'objet (Figure 30a). L'agent est préférentiellement un humain ou un animé. L'objet peut être défini, humain, etc. Dans le cas d'une construction tri-actancielle, la considération du second agent sous la forme d'un complément de direction transforme la construction tri-actancielle en bi-actancielle (Figure 30b).

Les travaux de Jacquesson (2008) portent sur le concept de personne, qu'il appréhende selon plusieurs angles. Sous l'angle fonctionnel, c'est-à-dire la façon dont une personne est désignée par une autre, il différencie la politesse, la familiarité, l'honorifique, du tabou (Figure 31). Pour le nombre, Jacquesson (2008) propose une organisation logique en regard du nombre d'individus représentés et des relations que ceux-ci entretiennent (Figure 32). Si la personne représente un seul individu, elle est qualifiée de Singulier. Si elle représente une unité constituée de moins de 4 individus, elle est qualifiée de duel ou de triel selon qu'elle réunisse 2 ou 3 individus. Dans le cas où il n'y a pas unité des personnes, ou si leur nombre est supérieur à 3, elle est qualifiée de plurielle. Le pluriel est inclusif dans le cas où l'allocutaire fait partie de l'ensemble, et exclusif dans le cas contraire. Enfin, la désignation des personnes par un pronom amène à considérer deux constructions syntaxiques différenciant le sujet du non-sujet selon le nombre de séries d'indices utilisé. Dans le cas où une seule série est utilisée, un morphème réunit les deux personnes (i.e. la forme sagittale) qui met en évidence l'existence d'une relation entre les personnes (Figure 33).

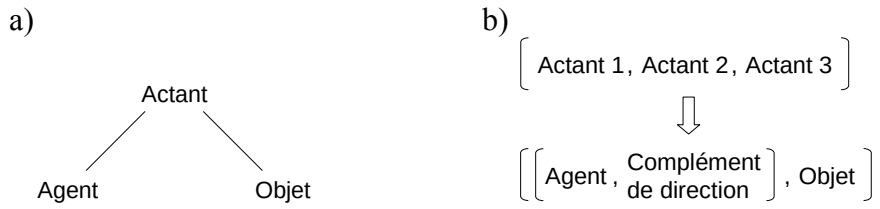


Figure 30 : Deux types d'actants (a) et transformation d'une construction tri-actancielle en bi-actancielle (b), d'après (Lazard, 1994).

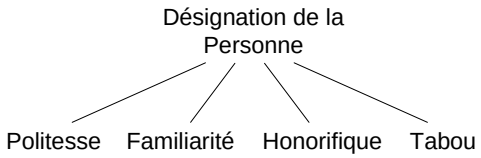


Figure 31 : Typologie de la désignation d'une personne.

tel-00458265, version 1 - 19 Feb 2010

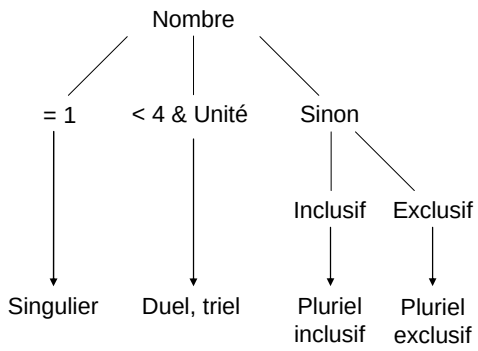


Figure 32 : Typologie du nombre.

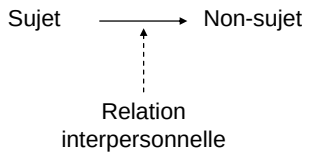


Figure 33 : Forme sagittale.

2.1.5 Quand et la Temporalité

La notion de temporalité est appréhendée selon plusieurs points de vue linguistiques. Pour caractériser la réponse à un questionnement sur la temporalité, Maingueneau (1994) utilise les notions de point, de durée et d'itération (Figure 34). Declerck (1997) par ailleurs, distingue deux formes de temporalité, l'une intemporelle, et l'autre qui s'inscrit dans la durée délimitée par un début et une fin (Figure 35a et 37b). Mascherin (2008) a identifié trois types de temps : le temps universel qui s'écoule en dehors de l'homme, le temps conventionnel partagé par une même communauté d'individus et le temps individuel correspondant à une perception personnelle de l'écoulement du temps (Figure 36). Enfin, Grenier (2001) différencie le temps physique du temps historique « où le contenu de chaque instant dépend du contenu de chacun des instants qui l'ont précédé » (Figure 37). Cette notion renvoie à la différence entre date calendaire et évènement. En ce sens, la distinction proposée par Grenier (2001) relève de la sémantique.

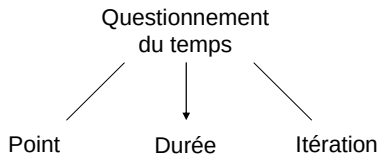


Figure 34 : Le questionnement du temps.

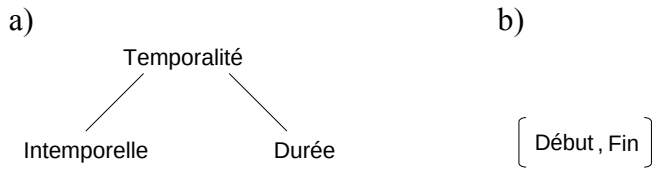


Figure 35 : Forme de la temporalité (a) et limites de la durée (b), d'après (Declerck, 1997).

tel-00458265, version 1 - 19 Feb 2010

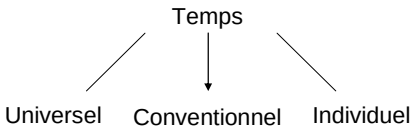


Figure 36 : Typologie des temps, d'après (Mascherin, 2008).

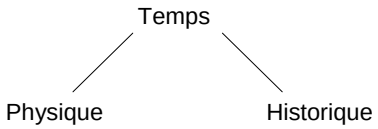


Figure 37 : Logique relative des temps (Grenier, 2001).

2.1.6 Quoi/que et acte

L'acte réunit le verbe et le complément essentiel. Plusieurs classifications des verbes sont proposées dans la littérature. Tesnière (1988) définit le verbe comme un descripteur d'état ou d'une action, et différencie l'état de l'action au moyen des verbes être et faire (Figure 38). Mélis (1983) propose une classification tripartite des verbes d'action au moyen des trois critères que sont le remplacement par le verbe 'faire', l'adjonction de la périphrase 'être en train de', et la combinaison avec les verbes 'pouvoir' et 'devoir' (Figure 39). Le verbe 'être' est un verbe d'action dès lors que son utilisation correspond au dernier critère ('Pierre est à la maison' par exemple). Dubois et Dubois-Charlier (1997) propose une classification des verbes d'après les usages syntaxiques recensés. La classification rassemble les verbes « synonymes » d'un point de vue syntaxiques selon cinq niveaux emboités. Le premier niveau différencie quatorze classes génériques (Figure 40) dans laquelle la distinction état vs action avancée par Tesnière (1988) ne se retrouve pas. La variation d'emploi d'un verbe peut entraîner son insertion simultanée dans diverses classes indépendamment des classes génériques.

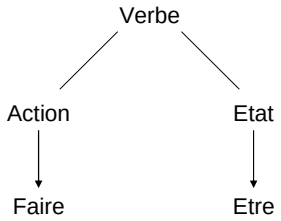


Figure 38 : Typologie du verbe d'après Tesnière (1988).

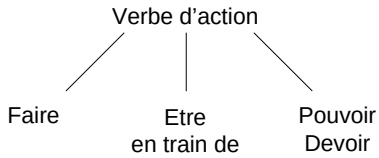


Figure 39 : Typologie des verbes d'action, d'après (Mélis, 1983).

tel-00458265, version 1 - 19 Feb 2010

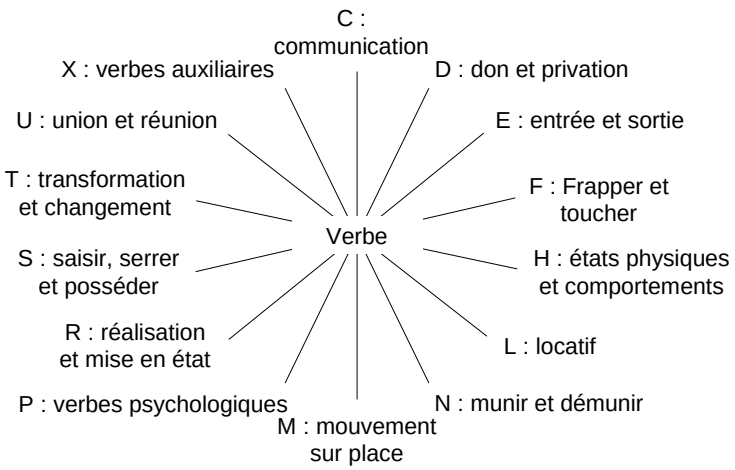


Figure 40 : Classification des verbes d'après Dubois et Dubois-Charlier (1997).

2.2 Structure de description et assemblage

Les références détaillées plus avant permettent d'établir les graphes relatifs à un certain nombre de notions. Certains de ces graphes présentent des traits communs. La notion de déplacement par exemple exprimée par Hadermann (1993) sous la forme d'un lieu initial/médian/final est homologue de celle donnée par Declerck (1997) pour la temporalité, sous la forme d'une durée, depuis un début jusqu'à une fin. Par ailleurs, dans l'étude de certains groupes fonctionnels, des notions importantes sont abordées qui préludent aux questions d'assemblage des groupes fonctionnels. Ainsi par exemple du principe d'incidence avancé par El Hasnaoui (2008) pour la Manière ou des notions de valence (Tesnière, 1988).

Dans les deux sections suivantes, on se propose de faire une étude comparative des groupes fonctionnels afin d'identifier (i) une structure de description partagée par tous les groupes fonctionnels et (ii) un formalisme d'assemblage.

2.2.1 L'hypothèse d'une structure commune aux groupes fonctionnels

Les travaux d'Hadermann (1993) relatifs à l'étude du groupe fonctionnel de localisation permettent de produire quatre graphes. Les graphes présentés en Figures 22a et 22b, donnent l'organisation du concept de lieu selon qu'il est permanent ou non. Les graphes présentés en Figures 23 et 25 traitent de la relation du lieu avec son référentiel. Cette relation peut être soit inclusive, soit de voisinage. Enfin la typologie présentée en Figure 24 traite de l'aspect sémantique inhérent à la localisation. En définitive, Hadermann (1993) considère trois classes de graphes, la première relative aux éléments qui constituent le groupe fonctionnel, la seconde relative au référentiel des éléments et la troisième à la sémantique des éléments constitutifs du groupe fonctionnel. Ces classes sont nommées respectivement Organisation, Référentiel et Sémantique. Selon cette construction, le groupe fonctionnel de localisation comporte trois classes. La relation entre ces classes s'effectue via les éléments de la classe Organisation qui se projettent sur des éléments des classes Référentiel et Sémantique (figure 41). ***L'hypothèse émise est que cette structure s'applique également aux autres groupes fonctionnels.***

tel-00458265, version 1 - 19 Feb 2010

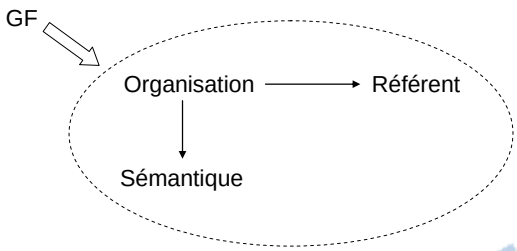


Figure 41 : Structure du groupe fonctionnel selon trois classes. La relation entre ces classes s'effectue via les éléments de la classe Organisation qui se projettent sur des éléments des classes Référentiel et Sémantique.

2.2.1.1 La classe Organisation

En ce qui concerne le temps, l'organisation est fournie par les graphes en Figures 35a et 35b avec une temporalité 'intemporelle' ou redevable d'une 'durée', cette dernière étant caractérisée par un début et une fin. Pour la Raison, l'organisation apparait en Figures 26a et 26b : la raison est soit une cause, soit une chaîne causale, caractérisée une succession de causes.

Pour la Manière, l'Actant et l'Acte, l'organisation du groupe fonctionnel n'est pas directement explicitée. Les éléments de bibliographie disponibles nous permettent toutefois de proposer une organisation.

Pour la Manière, le graphe présenté en Figure 28 différencie la façon (de faire) de l'instrument. La bibliographie n'offre pas d'information concernant ces deux notions et leur décomposition éventuelle. El Hasnaoui (2008) ne discute pas de l'instrument mais l'intègre dans son analyse comme une notion élémentaire. Pour la 'façon' (de faire), nous proposons de considérer qu'elle s'intègre dans un modus comportant une succession d'opérations : initiales, intermédiaires et finales (Figures 42a et 42b). Cette proposition permet de disposer d'une organisation des éléments constitutifs du groupe fonctionnel de Manière homologue avec celles de la localisation et de la temporalité.

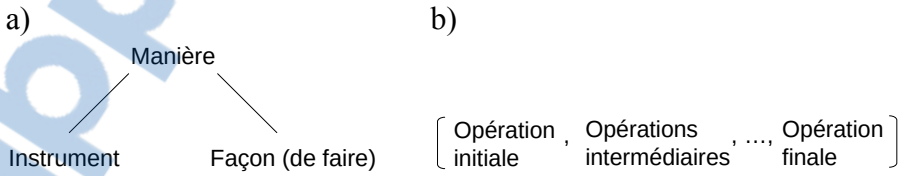


Figure 42 : Organisation du groupe fonctionnel de manière. Typologie de la constitution (a) et structure de la façon de faire (b).

Pour l'Actant, la typologie proposée par Lazard (1994) différencie l'agent de l'objet (Figure 30a). Cette typologie est suffisante dans le cas d'une construction uni-actancielle ou bi-actancielle. Dans le cas d'une phrase tri-actancielle 'je te donne une pomme', la typologie de l'agent se décompose en un agent principal 'je' et en un complément de direction 'te', la pomme étant l'objet. Cette considération permet à Lazard (1994) de traiter une phrase tri-actancielle en une bi-actancielle par accrétion du 'je' et du 'te', observé d'ailleurs dans

tel-00458265, version 1 - 19 Feb 2010

certaines langues (formes sagittales). Ces formes mettent en évidence une relation entre les agents combinés qui est externe au verbe (Figure 30b).
La question se pose maintenant des cas pour lesquels le nombre d'agents est supérieur à 2 ('je te donne une pomme via Maurice' par exemple). Ce cas n'est pas traité dans la littérature. Pour régler ce problème, je propose d'utiliser le concept de « chaine » avancé par Korzen (1986) pour la Raison. Dans l'exemple cité, la chaine s'applique à la succession 'je-Maurice-te', ce qui correspond à une relation d'ordre. Cette proposition permet de formuler une organisation du groupe fonctionnel de l'actant qui autorise les phrases uni, bi, tri et quadri-agentielles. Le groupe fonctionnel est constitué soit d'un agent « permanent » (l'auteur), soit d'une chaîne agentielle (Figure 43a). La chaîne agentielle est constituée d'un Acteur, qui initie l'acte, d'Intermédiaires, qui le perpétuent, et d'un destinataire, qui le termine.

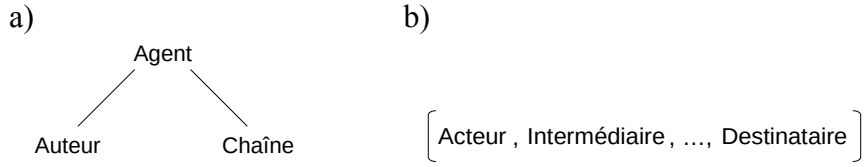


Figure 43 : Organisation du groupe fonctionnel de l'actant. Typologie de l'actant (a) et chaîne agentielle (b).

Pour l'Acte enfin, c'est le verbe qui est considéré. Sur le plan lexical, Tesnière (1988) différencie le verbe d'action du verbe d'état au moyen de 'faire' et 'être' (Figure 38). Le verbe d'état est caractérisé par une qualité ou une position. Cette distinction est approfondie par François (1989) qui utilise le critère [±dynamique] pour produire la typologie tripartite de la Figure 21. Le [+dynamique] est défini comme un évènement « perçu comme se déroulant dans le temps ». En ce sens, la temporalité est utilisée comme support de distinction entre verbe d'état et verbe d'action. De la même manière, le critère agentif participe à la typologie en tant que 'contrôle de l'action par un animé'. De la même façon, Tesnière (1988) utilise la notion de Localisation (la position) comme critère d'identification des verbes d'état. Dans notre construction, nous avons choisi de considérer les groupes fonctionnels de façon indépendante, ce qui n'est pas cohérent avec les typologies verbales proposées par Tesnière (1988) et François (1989).
Mélis (1983) considère que c'est l'emploi fait du verbe qui permet d'identifier les verbes d'action (Figure 39). Selon cette acception, l'auteur a montré qu'un emploi peut être fait du verbe 'être' en tant que verbe d'action. Etant donné que, dans le cadre du travail de thèse, l'acte est synonyme d'action, nous choisissons d'adopter la vision proposée par Mélis (1983) pour identifier les verbes d'action. Par rapport à l'acte lui-même, par contre, la littérature ne livre pas d'organisation. Faute de référence, je propose d'adopter l'organisation utilisée pour les autres groupes fonctionnels. Dans la phrase 'Je mange une pomme', 'manger ' constitue un acte permanent. Dans la phrase 'Je pèle une pomme, la découpe puis la mange' on dénombre trois actes successifs produits par le même couple (agent, objet) et qui constituent une chaine d'actions (Figure 44).

Que ce soit en référence à la littérature ou une proposition de ma part, tous les groupes fonctionnels partagent la même organisation. Celle-ci distingue l'élément permanent de la chaine considérée comme une séquence d'éléments : initial, médian (dont le nombre varie de 0 à n, et n >0), et final (Figure 45).

tel-00458265, version 1 - 19 Feb 2010

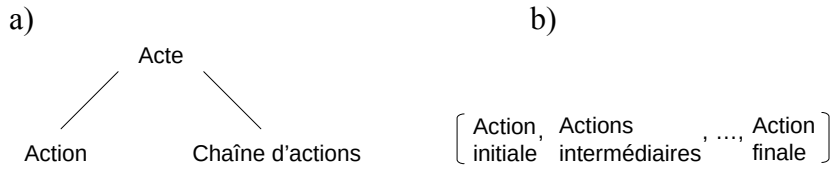


Figure 44 : Organisation du groupe fonctionnel de l'Acte. Typologie de l'acte (a) et chaîne d'actions (b).

Figure 45 : Organisation générale d'un groupe fonctionnel : typologie de la structure (a) et structure de la chaîne (b).

2.2.1.2 La classe Référentiel

La notion de référentiel est introduite par Hadermann (1993) dans le cas d'une pluralité de lieux (double localisation) et du questionnement sur le lien existant entre ces lieux. Le référé étant le lieu dont on parle, le référentiel est le lieu qui permet de situer le référé. Pour la Localisation, deux relations sont décrites, l'inclusion (Figure 23) et le voisinage (Figure 25). L'inclusion correspond à la relation contenant/contenu, et s'explicite en énonçant le contenant. Le voisinage requiert la spécification (i) des éléments voisins composant le référentiel et (ii) des liens entre le référé et les éléments du référentiel ('la maison située au bord de la mer' par exemple).

Par extension, nous considérons, pour les autres groupes fonctionnels, le référentiel comme l'ensemble des éléments de l'énoncé permettant de situer (sensu lato) le référé.

Dans la mesure où la question est de savoir si c'est le temps qui se déplace ou si c'est nous qui nous déplaçons par rapport à lui (Fillmore, 1997), la question du référentiel dans la temporalité est complexe. En effet la perception du temps est de différente nature (Mascherin, 2008) : Universel, Conventionnel et Individuel (Figure 36). Ceci amène à un référentiel lié à la nature du temps considérée dans l'énoncé. De plus, et comme dans le cas de la localisation, la coexistence de plusieurs temps peut se rencontrer dans l'énoncé ('l'été dernier, chaque après midi etc.' par exemple). Dans le cas du temps, on a donc un référentiel inclusif homologue de celui présenté dans la localisation ('l'été dernier' dans l'exemple), auquel s'ajoute une considération référentielle à la nature du temps dont il est question (le temps Conventionnel dans l'exemple). En termes de relation, nous émettons l'hypothèse que celle relative à la nature du temps est l'appartenance, et celle entre le référentiel et le référé lors de la coexistence de plusieurs temps est l'inclusion. Enfin, quel que soit le référentiel considéré, le positionnement relatif du référé par rapport au référentiel s'effectue en combinant les notions de point, de durée et d'itération (Figure 34).

Pour l'Actant, la typologie du nombre présentée en Figure 32 caractérise la personne d'après le nombre d'individus représentés et les liens que ceux-ci entretiennent. Quand un individu produit un énoncé, il représente une entité singulière, duelle, trielle ou plurielle. Dans ce cas,

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

tel-00458265, version 1 - 19 Feb 2010

nous considérons l'individu comme référé et l'entité comme référentielle. La relation entre référentiel et référé est par définition inclusive. Toutefois, selon Jacquesson (2008), les notions de duel et de triel sous-tendent l'existence d'un lien de composition entre individus. Dans ce cas, la relation du référé au référentiel est subordonnée au lien qu'entretiennent les individus, ce qui nécessite un complément d'information pour situer le référé.

En conclusion, pour la Localisation, la Temporalité et l'Actant, le référentiel se formule au moyen des différentes structures que sont l'inclusion et le voisinage avec ou sans lien entre individus. En ce qui concerne la Raison, la Manière et l'Acte, la bibliographie ne fournit pas d'information sur le référentiel. ***Pour la classe Référentiel, il ne m'est donc pas possible de proposer une structure commune de description.***

2.2.2.3 La classe Sémantique

La sémantique est introduite par Hadermann (1993) au moyen de la notion de valeur locative intrinsèque ou extrinsèque. Cette approche conduit à distinguer le lieu dont la sémantique est propre à lui-même, du lieu dont la sémantique découle des autres lieux (Figure 24).
Pour l'Actant, Jacquesson (2008) distingue diverses modalités de désignation d'une personne par une autre : politesse, familiarité, honorifique, tabou (Figure 31). Ces modalités correspondent à un code social extrinsèque à la personne. Nous proposons donc d'adopter l'opposition intrinsèque/extrinsèque proposée par Hadermann pour formuler la sémantique de l'actant (Figure 46).
Pour la Temporalité, Grenier (2001) différencie le temps physique du temps historique (Figure 37). Dans le temps historique « le contenu de chaque instant dépend du contenu de chacun des instants qui l'ont précédé ». Cette définition du temps historique renvoie au critère de définition du lien « extrinsèque » d'Hadermann (1993). Le temps physique est un système de mesure du temps universel auto-signifiant, qui renvoie au critère de lieu « intrinsèque » (Figure 47).
Pour la Manière, les critères de valeur de vérité (tacite/explicite) et d'entendement (objectif/subjectif) sont combinés pour qualifier différentes formes de savoir (Figure 29). Pour reprendre la définition de Charlot (1997), le Savoir est une donnée tacite, dans la mesure où elle est issue d'une expérience personnelle et objective du fait d'une édification méthodologique. La Connaissance est une donnée tacite et subjective car elle est issue de l'expérience personnelle du sujet. Dans la mesure où la valeur de vérité du savoir et de la Connaissance dépend d'un critère tacite, l'explicitation de la manière n'est pas propre à elle-même, ce qui renvoie au critère « extrinsèque ». Par opposition, l'Information explicite est « intrinsèque ». Dans le domaine des sciences, l'hypothèse est un élément fondamental des constructions intellectuelles. Nous proposons d'enrichir la typologie proposée en Figure 29 en introduisant « l'hypothèse » que nous considérons comme une forme de savoir explicite et subjective (Figure 48), ce qui la classe en « intrinsèque ».
Pour la Raison, la bibliographie en linguistique ne fournit pas d'information redevable de la classe sémantique. En philosophie toutefois, Spinoza (1677) utilise l'opposition intrinsèque extrinsèque avec deux modalités pour l'extrinsèque : intentionnel et par accident. Ce pourrait être une piste pour construire la typologique de la sémantique de la raison.
Concernant l'acte enfin, Sénéchal et al. (2007) considère que la classification des verbes proposée par Dubois et Dubois-Charlier (1997) ne tient pas suffisamment compte de la sémantique. L'aspect sémantique de l'acte mériterait donc d'être approfondi.

tel-00458265, version 1 - 19 Feb 2010

En conclusion, la sémantique de la Localisation, de l'Actant et de la Temporalité, peut être explicitée au travers des critères « Intrinsèque » et « Extrinsèque » introduits par Hadermann (1993). Cette dichotomie s'applique également à la Manière. Toutefois, dans la mesure où quatre types de Manière sont distingués, un second niveau de discrimination est introduit sur l'entendement (objectif/subjectif). Pour la Raison, la piste évoquée fait référence à une discrimination entre « intrinsèque » et « extrinsèque ». Enfin, pour l'Acte, une sémantique est requise mais non disponible à ce jour. Hormis « l'Acte » dont la structuration sémantique n'est pas aboutie, la sémantique des autres groupes fonctionnels est définie au moyen d'une typologie. La structure générale de la sémantique des groupes fonctionnels est un graphe.

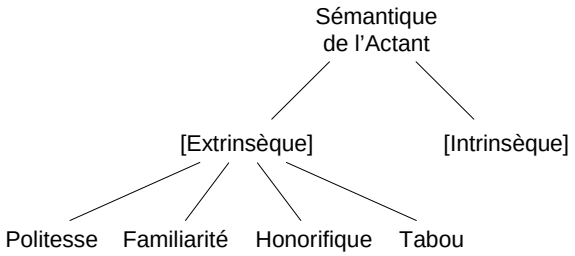


Figure 46 : Sémantique de l'Actant.

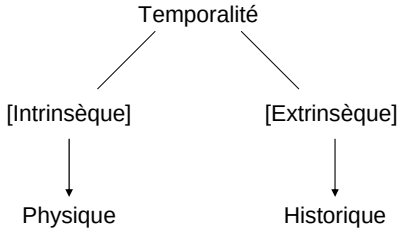


Figure 47 : Sémantique de la Temporalité.

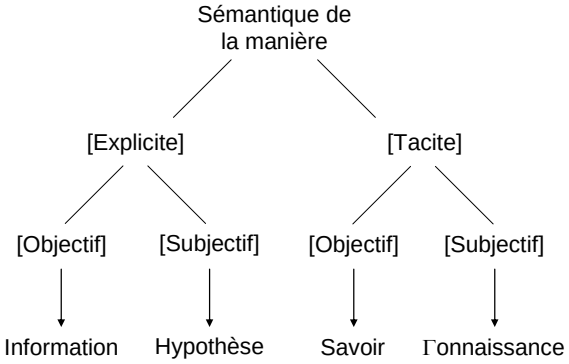


Figure 48 : Sémantique de la manière.

2.2.2 La structure de description

Différentes notions de grammaires ont été présentées qui préludent à l'assemblage des groupes fonctionnels. Parmi celles-ci, les notions d'actance et de valence qui sont omniprésentes pour l'Acte et l'Actant. La valence caractérise le nombre d'actants adjoints à

tel-00458265, version 1 - 19 Feb 2010

un verbe, et prend pour valeur zéro, un deux ou trois (Tesnière, 1988). Cette notion a été revisitée du point de vue de l'Actant par Lazard (1994), qui propose de formaliser les constructions comportant de zéro à trois-actants sous la forme de constructions uni ou bi-actancielles. Dans le cas d'une bi-actancielle, les deux actants sont l'agent et l'objet.

La seconde notion est le principe d'incidence, introduit par El Hasnaoui (2008) à l'occasion de l'étude de la Manière. Ce principe appréhende le substantif comme l'élément support de la phrase, complété par les autres éléments (les apports). Dans le cas d'une construction uni-actancielle, ce principe définit (i) l'actant comme le support, (ii) le verbe comme incident externe au verbe au premier degré, et (iii) les circonstants comme incident externe au verbe au premier degré (Figure 49a). Par transitivité, les circonstants sont incidents externes à l'actant au second degré. Lorsqu'il y a deux actants, un agent et un objet, le principe d'incidence stipule que l'objet est incident au verbe, et donc à l'Acte (Figure 49b). L'objet est par conséquent incident externe de second degré à l'actant. Cette construction correspond à la structure neutre de Lazard (1994).
Le principe d'incidence est un outil d'analyse de la syntaxe, qui permet entre autres d'expliquer la conjugaison du verbe. Ilinski (2003) et El Hasnaoui (2008) montrent que, si ce principe est simple en apparence, il requiert la prise en compte simultanée des points de vue syntaxique et sémantique. Par conséquent, les flèches de la Figure 48 reliant les différents groupes fonctionnels et répondant à une construction syntaxique, véhiculent également une sémantique.

Pour ce qui concerne la construction uni-actancielle, les flèches indiquent que les circonstances ont une influence sur l'Acte, qui à son tour influe sur l'Actant. Pour la construction bi-actancielle, la flèche entre l'Objet et l'Acte est identique à celles des circonstances : Localisation, Temporalité, Manière et Raison. Autrement dit, vis-à-vis de l'action produite, l'objet n'est pas considéré par les circonstances, mais sa contribution à la production de l'Acte par l'Agent est de même degré d'incidence que les circonstances.

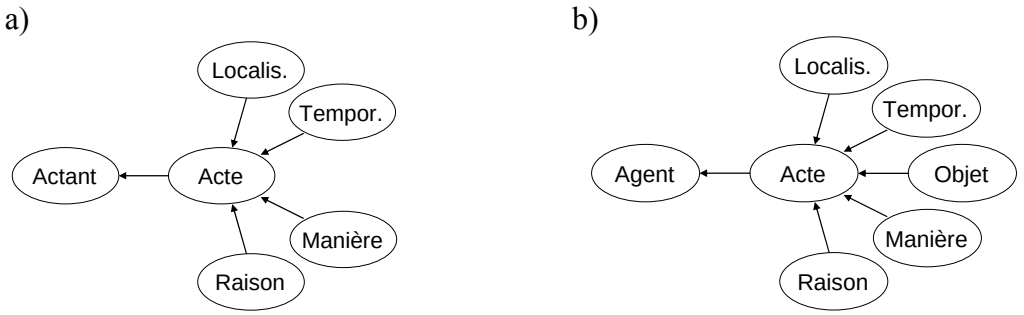


Figure 49 : Structure de Description, établie au moyen du principe d'incidence, pour une construction uni-actancielle (a) et bi-actancielle (b).

Dans l'étude du circonstant de Manière, El Hasnaoui (2008) montre que le groupe fonctionnel de Manière peut également être incident au sujet de la phrase (Figure 50a). Dans ce cas, le groupe fonctionnel de Manière comporte un acte conduit par l'Actant de façon simultanée à l'acte exprimé par le verbe (Figure 50b). C'est le cas par exemple dans la phrase 'Jean marche en sifflotant'. La conduite de deux actes par un même actant correspond à la production de deux graphes de la Figure 49. Chaque graphe correspond à une action, et l'actant des graphes (ou l'agent suivant le cas) est un même individu.

tel-00458265, version 1 - 19 Feb 2010

La question de la simultanéité de plusieurs actes pose la question du lien pouvant exister entre leurs groupes fonctionnels respectifs de Localisation, Temporalité, Manière, Raison et Objet. La simultanéité fait référence à un parallélisme temporel. Bien que de « même durée », deux actes peuvent toutefois être produits dans des référentiels temporels différents. Pour la Localisation, le référentiel et/ou le lieu peuvent être différents. Dans l'exemple, 'Jean marche en soufflant', les deux lieux sont distincts : 'le chemin' et 'en l'air' par exemple. L'interrelation entre les deux actes est possible. Dans l'exemple 'Jean marche en jonglant avec un ballon', une interaction entre les deux actes s'effectue au niveau des pieds, donc de la manière 'de marcher' et 'de jongler'. L'interaction peut être plus complexe dans le cas où les groupes fonctionnels sont combinés, par exemple l'espace de l'un est la temporalité de l'autre. C'est le cas par exemple en biologie de la durée de vie larvaire raccourcie par la consommation d'un champignon (Mondy et Corio-Costet, 2000).

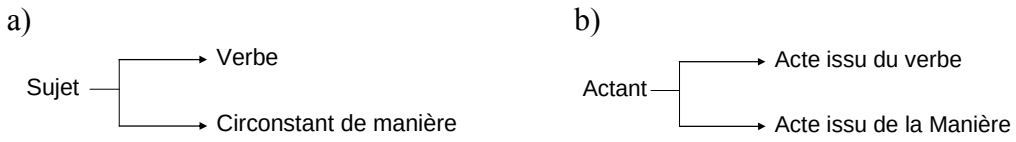


Figure 50 : Simultanéité de conduite d'un acte exprimé par le circonstant de manière du point de vue syntaxique (a) et exprimé au moyen des groupes fonctionnels (b).

2.3 Description du contexte

Deux approches de la notion de contexte ont été présentées dans la bibliographie en Linguistique. Sarfati (2002) distingue le contexte restreint, réunissant les éléments tangibles (circonstances et interaction) du contexte étendu, réunissant les éléments à priori intangibles (situation et présupposition), voir Figure 51. Dans cet entendement, la description du contexte réunit les informations qui environnent le discours. Dans la seconde approche Stalnaker (1998), le contexte est appréhendé comme un univers qui évolue de façon concomitante avec le discours. La description du contexte comporte alors des informations supposées partagées par tous les protagonistes : l'objet de l'action, les faits qui apparaissent au cours de son déroulement, les relations causales entre situations locales apparues durant le déroulement de l'action, etc.

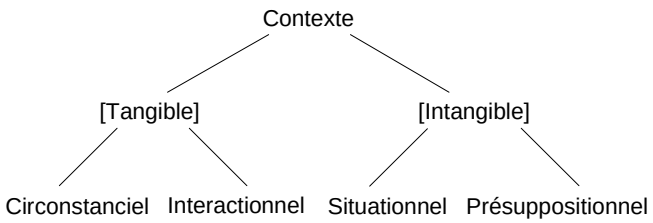


Figure 51 : Typologie du contexte d'après (Sarfati, 2002).

Le contexte circonstanciel livre l'identité des interlocuteurs, l'environnement physique, le lieu et le temps où la locution est produite. Le contexte interactionnel renseigne sur la séquence d'enchaînement des locutions dans une séquence inter-discursive. Hormis l'environnement physique, les traits de ces deux contextes correspondent au contenu de la classe Organisation constitutive des groupes fonctionnels. L'environnement physique, par contre, relève de la classe Référentiel.

tel-00458265, version 1 - 19 Feb 2010

Le contexte situationnel fournit les éléments relatifs à l'environnement culturel de l'action définissant les critères locaux de validité de la locution et le contexte présuppositionnel informe sur les croyances et valeurs communes aux locuteurs. Les traits de ces deux contextes correspondent au contenu de la classe Sémantique constitutive des groupes fonctionnels.

L'approche de Stalnaker (1998) demande qu'à l'objet de l'action soient associés les faits qui apparaissent au cours de son déroulement, les relations causales entre situations locales apparues durant le déroulement de l'action, etc. Ces faits et relations correspondent peu ou prou à la définition de l'énonciation avancée par Benveniste (1966). Dans la mesure où cette définition d'un contexte interactif n'est pas accompagnée d'une structure dédiée particulière, nous émettons l'hypothèse que le contexte selon Stalnaker (1998) se décrit comme le contexte selon Sarfati (2002), mais requiert l'adjonction d'une énonciation à l'énoncé. Nous admettons que l'énonciation représente un énoncé particulier auquel peut donc s'appliquer les trois classes constitutives des groupes fonctionnels d'Organisation, de Référentiel et de Sémantique. En termes de relation, nous considérons que l'énoncé et l'énonciation représentent deux actions simultanées selon la structure décrite en Figure 50b.
La distinction entre les approches de Sarfati (2002) et de Stalnaker (1998) réside dans la nature du lien existant entre contexte et énoncé, respectivement indépendante ou inter-reliée. Selon l'approche considérée, il convient ou non d'établir une interrelation entre le discours et le contexte. Les différentes formes d'interrelation sont énoncées en Figure 02. L'objectif de l'étude étant de proposer un modèle de description d'un terme, l'évolution concomitante de l'énoncé et du contexte ne correspond pas à notre cas d'étude. L'interrelation entre le contexte et l'énoncé, au sens défini pas Stalnaker (1998), ne sera par conséquent pas considérée dans la suite de ce document.

3 Proposition d'un modèle formel d'accès aux connaissances pragmatiques

Au travers de la linguistique, nous avons pu identifier une méthode de description basée sur le langage naturel et exprimée par un ensemble de graphes. La méthode utilise sept groupes fonctionnels indépendants : l'Agent et l'Objet, qui représentent les Actants, l'Acte, la Localisation, la Temporalité, la Manière et la Raison. Ces groupes fonctionnels sont constitués des trois classes d'Organisation, de Référentiel et de Sémantique. L'utilisation du principe d'incidence a permis de situer relativement les groupes fonctionnels et de proposer une structure de description. L'accès à une connaissance pragmatique est obtenu par adjonction d'un contexte, la description du contexte et de l'énoncé étant décrits au moyen d'une même structure de description.

La question qui se pose à présent est celle de la manipulation des structures de description par l'outil informatique. Par définition, l'informatique est la science du traitement automatique de l'information. L'utilisation des structures de description dans les plateformes logicielles ne peut donc s'envisager qu'à condition que celle-ci soit formalisée.

tel-00458265, version 1 - 19 Feb 2010

3.1 Formalisation syntaxique d'un terme

Un groupe fonctionnel est constitué des classes Organisation, Référentiel et Sémantique (Figure 41) et une structure commune à tous les groupes fonctionnels a été identifiée pour les classes Organisation et Sémantique.
Pour la classe Organisation, le groupe fonctionnel est constitué soit d'une donnée unique, soit d'une succession de données (Figure 45). Du point de vue mathématique, ces deux structures peuvent être représentées par un graphe simple orienté **GO** = (VO, EO), où VO est l'ensemble des sommets non ordonnés, et EO l'ensemble des arcs. Dans le cas de la donnée unique, l'ensemble EO est vide, et dans le cas contraire, EO est une chaîne eulérienne (chaîne empruntant une fois et une fois seulement chaque arête de EO). Pour la classe Sémantique, les structures proposées sont des typologies discriminant des concepts à priori élémentaires (Figure 24 pour la Localisation, Figure 46 pour l'Actant, Figure 47 pour la Temporalité, Figure 48 pour la Manière). En termes mathématiques, la typologie peut être représentée par un graphe arborescent **GS** = (VS, ES), où VS est l'ensemble des sommets, et ES l'ensemble des arcs. Pour la classe Référentiel, dans la mesure où la bibliographie linguistique n'a pas permis d'identifier une structure commune, il n'est pas possible d'en proposer une représentation mathématique. Toutefois, nous supposerons l'existence d'un graphe associé que nous noterons **GR** = (VR, ER), où VR est l'ensemble des sommets, et ER l'ensemble des arcs. Le graphe associé au groupe fonctionnel est **GGF** = (VF, EF), où VF est constitué des objets GO, GR et GS, et EF des arcs qui relient GO, GR et GS. Dans le cas présent, l'ensemble EF est vide. La relation établie entre classes s'effectue au niveau des éléments. Chaque élément de la classe Organisation, est projeté dans les classes Sémantique et Référentiel. Un exemple de projection est présenté en figure 52. Dans cet exemple, la classe Organisation est constituée de 3 éléments qui se succèdent. Ces éléments sont représentés par des points et la relation de succession par les arcs orientés. A chaque élément de la classe Organisation correspond un point projeté dans les classes Sémantique ou Référentiel. Ces points projetés participent aux graphes particuliers des classes Sémantique et Référentiel, respectivement structurés selon une relation de subsomption et de composition dans l'exemple de la figure 52.

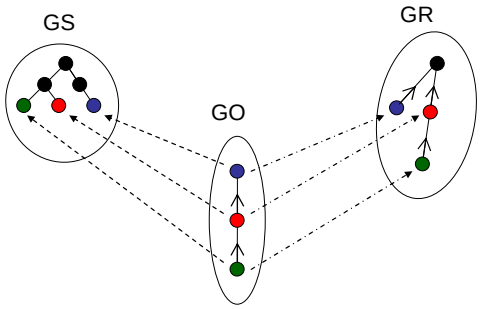


Figure 52 : Exemple de relations entre éléments des classes Organisation (GO), Sémantique (GS) et Référentiel (GR).

tel-00458265, version 1 - 19 Feb 2010

Deux formes de structure de description sont possibles selon le nombre d'actants (Figure 49). Dans le cas d'un seul actant, la structure de description est constituée de six groupes fonctionnels (Actant, Acte, Localisation, Temporalité, Manière, Raison). Dans le cas de deux actants (Agent et Objet), la structure de description comporte sept groupes fonctionnels (Agent, Acte, Objet, Localisation, Temporalité, Manière, Raison). Quelque soit le nombre d'actants, les groupes fonctionnels sont organisés selon le principe d'incidence, qui établit une relation orientée par paire de groupes fonctionnels. En termes mathématiques, la structure de description peut être représenté sous la forme d'un graphe orienté **GSD** = (VSD, ESD), où VSD est l'ensemble des sommets, et ESD l'ensemble des arcs. Les éléments de VSD sont des graphes GGF correspondant aux différents groupes fonctionnels (GF_{Ag}, GF_{Ac}, GF_{Lo}, GF_{Te}, GF_{Ma}, GF_{Ra}), avec ou non la présence de GF_{Ob}, selon le nombre d'actants. L'indice accolé au GF correspond aux deux premières lettres du groupe fonctionnel représenté, 'Lo' pour Localisation, par exemple. De façon à différencier l'Actant de l'Acte dans les indices, l'Actant a été nommé GF_{Ag}, 'Ag' correspondant à Agent. Les arcs de ESD représentent les relations entre les groupes fonctionnels, définies selon le principe d'incidence (Figure 49).

D'après la linguistique pragmatique (chapitre B, paragraphe 1.1), un terme est représenté par le couple (contexte, énoncé). Dans la mesure où le contexte et l'énoncé sont décrits en utilisant la structure de description, le terme T s'exprime en mathématique sous la forme du graphe **GT** = (VT, ET), où VT est l'ensemble des GSD, à savoir SD_E et SD_C correspondant respectivement au GSD associé à l'énoncé du terme et à celui associé à son contexte d'élaboration. ET est l'ensemble des arcs entre SD_E et SD_C, i.e. l'énoncé et le contexte.

La relation entre les éléments de l'énoncé et ceux du contexte, formalisée par ET, n'est pas explicitée dans la bibliographie. Bien que le contexte soit défini comme « comportant les éléments descriptifs de la situation de locution nécessaires à la juste compréhension de ce qui est dit » (Armengaud, 2007), il ne permet pas d'identifier les ensembles où les liens sont établis. Dans la définition, l'expression « la juste compréhension de ce qui est dit » pourrait évoquer la relation entre les graphes relatifs à la sémantique, voire au référentiel (GS et GR). En l'absence de précisions, nous proposons de considérer ET comme un ensemble vide $\{\varnothing\}$.

Le regroupement des graphes élémentaires permet de proposer un modèle de formalisation de la description d'un terme. Ce modèle correspond à un graphe multi-niveaux (Figure 53). Au niveau du terme, ce modèle comporte les structures de description du contexte SD_C et de l'énoncé SD_E. Chaque structure de description est décrite en termes de groupes fonctionnels (GF_{AG}, GF_{AC}, etc.), ces derniers étant constitués des classes Organisation, Référentiel et Sémantique.

tel-00458265, version 1 - 19 Feb 2010

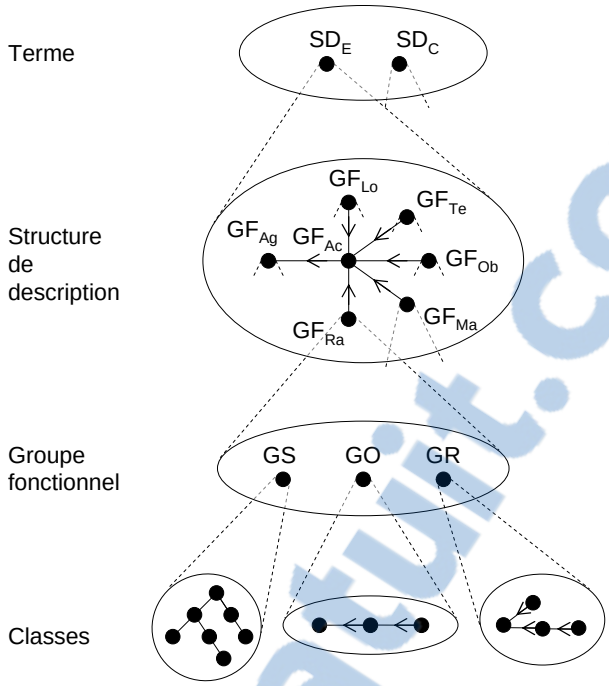


Figure 53 : Description multi-niveaux d'un terme

3.2 Dictionnaires

Les classes Organisation et Référentiel ont pour objet de transmettre une donnée, tandis que la classe Sémantique a pour objet de faire comprendre la donnée transmise. L'objet de la description du terme étant de résoudre les problèmes de sémantique, la constitution de dictionnaires permet de rassembler les vocables situés dans les différentes classes Sémantique et d'identifier les différentes utilisations d'un même vocable.

Les graphes des classes Sémantique sont des arborescences de termes reliés au moyen de la relation de subsomption. Par construction, la sémantique d'un vocable est définie selon l'arborescence des vocables situés en amont. Différents types de regroupements peuvent être produits selon l'objectif attendu. La réunion de tous les vocables, indépendamment de leur arborescence, correspond à la production d'un lexique. La prise en compte des relations entre vocables donne lieu au vocabulaire. Un premier vocabulaire, appelé 'vocabulaire somme', peut être produit en effectuant la somme des graphes des GS. Un terme pouvant être défini selon des arborescences différentes, le graphe résultant fournit les différentes définitions sémantiques attribuées à chacun.
Un autre vocabulaire, appelé 'vocabulaire somme disjointe', peut résulter de la somme disjointe des GS, où la nature du groupe fonctionnel (Localisation, temporalité, etc.) est associée à chaque GS. Les GS de l'énoncé et du contexte étant sommés, l'ensemble résultant montre les différentes définitions sémantiques des vocables par nature de groupe fonctionnel. Ce dernier vocabulaire comporte les variations d'usage des vocables, et donc des variantes sémantiques induites par l'usage (polysémie et polyphonie).

La syntaxe des vocables utilisés dans l'arborescence étant définie par la structure syntaxique (accusative, ergative, neutre, mixte, disjointe ; chap. B, § 1.3.4), le mécanisme nécessaire à la construction automatique des dictionnaires est dépendant du choix de la structure syntaxique.

tel-00458265, version 1 - 19 Feb 2010

L'adoption de la structure neutre offre l'avantage de contourner la question de la syntaxe des vocables. Dans la mesure où les vocables ne nécessitent pas de traitement syntaxique particulier, la construction des trois dictionnaires, à savoir le lexique, le vocabulaire somme et le vocabulaire somme disjointe, est alors immédiate.

3.3 Mise en œuvre du modèle formel

Par construction, le modèle formel proposé s'approche de la famille des réseaux sémantiques. Après avoir présenté deux variantes de réseaux sémantiques (Kayser, 1997), nous nous proposons d'utiliser les graphes conceptuels comme support d'implémentation. L'enjeu est de bénéficier d'un support théorique préexistant.

3.3.1 Les réseaux sémantiques

Un réseau sémantique est un langage de « représentation de connaissances » (Kayser, 1997 ; Schreiber, 2008). Ce langage dispose d'un formalisme permettant la constitution d'un graphe qui associe des connaissances, et propose un mécanisme de raisonnement basé sur l'inférence. Le graphe de connaissances est un graphe orienté et étiqueté, dans lequel un nœud représente un concept, et un arc reliant des nœuds représente une relation sémantique entre les concepts. Trois arcs sont principalement utilisés : la subsomption 'Is A', qui permet la construction de structures hiérarchiques de classes de concepts et de relations, la spécialisation 'Is a kind of', qui a pour vocation de relier la classe et ses instances, et la composition 'Has A', qui permet l'inclusion de concepts et de relations.

Une première variante aux Réseaux Sémantiques, appelée Réseaux Sémantiques Partitionnés (Hendrix, 1979), dispose d'une notion en sus de celles de concept et de relation : la partition. Selon Kayser (1997), la partition permet de produire des sous-ensembles de concepts et de relations sémantiques non nécessairement connectés. Ces partitions sont manipulées au même titre que les concepts. Elles permettent d'associer des contraintes particulières ou des propriétés, et d'appliquer, à tous les éléments de la partition, des opérations telles que la disjonction et la négation. Quatre types de lien sont différentiés : 'e', 's', 'de' et 'ds'. 'e' représente l'appartenance d'une entité à une classe. 's' représente l'inclusion d'une sous-classe de concept dans une classe de concept. 'de' représente l'appartenance d'un élément à une classe mais l'élément est distinct de tous les autres éléments de la classe liés par un lien 'de'. 'ds' présente la même définition du lien 'de' mais appliqué aux sous-classes. A titre d'illustration, le graphe de connaissances, présenté en figure 54, représente l'affirmation « tout mammifère a un cœur », d'après Kayer (1997). Ce graphe correspond à la formule logique :

$$(\forall x)\ (\text{mammifère}(x) \supset (\text{cœur}(y) \wedge \text{possède}(x,y))$$

où le symbole $\supset$ représente l'implication.

tel-00458265, version 1 - 19 Feb 2010

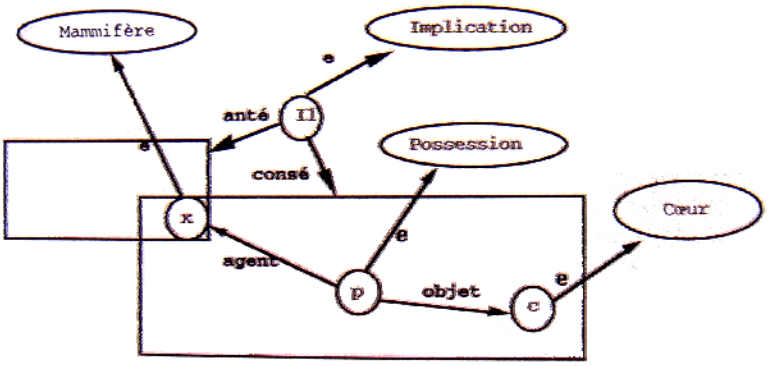


Figure 54 : Exemple de graphe de connaissances selon la méthode des réseaux partitionnés, d'après Kayser (1997). Dans cette figure, les rectangles, les cercles et les flèches représentent respectivement les partitions, les nœuds du graphe, et les relations orientées entre nœuds et partitions. La relation notée 'e' correspond à l'appartenance de l'entité située à l'origine de l'arc à celle située à l'extrémité. Les relations 'anté' et 'consé' représentent respectivement les notions d'antériorité et de conséquence. Celles-ci permettent la formulation de la relation d'implication I1.

Une seconde variante, appelée Graphes Conceptuels (Sowa, 1984), présuppose de l'existence d'une organisation relative des concepts à assembler. Selon Chein et Mugnier (2008), un graphe conceptuel comporte (i) un vocabulaire, (ii) une collection de graphes (de connaissances) reliant les concepts au moyen de relations sémantiques, et (iii) une collection de relations, dites de coréférence, qui permettent de représenter un même objet au moyen de plusieurs concepts. Le vocabulaire différencie explicitement la classe de l'instance. Le vocabulaire est constitué (i) de l'ensemble des relations sémantiques organisées au moyen de la relation de subsomption 'Is a', (ii) de l'ensemble des classes des concepts organisés au moyen de la relation de subsomption 'Is a', (iii) de l'ensemble des classes de concepts organisés au moyen de la relation de composition 'Has a', et (iv) de l'ensemble des marqueurs. Cet ensemble comporte le marqueur générique représentant l'instance quelconque d'une classe, et les marqueurs de classes correspondant à l'instance d'une classe de concept au moyen de la relation d'instanciation 'Is a Kind of'. La sémantique d'un nœud dans un graphe de subsomption, à savoir (i) et (ii), s'obtient en concaténant la sémantique des nœuds situés en amont du nœud considéré dans le graphe.
L'établissement du raisonnement s'effectue par composition du vocabulaire et des graphes de connaissances. Les relations de subsomption spécifiées dans le vocabulaire correspondent à des règles de généralisation sémantique. Elles permettent de mettre en relation les graphes. La figure 55 montre un exemple de graphe conceptuel, issu de Kayser (1997). Le réseau situé au centre de la figure 55b correspond à la formule logique suivante :

$$f(H) = sol(s) \wedge (\exists\, x,y,z,t)\ (polygone(x) \wedge objet\text{-}géométrique(y) \wedge carré(z) \wedge rectangle(t) \wedge sur(x,z) \wedge sur(y,t) \wedge sur(t,s))$$

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

tel-00458265, version 1 - 19 Feb 2010

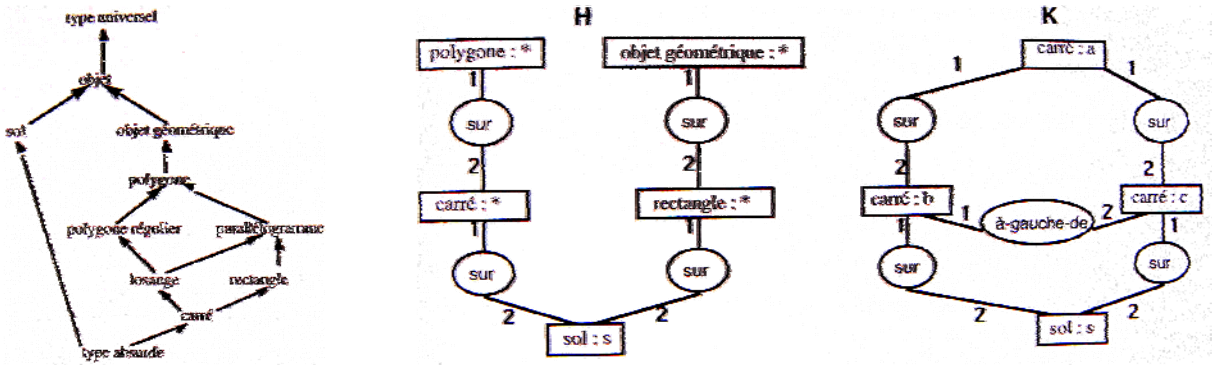


Figure 55 : Exemple de graphe conceptuel (Kayser, 1997) constitué d'un vocabulaire à gauche et de deux graphes de connaissances H et K situés, respectivement, au centre et à droite de la figure. Pour le vocabulaire, la flèche entre deux concepts représente la relation de subsomption 'Is a'. Pour les graphes, les concepts sont représentés par des rectangles. A l'intérieur du rectangle, l'étoile représente le marqueur générique. Les symboles 's' et 'a' représentent respectivement les instances des concepts 'sol' et 'carré'. Les relations sont représentées par des cercles.

La description d'une connaissance s'effectue de façon différente en fonction de la variante de réseau sémantique. Pour les réseaux partitionnés, les nœuds du graphe de connaissances sont des partitions, des concepts et des instances des concepts. Pour les graphes conceptuels, deux types de graphes sont constitués : le graphe de vocabulaire, composé de l'ensemble des concepts reliés au moyen de la relation de subsomption, et les graphes de connaissances constitués soit des concepts instanciés par le marqueur générique, soit par un marqueur instance de classe. La notion de relation entre éléments au sein d'un graphe de connaissances est appréhendée de façon différente selon la variante considérée de réseau sémantique. Pour les réseaux partitionnés, une relation est une flèche, à savoir un lien binaire orienté, alors que pour un graphe conceptuel, il s'agit d'un nœud particulier disposant d'une arité définie par l'utilisateur.
Les trois arcs, décrits dans l'introduction de ce paragraphe, sont ainsi utilisés de façon différente. Pour les réseaux partitionnés, la relation de composition 'Has a' est explicitée au moyen d'une partition, tandis que pour les graphes conceptuels, elle est formulée au moyen d'un concept de composition. De plus, pour les graphes conceptuels, l'établissement d'un lien de coréférence revient à considérer que les concepts coréférencés représentent le même objet. Il s'agit là d'une forme d'inclusion particulière auprès d'un objet tiers. Pour les réseaux partitionnés, la relation de spécialisation 'Is a kind of' correspond à une relation ordinaire au sein du graphe de connaissances, alors que pour un graphe conceptuel, cette relation est définie dans le vocabulaire. Enfin, la relation de subsomption 'Is a' sert à la constitution du vocabulaire d'un graphe conceptuel, alors que dans les réseaux partitionnés elle est utilisée au même titre qu'une autre relation dans le graphe de connaissances.

3.3.2 Représentation du modèle formel au moyen des graphes conceptuels

Le Modèle Formel décrit au § 3.1 du chapitre B présente une structure multi-niveaux emboités (figure 53). Chaque niveau (i) traite d'un aspect particulier de la description et (ii) utilise un opérateur spécifique de mise en relation des éléments. Par exemple, au niveau K, il est constitué de deux Structures de Description non reliées, l'énoncé et le contexte

tel-00458265, version 1 - 19 Feb 2010

d'élaboration. Au niveau de la structure de description, les groupes fonctionnels (Acte, Actant, etc.) sont reliés d'après leur nature et selon le principe d'incidence. Selon Kayser (1997), « le système de représentation proposé par Sowa allie une certaine souplesse, par laquelle il se rapproche de l'efficacité descriptive du langage naturel, et un bon degré de rigueur, grâce auquel les procédures inférentielles peuvent être justifiées ». Nous nous appuierons sur le jugement de Kayser (1997) et utiliserons donc la méthode des graphes conceptuels comme support d'implémentation du Modèle Formel.

Représenter un graphe au moyen d'un Graphe Conceptuel consiste à identifier les concepts et les relations pour chaque graphe constitutif du Modèle Formel (figure 53). Dans la mesure où la construction est multi-niveaux, nous procéderons dans un premier temps à la représentation du ou des graphes de chaque niveau, puis dans un second à la combinaison des graphes entre niveaux.
Pour la classe Organisation, les concepts du graphe de connaissances sont les éléments de VO et les relations entre les concepts du graphe de connaissances correspondent aux éléments de EO. La relation de la classe Organisation exprime l'enchaînement d'éléments. La figure 56 montre un exemple de séquence constituée de concepts instanciés du marqueur générique. Dans cet exemple, la relation est appelée 'Précède', mais peut porter d'autres noms.
Pour la classe Sémantique, les concepts du graphe de connaissances sont les éléments de VS et les relations entre les concepts ceux de ES. La relation reliant les concepts de cette classe est la subsomption. Cette classe permet de constituer le vocabulaire (§ 3.2 du chapitre B). Dans les Graphes Conceptuels, la relation de subsomption est également utilisée pour la constitution du vocabulaire. Il s'en suit que la classe Sémantique du Modèle Formel correspond au vocabulaire des graphes conceptuels. Pour la classe Référentiel, les concepts du graphe de connaissances sont les éléments de VR et les relations entre les concepts du graphe de connaissances correspondent aux éléments de ER. Différentes relations peuvent être utilisées dans cette classe. Par exemple, dans l'étude de la Temporalité, deux relations ont été identifiées : l'inclusion (figure 23) et le voisinage (figures 25).

Au niveau du groupe fonctionnel, les concepts du graphe de connaissances sont les éléments de VF, et les relations ceux de EF. La relation entre ces classes s'effectuant par la projection de leurs éléments respectifs (figure 41), il n'y a donc pas de relation entre les classes. Par conséquent, EF est vide.

Au niveau de structure de description (figure 53), les groupes fonctionnels sont reliés au moyen de la relation d'incidence. Le groupe fonctionnel d'Acte correspond à un verbe. Selon le mode d'appréhension du verbe dans les graphes conceptuels, deux formulations de la structure de description sont possibles. Dans le premier cas, les Graphes Conceptuels considèrent qu'un verbe est obligatoirement représenté par une relation (Figure 57a). Dans ce cas, le groupe fonctionnel d'Acte est la relation d'alliance des groupes fonctionnels. L'arité de la relation prend alors pour valeur six ou sept selon que le groupe fonctionnel d'Objet est présent ou non (figure 49). De façon à différencier les groupes fonctionnels au sein du graphe, un ordre de liaison des concepts à la relation doit être prédéfini. Dans l'exemple de la figure 57a, l'ordre considéré est l'Actant, puis la Temporalité, la Localisation, la Manière, la Raison et enfin l'Objet. Dans le second cas, les Graphes Conceptuels considèrent qu'un verbe peut être représenté par un concept (figure 57b). Ce faisant, un concept, même s'il est instancié, ne représente pas alors nécessairement un substantif. Dans ce cas, le groupe fonctionnel d'Acte est représenté par un concept au même titre que les autres groupes fonctionnels. Les concepts du graphe de connaissances sont les éléments de VSD, et les relations les éléments de ESD. La sémantique des relations contenues dans ESD correspond à celle des groupes fonctionnels.

tel-00458265, version 1 - 19 Feb 2010

En figure 57b, les relations ont été notées ‘Que’, ‘Quoi’, ‘Ou’, ‘Quand’, ‘Comment’ et ‘Pourquoi’ pour relier respectivement l’Actant et l’Acte, l’Acte et l’Objet, l’Acte et la Localisation, l’Acte et la Temporalité, l’Acte et la Manière, et l’Acte et la Raison.
Au niveau K enfin, les nœuds du graphe de connaissances sont les éléments de VK, et les relations ceux de EK. Dans la mesure où EK est un ensemble vide {∅} (§3.2 chapitre B), le graphe de connaissances est constitué de deux concepts instanciés non reliés.

Dans le Modèle Formel, l’imbrication des niveaux correspond à la relation de composition. La représentation du Modèle Formel consiste à relier les différents graphes de connaissances au moyen de la relation de composition ‘Has A’, proposée par les graphes conceptuels (§ 3.3.1). Il s’en suit que les classes, les groupes fonctionnels, la structure de description et le terme sont représentés par un concept de composition. La figure 58 montre un exemple de description d’un groupe fonctionnel.

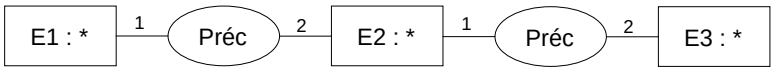


Figure 56 : Succession d’éléments décrite au moyen des graphes conceptuels.

a) b)

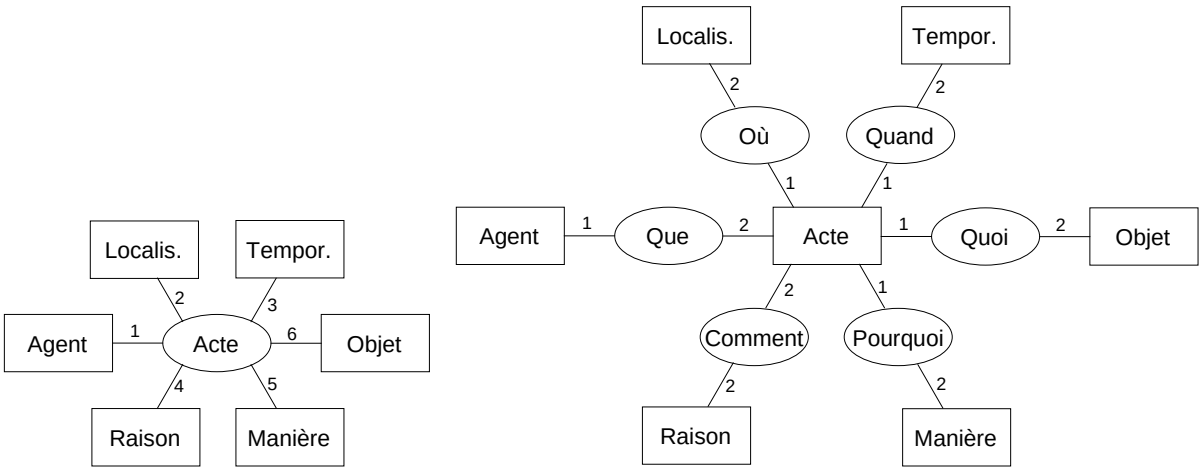


Figure 57 : Structure de description décrite au moyen des graphes conceptuels. En (a), le groupe fonctionnel d’Acte est la relation qui relie les différents groupes fonctionnels, et en (b), le groupe fonctionnel d’Acte est un concept au même titre que les autres.

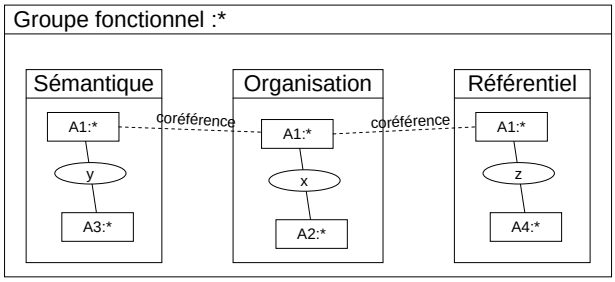


Figure 58 : Exemple de description d’un groupe fonctionnel.

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

4 Discussion

Le point de départ de notre proposition est l'utilisation du questionnement partiel en mode direct afin d'accéder à la description du terme. Une telle démarche est déjà adoptée par ailleurs. A titre d'exemple, la métadonnée OpenMI distingue les quatre classes d'information What, When, Where, How (Gregersen et al., 2007), qui constitue un sous-ensemble de notre base de départ (Qui, Quoi, Où, Quand, Comment, Pourquoi). Dans d'autres disciplines, l'ensemble formé par ces pronoms et adverbes interrogatifs, appelé hexamètre de Quintilien, est déjà utilisé au quotidien par les journalistes ou les manageurs de la qualité, pour s'assurer que l'information est caractérisée la plus complètement possible. Par rapport à ces exemples, l'originalité de notre travail réside (i) dans la façon d'aborder ces adverbes et pronoms interrogatifs sous forme de groupe fonctionnel et d'identifier une structure partagée des réponses, et (ii) d'établir des liens entre les groupes fonctionnels au moyen du principe d'incidence.
La définition des classes Organisation, Sémantique et Référentiel que nous proposons renvoie à des travaux existants. Parmi ceux-ci, on peut citer l'ontologie des classes de concepts temporels utilisés dans la description des contenus des pages Web et des propriétés des services Web (Hobbs et Pan, 2006). Cette ontologie comporte les classes 'TemporalEntity', 'DurationDescription', 'DateTimeDescription', 'TemporalUnits', et 'DayOfWeek'. La classe 'TemporalEntity' est constituée des sous-classes 'Instant' et 'Interval'. La sous-classe 'Interval' est constituée de la sous-classe 'PropertyInterval', elle-même comportant la sous-classe 'DateTimeInterval'.
Cette ontologie s'inscrit dans un référentiel temporel classique correspondant à la position de la terre par rapport au soleil. Dans la mesure où de nombreux phénomènes biologiques (morphogénèse, etc.) s'expriment en fonction de la température, cette ontologie ne peut pas s'appliquer à notre étude. Un autre exemple est fourni par la base de données WOLF (WORDNET Libre du Français). WOLF est une base de données lexicale sémantique pour le français (Sagot et Fiser, 2008), dans laquelle les mots sont répartis en catégories et organisés hiérarchiquement. Cette construction ontologique présuppose qu'il est possible de recenser tous les sens des mots. Dans notre étude, les programmes considérés sont des produits de la recherche, où l'innovation conduit à proposer des sens nouveaux aux mots, voire à créer des mots nouveaux. Pour cette raison, cette ressource n'a pas été considérée dans l'étude.

Pour l'assemblage des groupes fonctionnels, nous avons envisagé d'utiliser la théorie X-barre proposée par Chomsky (1970). Cette théorie s'intéresse aux relations syntaxiques des mots au sein de la phrase, entre lesquels des relations de subordination semblent exister. Cette théorie utilise les notions de syntagme, d'arbre lexical, etc., qui sont des outils de la grammaire. L'objectif de notre étude étant d'accéder au sens implicite d'un terme, et non de produire des phrases, cette piste a été rejetée au profit du principe d'incidence.
Enfin, pour la description du contenu des contextes, Lénat (1998) recherche une méthode de caractérisation complète du contexte. Ses travaux, conduits dans le cadre du projet CYC (http://www.cyc.org), l'ont amené à décrire le contexte comme un point situé dans un espace à douze dimensions : Absolute Time, Type of Time, Absolute Place, Type of Place, Culture, Sophistication/Security, Granularity, Epistemology, Argument-Preference, Topic, Justification, and Let's (and misc. domain assumptions). Chacune de ces dimensions a été obtenue à partir de critères de pertinence de regroupement de concepts au sein d'une base de connaissances. Ces critères considèrent par exemple l'effet sur la base de connaissances de séparer des sous-ensembles de connaissances, la facilité de traiter informatiquement le chevauchement et la séparation des sous-ensembles, la perception familière des critères, etc. Pour notre objectif de description du contexte d'élaboration d'un terme, nous avons préféré

adopter une structure de description identique à celle du contenu du programme. Ce faisant, nous avons émis l'hypothèse que notre proposition permettrait d'exprimer les mêmes éléments que ceux identifiés par Lenat (1998). Par exemple, les dimensions Absolute Time, Type of Time, Absolute Place et Type of Place sont redevables des groupes fonctionnels de Temporalité et de Localisation. Les dimensions Culture, Topic et Justification sont appréhendées par les groupes fonctionnels D'Actant, d'Acte, et de Raison. Les dimensions Sophistication/Security, Epistemology et Let's sont redevables de la Manière. Les dimensions Granularity et Argument-Preference n'apparaissent pas explicitement dans notre construction. La dimension Granularity permet de formuler la notion d'emboitement d'éléments, que l'on retrouve dans la classe Référentiel. La dimension Argument-Preference permet de formuler des préférences entre connaissances. Cette notion de préférence, faisant référence à une règle d'inférence, n'existe pas dans notre proposition. Elle sera vraisemblablement traitée plus tard lors de l'introduction des notions de gradation et de scalarité dans le modèle formel.
Une autre façon d'aborder le contexte est celle proposée par Brézillon (2003). Brézillon (2003) situe le contexte dans une logique d'action appliquée à des situations particulières, tel que l'accomplissement d'une tâche par un opérateur au moyen d'un logiciel. Le contexte est défini comme la réunion de données « contextuelles », qui renseignent sur la présence et l'état des éléments environnants immédiats précédant l'action, et de données « externes », qui sont complémentaires aux données contextuelles, mais qui ne sont pas directement utiles. L'objet des travaux de Brézillon (2003) est d'étudier la dynamique du contexte qui a amené l'opérateur à accomplir la tâche. Ce travail ne correspond pas à notre objectif d'étude.

L'emprunt fait à la linguistique ne prétend pas à l'exhaustivité, loin s'en faut. Certaines notions nécessitent d'être complétées, à l'exemple de la classe référentiel qui n'est pas finalisée. D'autres notions méritent d'être revisitées. Ainsi, en § 2.1.4 (chapitre B), la notion de personne a permis de proposer une typologie du nombre (figure 32), utilisée ensuite pour constituer la classe référentiel de l'actant (§ 2.2.1.2). Dans la littérature, (Leeman-Bouix, 2009) introduit la notion de foule et de structure de foule. Dans une perspective de description de la nature par l'agronome, la notion de foule pourrait être intéressante pour le groupe fonctionnel de l'actant, afin de représenter une population de plantes ou d'insectes par exemple.

Suite à l'identification d'une structure commune de description des termes, le modèle formel a été implémenté au moyen des graphes conceptuels (§ 3.3). Son implémentation a soulevé une question concernant la représentation du concept et de la relation. Une première option consiste à considérer qu'un verbe est nécessairement représenté par une relation, et la seconde qu'un verbe peut être représenté par un concept. La première option offre un cadre clair et rigoureux, et la seconde offre l'avantage de permettre au concept de représenter autre chose qu'un substantif. Pour répondre à cette question, une solution est de faire appel aux travaux des linguistes. Par exemple, Leeman-Bouix (2009) cite les travaux de Fourquet (1950), qui fait remarquer que, si l'opposition fondamentale entre le nom et le verbe est habituelle (verbe = action ou état, et nom = être ou chose), les noms peuvent exprimer des actions (course, lutte, etc.), des états (sommeil, mort, etc.) et des qualités (blancheur, intelligence), au même titre que les verbes (présider, rougeoyer). De façon à pouvoir répondre à l'option retenue, une structure de graphe de connaissances a été proposée pour chacune.

La constitution de dictionnaires s'effectue à partir des classes Sémantiques. Trois types de dictionnaires ont été identifiés : le lexique, le vocabulaire somme et le vocabulaire somme disjointe. Le lexique rassemble les vocables, indépendamment de leur usage et du sens conféré par la relation de subsomption dans les classes Sémantique. Dans ce dictionnaire, un

tel-00458265, version 1 - 19 Feb 2010

concept n'est présent qu'une seule fois. Le vocabulaire somme comporte les arborescences des vocables reliés au moyen de la relation de subsomption, indépendamment de leur appartenance à un groupe fonctionnel. Dans ce dictionnaire, un vocable peut être présent plusieurs fois, et disposer par conséquence de différents sens. L'omission du groupe fonctionnel implique que l'usage du vocable n'est pas considéré dans la spécification de son sens. Le vocabulaire somme comporte les arborescences des vocables reliés au moyen de la relation de subsomption, où l'arborescence tient compte du groupe fonctionnel. Dans le cas où un vocable se trouverait plusieurs fois dans le vocabulaire, et que le sens accordé dans chaque groupe fonctionnel soit identique, c'est l'usage du vocable qui fait la différence. Ces trois dictionnaires présentent donc une gradation en termes d'accès au sens implicite des vocables, depuis le lexique, qui omet le sens implicite des vocables, jusqu'au vocabulaire somme disjointe, qui comporte l'usage. Dans l'objectif d'accéder au sens implicite des entrées/sorties des programmes, il est nécessaire de disposer du sens qui soit le plus précis. Il semble donc pertinent d'adopter le vocabulaire somme disjointe.

D'autre part, que ce soit pour le Modèle Formel ou les Graphes Conceptuels, le vocabulaire est constitué par un ensemble de concepts reliés au moyen de la relation de subsomption. Le sens d'un concept est précisé par la concaténation des sens des concepts du sous-graphe situé en amont. La façon d'appréhender le vocabulaire diffère toutefois entre le Modèle Formel et les Graphes Conceptuels. Pour le Modèle Formel, un même concept peut disposer de plusieurs sens dans le vocabulaire somme disjointe. Pour les Graphes Conceptuels, un concept n'apparaît qu'une seule fois dans le vocabulaire, et ne dispose donc que d'un seul sens. Le mécanisme de raisonnement des Graphes Conceptuels est établi selon cette approche. Disposer de plusieurs sens pour un même concept nécessite de revisiter le mécanisme de raisonnement. Cette différence d'approche du vocabulaire est donc une limite à l'usage des Graphes Conceptuels pour l'implémentation du Modèle Formel que nous proposons.

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

Chapitre C
Illustration

A ce stade du travail, la proposition de modèle formel doit être confrontée à l'exercice de décrire des termes pour les associer. C'est ce que nous abordons dans ce chapitre au travers d'un exemple. De cet exercice, nous tirons des enseignements pour alimenter la discussion et identifier la suite à donner au présent travail de thèse.

1 Exemple de description d'un système complexe

Le modèle formel a été présenté dans le chapitre B comme une méthode de description d'un terme en vue d'accéder aux connaissances pragmatiques. Ce modèle formel peut également être utilisé pour décrire l'assemblage des termes, i.e. des programmes de simulation. Dans cette approche, (i) les termes sont représentés par les groupes fonctionnels d'Agent et d'Objet, (ii) l'action d'un terme sur un autre (ou sur lui-même) est exprimée par les groupes fonctionnels d'Agent (celui qui produit l'acte), d'Acte et d'Objet (celui qui subit l'acte), et (iii) les entrées du terme Agent sont indiquées par les groupes fonctionnels de Localisation, de Temporalité, de Manière et de Raison.

L'exemple choisi pour illustrer la mise en œuvre du modèle formel porte sur la relation mutualiste existant entre deux bio-agresseurs de la vigne : *Botrytis cinerea* (*B. cinerea*), un champignon pathogène, et *Lobesia botrana* (*L. botrana*), un insecte phytophage (Mondy et Corio-Costet, 2000). La vigne, *B. cinerea* et *L. botrana* sont des organismes étudiés de façon disjointe dans le cadre de domaines thématiques différents, respectivement la physiologie végétale, l'entomologie et la phytopathologie.

Dans l'exercice, nous supposons que les termes vigne, *B. cinerea* et *L. botrana* ont été développés indépendamment et nous nous intéressons à leur assemblage dans le cadre de la relation mutualiste. L'assemblage de ces termes nécessite la mise en place de plusieurs interrelations. L'assemblage est appréhendé sous la forme d'un terme, appelé système biologique. Dans le § 1.1, le système biologique est décrit selon un ensemble d'actions élémentaires, et exprimé sous forme littérale. Chaque action, correspondant à un acte produit par un terme sur un autre, est référencée au moyen d'une lettre. Cette lettre est utilisée dans la figure 61 du § 1.2 pour établir la correspondance entre la description littérale de l'action et sa représentation, obtenue en appliquant le modèle formel. La description du système biologique et la construction du vocabulaire afférent ont été produites en collaboration avec un agronome. La superposition des graphes de connaissances permet d'accéder à l'ensemble des interrelations à mettre en place pour construire le terme « système biologique ». Le vocabulaire généré caractérise le sens des entrées des termes vigne, *B. cinerea* et *L. botrana*. D'un point de vue théorique, cet exemple permet d'illustrer l'intérêt de la méthode pour traiter de la polysémie et de la polyphonie.

tel-00458265, version 1 - 19 Feb 2010

1.1 Description littérale du système biologique

A : En l'absence de *B. cinerea*, les larves de *L. botrana* se nourrissent des grains de raisin sur lesquels ils provoquent des dégâts, i.e. des lésions de la cuticule protectrice des grains et des galeries dans les grains (Clouvel et al., 2008). La vigne en tant qu'hôte fournit aux larves de *L. botrana* les stérols nécessaires à leur développement.

B : Parmi ces nutriments, la vigne produit des stérols, utilisés par les larves pour produire le cholestérol nécessaire à la biosynthèse des hormones de mue.

C : Le cycle de *B. cinerea* comporte une phase de dispersion aérienne par le vent sous forme de conidies, et une phase de développement dans la plante hôte sous forme de mycelium.

D : En l'absence de lésion de la cuticule des grains, la pénétration du pathogène au travers de la cuticule se fait de façon active grâce à la production d'une enzyme produite par les conidies lors du processus de germination. La germination n'est toutefois possible que dans des conditions de température et d'humidité favorables des grappes, en relation avec le climat de la parcelle agricole et l'exposition particulière des grappes sur les plants.

E : En présence de lésions, occasionnées par *L. botrana*, le champignon pathogène se comporte comme un saprophyte et contamine les grains dont la cuticule est fragilisée. Pour *B. cinerea*, la présence de *L. botrana* permet une contamination opportuniste des grains de façon passive, et élargit la fenêtre de contamination vis-à-vis des conditions d'humidité de l'air ambiant.

F : De plus, par leur activité, les larves contribuent à la dissémination des conidies sur les grappes et à l'intérieur des grains dont ils se nourrissent.

G : Par ailleurs, le champignon synthétise naturellement des stérols.

H : En consommant les grains contaminés par *B. cinerea*, les larves *L. botrana* ont accès à une source de stérols supplémentaire par rapport aux stérols produits par la vigne au niveau des grains.

I : *L. botrana* présente 5 stades larvaires successifs, au terme desquels l'insecte se transforme en chrysalide puis en adulte ailé. La consommation de stérols d'origine fongique se traduit par plusieurs avantages. Le premier réside dans une réduction de la durée de vie larvaire.

J : La réduction de la durée de vie larvaire réduit les risques d'évènements mortifères (prédateurs, lessivage par pluie etc..). Enfin, indépendamment de phénomènes extrinsèques, les stérols d'origine fongique augmentent la probabilité de passage d'un stade larvaire au suivant. Globalement, les stérols d'origine fongique augmentent la probabilité d'accéder aux stades chrysalide puis adulte.

1.2 Représentation formelle du système biologique

La formalisation du système biologique consiste en son écriture sous forme de graphe de connaissances. Le support d'implémentation adopté est celui des graphes conceptuels. La modalité adoptée pour produire le graphe de connaissances est celle exposée en figure 57a. Le groupe fonctionnel d'Acte est implémenté sous la forme d'une relation qui l'associe aux autres groupes fonctionnels (Agent, Objet, Temporalité, Localisation, Raison, Manière). D'après le Modèle Formel (figure 53, Description multi-niveaux d'un terme), la formulation d'un graphe de connaissances s'effectue selon différents niveaux (structure de description, groupes fonctionnels et classes) imbriqués au moyen de concepts de composition (figure 59a). La lecture du graphe de connaissance s'effectue selon l'ordre donné par le principe

d'incidence : l'Agent commet un Acte subi par l'Objet, cet Acte ayant été produit, dans une Localisation particulière, dans une Temporalité particulière, pour une Raison particulière et selon une certaine Manière. Dans l'implémentation des graphes de connaissance au moyen des graphes conceptuels, le principe d'incidence est exprimé par l'arité de la relation (1 = Agent, 2 = Localisation, 3 = Temporalité, etc.).
De façon à rendre la formulation des graphes de connaissances lisible dans le document, une représentation simplifiée en est donnée : (i) les marqueurs instance des concepts de la classe Organisation ont été inscrits dans le groupe fonctionnel duquel ils sont redevables, indépendamment des concepts de composition (flèches de couleur dans la figure 59), et (ii) la nature du groupe fonctionnel (Agent, Objet, Temporalité, etc.) est indiquée en entête du concept de composition qui le représente (figure 59b).

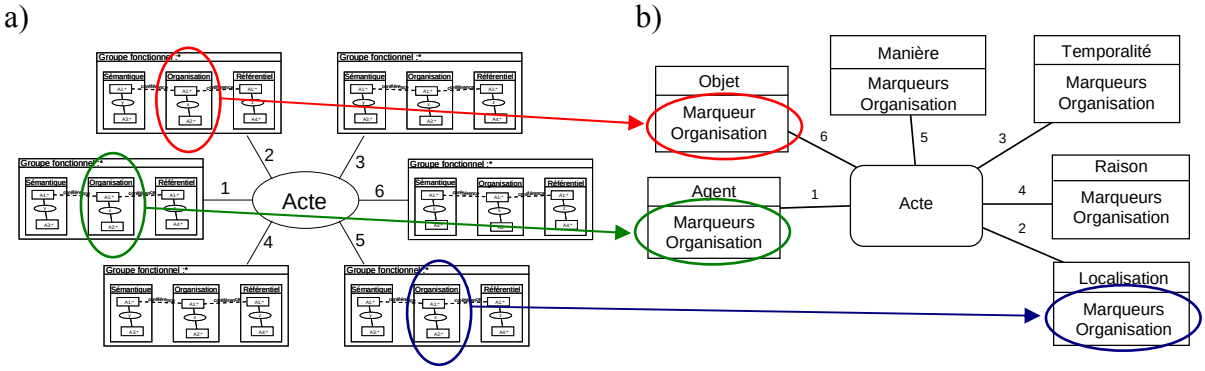


Figure 59 : simplification de la présentation d'un graphe de Connaissances, dans lequel les marqueurs de la classe Organisation sont mis en avant (a = graphe complet, b = représentation simplifiée).

La formalisation du système biologique sous forme de graphes de connaissances s'effectue à partir de sa description littérale. L'identification des différents groupes fonctionnels s'obtient par l'interrogation. A la question « Qui fait quoi ? », 'Qui' fait référence au groupe fonctionnel d'Agent, 'fait' à celui de l'Acte, et 'Quoi' à celui de l'Objet. De la même façon, l'utilisation des adverbes interrogatifs 'Où', 'Quand', 'Comment' et 'Pourquoi', renseignent sur les groupes fonctionnels de Localisation, de Temporalité, de Manière et de Raison. La réponse à la question « Où 'qui' a 'fait' 'quoi' ? » renseigne sur le groupe fonctionnel de Localisation. De la même façon, la réponse « Quand 'qui' a 'fait' 'quoi' ? » renseigne sur le groupe fonctionnel de Temporalité, et idem pour les adverbes Comment et Pourquoi pour respectivement la Manière et la Raison.

Le contexte d'élaboration de l'illustration est la description du système biologique par un agronome du CIRAD, réalisée au moyen du Modèle Formel, à Montpellier, en 2009. A la question « Qui fait quoi ? », la réponse est « l'agronome du CIRAD a décrit le système biologique ». Le groupe fonctionnel d'Agent est renseigné par l'expression « l'agronome du CIRAD ». L'agronome est l'élément informatif du groupe fonctionnel d'Agent, et « du CIRAD » son référentiel. Pour la classe Organisation, l'expression « l'agronome » est un élément permanent (§ 2.2.1.1, chapitre B). La classe Organisation du groupe fonctionnel d'Agent est donc constituée du marqueur 'Agronome', instance du concept 'Permanent'. Le CIRAD étant une entité d'inclusion, les marqueurs instances des concepts de la classe Référentiel du groupe fonctionnel d'Agent sont 'l'agronome' et 'le CIRAD'. Ce même raisonnement, présenté ici pour le groupe fonctionnel d'Agent, est appliqué aux autres groupes fonctionnels. Il s'en suit que les classes Organisation du groupe fonctionnel d'Objet,

tel-00458265, version 1 - 19 Feb 2010

de Temporalité, de Localisation, de Manière et de Raison sont respectivement constituées du marqueur 'système biologique', '2009', 'Montpellier', 'Modèle Formel', et 'Illustration thèse' (figure 60).

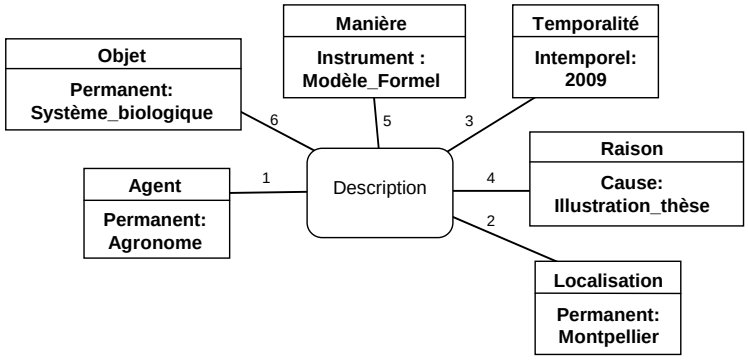


Figure 60 : Graphe de connaissances du contexte d'élaboration de l'illustration.

La formalisation des actions élémentaires du système biologique, énoncées dans le § 1.1, en graphes de connaissances est présentée en figure 61. Chaque action a été transcrite sous forme de graphe élémentaire en utilisant la même démarche que celle présentée ci-dessus (transcription du contexte en un graphe de connaissances). Par exemple, la description littérale de l'action A est la suivante : « En l'absence de *B. cinerea*, les larves de *L. botrana* se nourrissent des grains de raisin sur lesquels ils provoquent des dégâts, i.e. des lésions de la cuticule protectrice des grains et des galeries dans les grains (Clouvel et al., 2008). La vigne en tant qu'hôte fournit aux larves de *L. botrana* les stérols nécessaires à leur développement. ». A la question « Qui fait Quoi », la réponse est « la larve consomme les stérols produits par la vigne ». Dans cette réponse, l'Agent est la larve, l'Acte la consommation, et l'Objet les stérols de la vigne. D'après l'agronome qui a produit la description, l'Agent et l'Objet sont des éléments permanents (figure 45, p. 76) vis-à-vis de l'Acte de consommation. Pour le graphe de connaissances, cette description correspond à produire (i) un marqueur larve instance du concept Permanent de la classe Organisation pour le groupe fonctionnel d'Agent, (ii) une relation intitulée 'Consommation' pour représenter l'Acte, et (iii) un marqueur stérol larve instance du concept Permanent de la classe Organisation pour le groupe fonctionnel d'Objet.
Toujours pour l'action A, à la question « Quand est-ce que qui fait quoi », l'agronome indique « le climat de la vigne combiné du stade du grain de raisin ». Le climat de la vigne et le grain de raisin sont alors représentés par deux descripteurs notés respectivement Climat_vigne et G_raisin. D'après l'agronome, ces deux concepts s'inscrivent dans la durée (figure 45, p. 76). Il s'en suit que dans le graphe de connaissance relatif à l'action A (figure 61), le groupe fonctionnel de Temporalité comporte deux marqueurs 'Climat_vigne' et 'Stade_G_raisin' instance du concept « Durée ». Bien que ce ne soit pas notifié dans le groupe fonctionnel de Temporalité du graphe de connaissance, ces marqueurs sont reliés au moyen d'une relation associative (Et logique). Quels que soient le graphe de connaissances et le groupe fonctionnel considérés, la présence de plusieurs marqueurs au sein d'un groupe fonctionnel correspond à la relation associative. D'autre part, l'absence de marqueur dans un groupe fonctionnel indique que l'information relative à ce groupe fonctionnel vis-à-vis de l'action n'est pas fournie dans la description, et que l'agronome considère l'information négligeable. C'est le cas, par exemple, du groupe fonctionnel de Manière pour l'action A (figure 61).

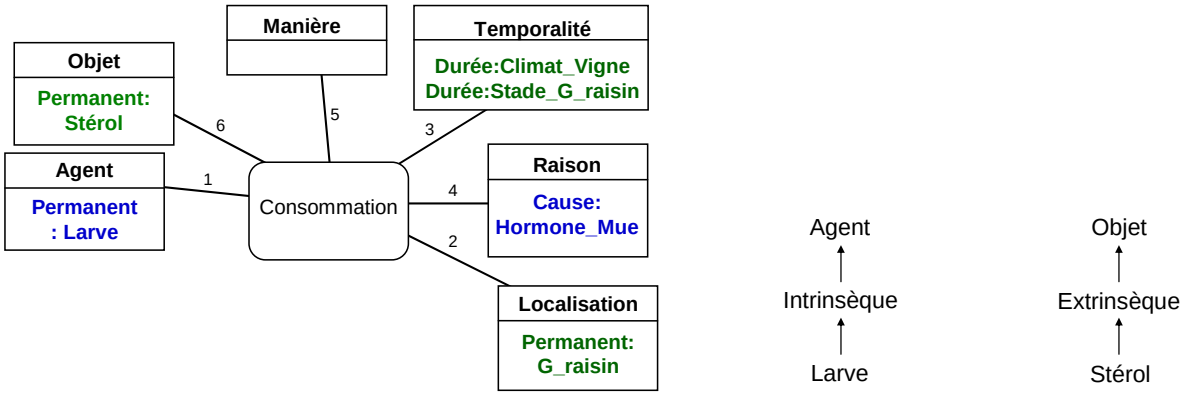
Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

tel-00458265, version 1 - 19 Feb 2010

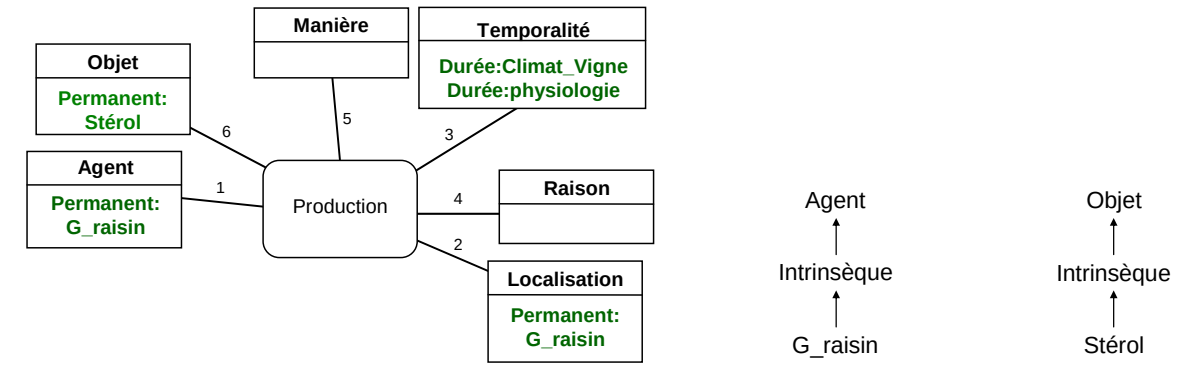
En figure 61, pour chaque action, le graphe de la classe Sémantique du groupe fonctionnel de l'Agent et celui de l'Objet sont portés en vis-à-vis du graphe de connaissance correspondant. Ces graphes ont été établis sur la base de la figure 46 relative à la typologie de la sémantique de l'Actant, qui différentie le caractère intrinsèque du caractère extrinsèque de l'Actant. Pour mémoire, l'Actant extrinsèque correspond à un actant « étranger » à l'Actant qui produit l'Acte. Dans la figure 46, le caractère extrinsèque est décomposé en Politesse, Familiarité, Honorifique, et Tabou. Dans le cadre de l'illustration, la notion de Personne comme Actant n'est pas pertinente pour le système biologique considéré. Par conséquent, pour cette illustration, seule la distinction intrinsèque / extrinsèque est conservée. Pour l'action A (figure 61) par exemple, les deux Actants Agent et l'Objet sont respectivement les marqueurs Larve et Stérol. L'Agent est nécessairement intrinsèque à lui-même vis-à-vis de l'Acte. Pour ce qui concerne l'Objet, l'Agent étant redevable de *L.Botrana* et l'Objet (le stérol) de la vigne, le Stérol est donc qualifié d'extrinsèque (à l'Agent Larve). Dans le cadre de l'action B (figure 61) par contre, le stérol est produit par la vigne. Ce qui confère à l'Objet Stérol le statut d'intrinsèque vis-à-vis de l'Agent G_raisin (le grain de raisin).

Dans les graphes de connaissances montrés en figure 61, des couleurs ont été attribuées aux termes. Les concepts relatifs aux termes *L.botrana*, à *B.cinerea* et à la vigne sont respectivement notés en bleu, rouge et vert. Ce jeu de couleur permet d'identifier le terme dont le marqueur est redevable. Par exemple, pour l'action A (figure 61), les marqueurs 'Larve' et 'Hormone-mue' sont de couleur bleue, donc redevables du terme *L.botrana*, les marqueurs 'Stérol', 'Climat_vigne' et 'G_raisin' sont de couleur verte, donc redevables du terme vigne. Pour la constitution du graphe de la classe Sémantique, la différence de couleur entre le marqueur considéré et le marqueur de l'Agent permet d'identifier immédiatement si celui-ci est intrinsèque ou extrinsèque à l'Agent.

A

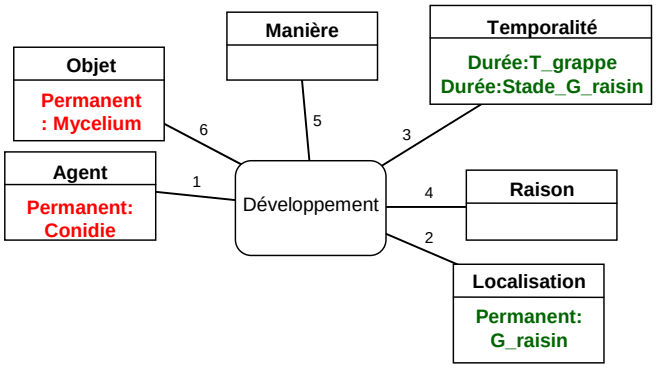


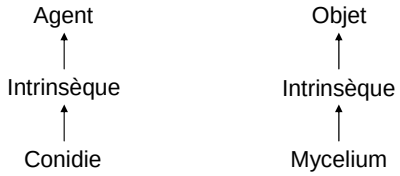
B



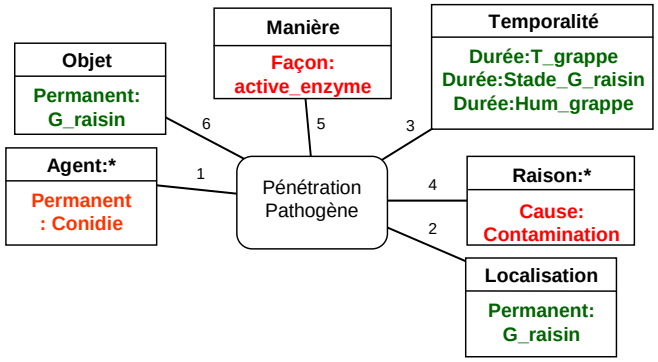
tel-00458265, version 1 - 19 Feb 2010

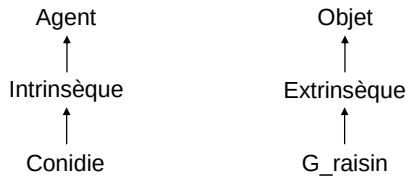
C



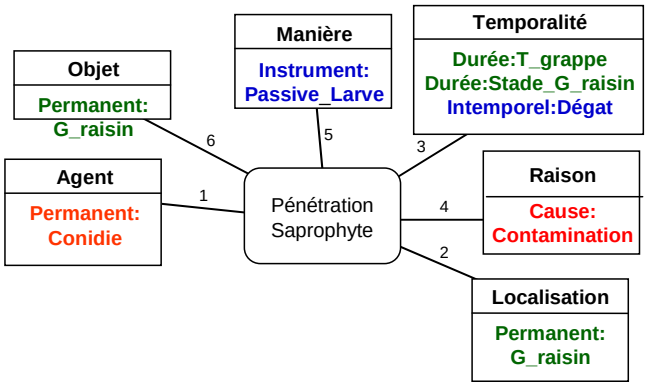


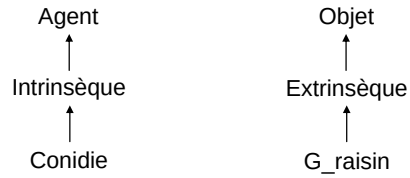
D





E

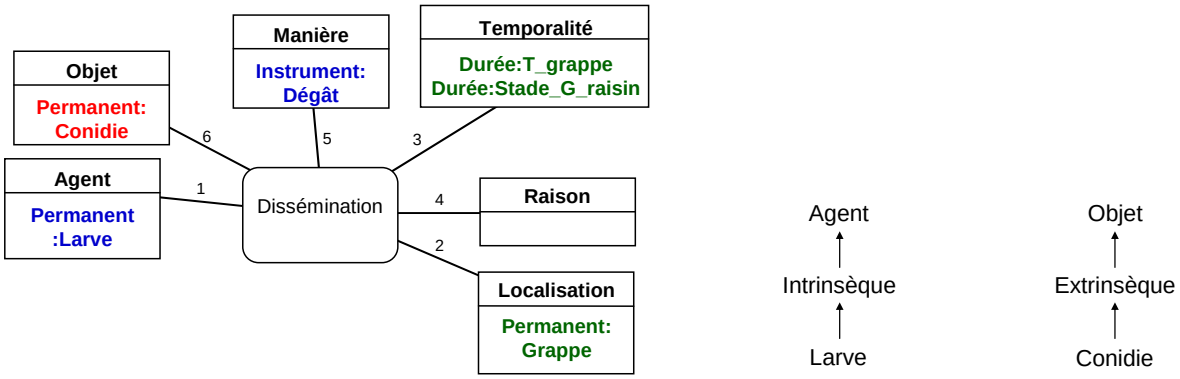




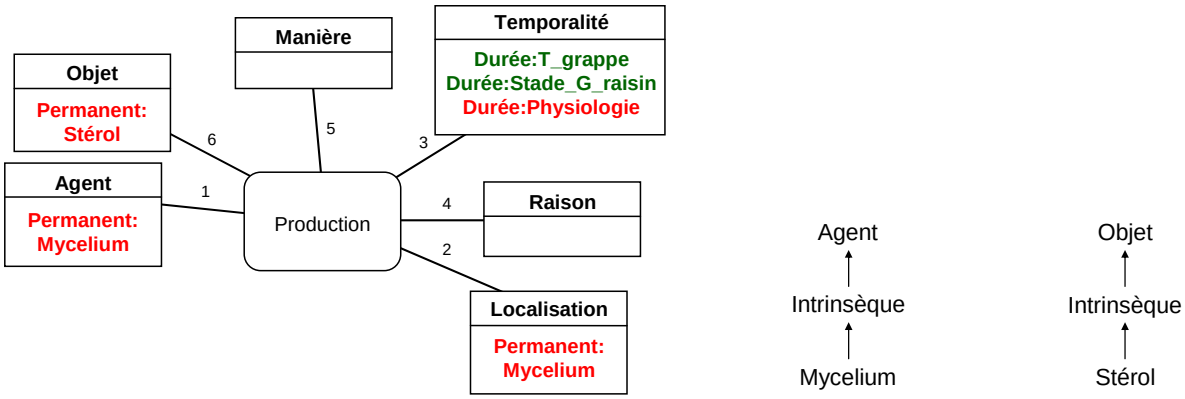
tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

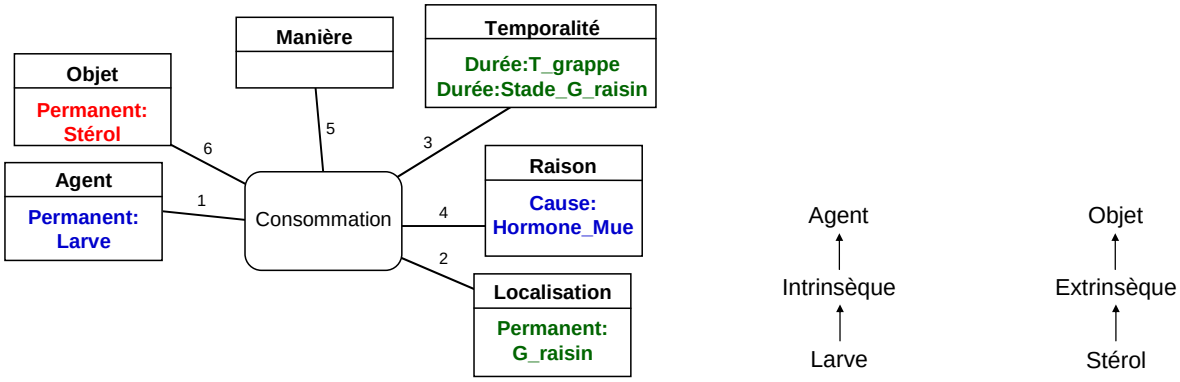
F



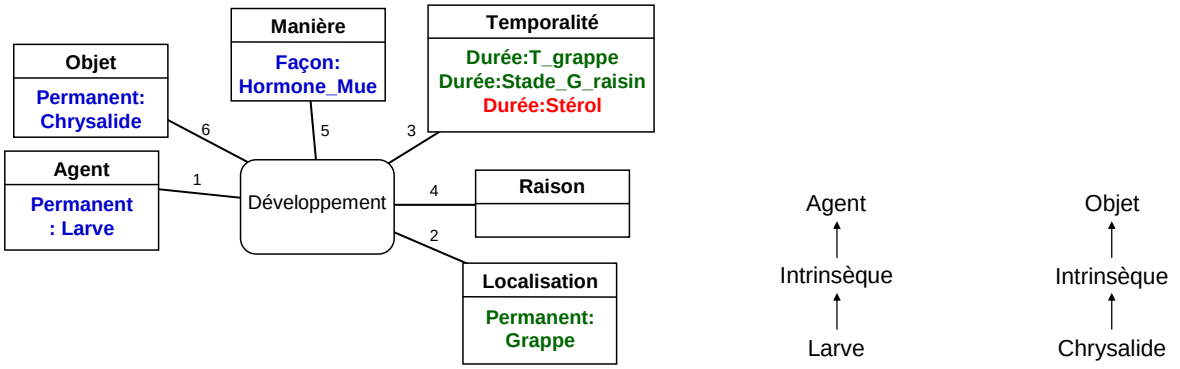
G



H



I



J

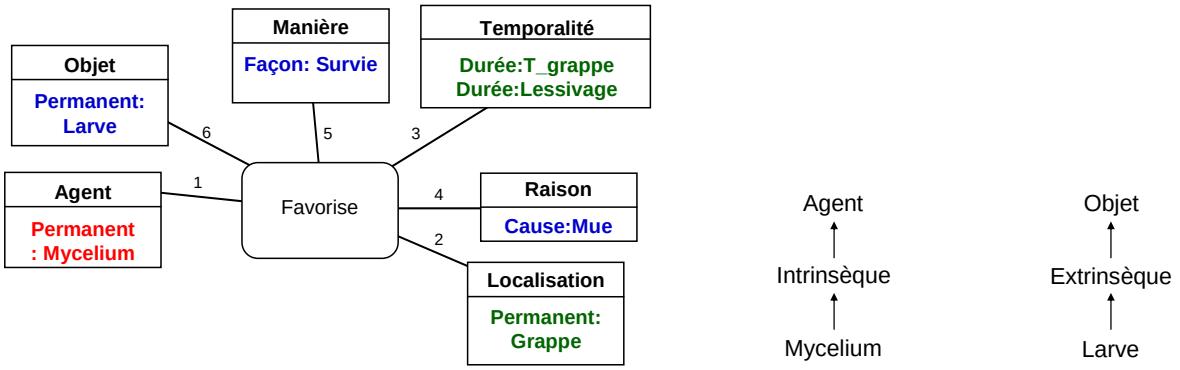


Figure 61 : Graphes de connaissances, représentant la relation mutualiste de *Lobesia botrana* et *Botrytis cinerea* en regard de la vigne (colonne de gauche), et classes Sémantiques des groupes fonctionnels d'Agent et d'Objet (colonne de droite). Les concepts associés à *L. botrana*, *B. cinerea* et la vigne sont respectivement notés en bleu, rouge et vert. Les concepts en noir correspondent à d'autres entités naturelles intervenant sur le système biologique considéré.

1.3 Vocabulaire et sens implicite

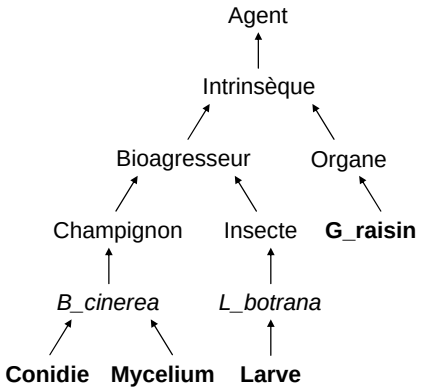
En référence à la production de dictionnaires (lexique et vocabulaires) dans le modèle formel, le vocabulaire est établi en réunissant les classes sémantiques des groupes fonctionnels. Pour rappel, deux types de vocabulaires sont possibles selon la fonction mathématique utilisée (somme vs somme disjointe). Dans le cas présent, nous procédons à une somme disjointe afin de conserver la sémantique portée par la nature du groupe fonctionnel (Actant, Localisation et Temporalité dans l'illustration). La construction du vocabulaire s'effectue de proche en proche et par généralisations successives des concepts. Cette opération de généralisation demande au spécialiste du domaine, l'agronome en l'occurrence, d'expliciter le sens implicite des concepts. Par exemple, dans le vocabulaire de l'Actant (figure 62a), le concept 'B_cinerea' est introduit pour généraliser les concepts 'Conidie' et 'Mycelium'. A un niveau supérieur de l'arborescence, les concepts 'B_cinerea' et 'L_botrana' sont généralisés par le concept 'Bioagresseur', modulo les concepts intermédiaires 'Insecte' et 'Champignon' qui en complètent le sens en regard des énoncés A à J. La figure 62 présente la somme disjointe des classes Sémantiques pour les groupes fonctionnels d'Agent, d'Objet, de Localisation et de Temporalité.

Le vocabulaire du groupe fonctionnel d'Agent (figure 62a) correspond à la réunion des classes Sémantiques de l'Agent de la figure 61. Les quatre concepts identifiés, à savoir 'Conidie', 'Mycelium', 'Larve', et 'G_raisin' héritent du concept intrinsèque.
Le vocabulaire du groupe fonctionnel d'Objet correspond à la réunion des classes Sémantiques d'Objet de la figure 61 (figure 62b). Au sein de ce vocabulaire, le concept Stérol se retrouve sous les arborescences initiées par les concepts Intrinsèque et Extrinsèque. Le chemin en aval du concept Stérol diffère selon qu'il hérite des concepts Intrinsèque et Extrinsèque. Dans l'arborescence Intrinsèque, le concept 'Stérol' est qualifié de 'Produit', tandis que dans l'arborescence Extrinsèque, il est qualifié de 'Nutriment'. Cette différence de chemin traduit une variation du sens de Stérol. En tant qu'Intrinsèque, le Stérol est un produit élaboré par deux agents différents (G_raisin et mycelium), alors qu'en Extrinsèque, il est

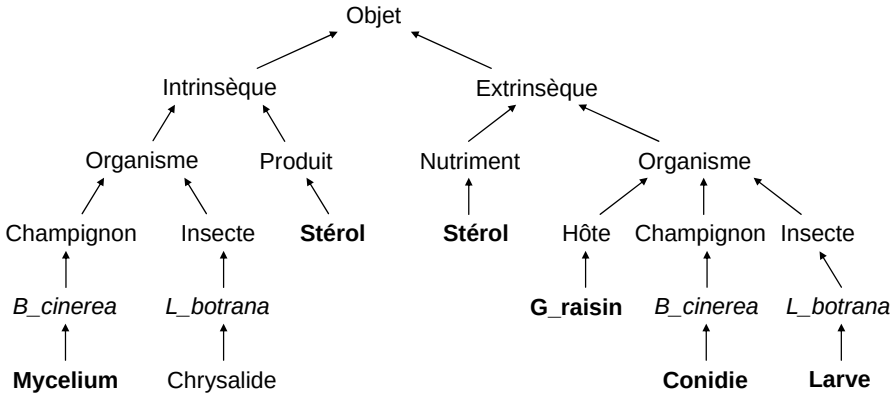
tel-00458265, version 1 - 19 Feb 2010

nutriment. Ainsi, bien que ce soit la même entité biologique qui est considérée, son sens diffère.
Le vocabulaire du groupe fonctionnel de Localisation est moins étoffé que celui de l'Agent (figure 62c). Il comporte néanmoins un concept doublon : le concept G_raisin. De la même façon que dans le vocabulaire du groupe fonctionnel d'Objet, les chemins en amont du concept diffèrent selon qu'il est qualifié d'Intrinsèque ou d'Extrinsèque. En tant qu'Intrinsèque, le grain de raisin est siège de réactions biochimiques en lien avec le métabolisme de la plante, et tant qu'Extrinsèque, c'est un support au service de l'activité biologique des bio-agresseurs (substrat nutritif et protection mécanique).
Le vocabulaire du groupe fonctionnel de Temporalité ne comporte pas de doublon (figure 62d). Par contre, il comporte le concept Stérol, qu'on retrouve également dans le vocabulaire des groupes fonctionnels d'Agent et d'Objet. Il est intéressant de noter ici, qu'un concept du même nom peut officier en tant qu'Actant (Agent ou Objet) et en tant que Temporalité (action I : effet du stérol sur la durée de vie larvaire de *L.botrana*).
Les vocabulaires des groupes fonctionnels d'Acte, de Manière et de Raison, offrant peu d'intérêt quant à l'illustration (pas de concept que l'on retrouve dans le vocabulaire des autres groupes fonctionnels), ne sont pas présentés dans ce document.

a)

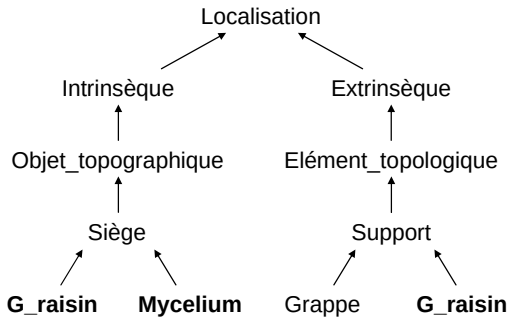


b)



tel-00458265, version 1 - 19 Feb 2010

c)



d)

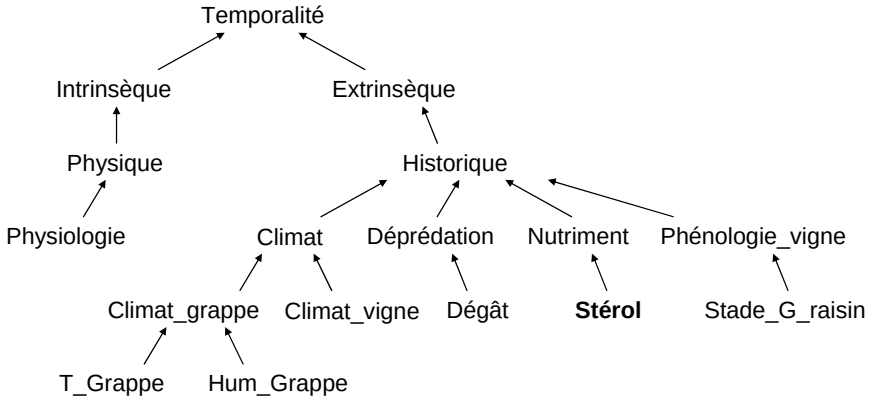


Figure 62 : Vocabulaires des groupes fonctionnels d'Agent (a), d'Objet (b), de Localisation (c) et de Temporalité (d). Les concepts écrits en caractère gras sont présents dans différents vocabulaires.

1.4 Transcription informatique

Dans le § 1.2 (chapitre A), trois règles syntaxiques d'assemblage des termes sont identifiées : R1 qui comporte l'ordre de juxtaposition des termes au moyen des relations série et parallèle, R2 qui permet la mise en coïncidence des entrées/sorties des termes, et R3 qui définit la méthode physique de transmission des données entre les termes. Dans la mesure où R3 est liée au support et au langage d'assemblage, nous ne prenons pas cette règle en compte dans l'illustration.

Ordre de juxtaposition : R1 et S1

L'illustration est constituée des termes *L.botrana*, *B.cinerea* et vigne assemblés au moyen du terme système biologique. Dans l'objectif de simuler le système complexe présenté dans l'illustration, construire R1 du système biologique revient à identifier l'ensemble des interrelations à établir entre les termes *L.botrana*, *B.cinerea* et la vigne. Dans le § 1.1 (chapitre C), le système biologique, i.e. le terme d'assemblage, a été décrit par actions élémentaires. Chaque action est ensuite transcrite, dans le § 1.2 (chapitre C), sous la forme d'un graphe de connaissances.

tel-00458265, version 1 - 19 Feb 2010

Deux natures d'interrelations sont présentées dans chaque graphe de connaissances. La première concerne l'Agent qui produit l'Acte sur l'Objet. Cette Action correspond à la mise en place d'une interrelation entre le terme Agent et le terme Objet, où l'exécution du terme Agent précède celle du terme Objet. La superposition des interrelations produites par les Agents sur les Objets indique la nature des interrelations que le terme système biologique doit établir entre les termes vigne, *B.cinerea*, et *L.botrana* (figure 63). En matière de sémantique (S1), il y a tout d'abord une action produite par le terme *L.botrana* sur le terme vigne (action A). Deux actions d'un terme sur l'autre doivent être produites en parallèles: (i) *L. botrana* sur *B. Cinerea* par les actions H et F, et (ii) de *B.cinerea* sur la vigne par les actions D et E. Deux actions d'un terme sur deux autres doivent être produites en parallèle (schéma 10 de la figure 02 : les 18 formes d'interrelation entre deux éléments monovalents) : (i) *L.botrana* sur vigne et *B.cinerea* par, respectivement, les actions A et F et/ou H, et (ii) *B.cinerea* sur *L.botrana* et Vigne par, respectivement, les actions J et D et/ou E. L'interaction (= la relation mutualiste) entre *B.cinerea* et *L. botrana* doit être établie (F, H et J). Enfin, les actions B, C, G et I représentent une rétroaction pour les trois termes *L.botrana*, *B.cinerea* et la vigne.

La seconde nature d'interrelation émane des groupes fonctionnels de Temporalité, Localisation, Manière et de Raison. D'après le principe d'incidence, ces groupes fonctionnels renseignent sur les éléments qui caractérisent l'acte produit par l'Agent. Il s'en suit que les marqueurs de la classe Organisation situés dans ces graphes de connaissances correspondent aux entrées dont a besoin le terme Agent pour produire la simulation.
Pour l'Action A par exemple (figure 61-A), le terme *L.botrana* (Agent) consomme (Acte) des stérols (Objet) du terme vigne (couleur verte du texte dans le graphe de connaissances). Pour produire sa simulation, le terme *L.botrana* requiert donc, en données d'entrée, le climat de la vigne (groupe fonctionnel de Temporalité) et d'informations relatives au grain de raisin (groupes fonctionnels de Temporalité et de Localisation). Ces données étant d'une couleur différente à celle de l'Agent (vert vs. bleu), la transmission des données du terme disposant de ces données d'entrée, le terme vigne en l'occurrence, au terme *L.botrana* doit précéder la simulation de *L.botrana*.
Ainsi, pour disposer de l'ensemble des interrelations à mettre en place entre les termes, la figure 63 doit être complétée par cette seconde nature d'interrelations.

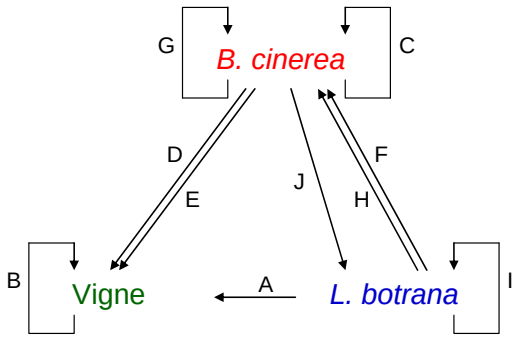


Figure 63 : Interrelations entre *B_cinerea*, *L. botrana* et le grain de raisin de la vigne.

Mise en coïncidence des entrées / sorties : R2 et S2

En ce qui concerne R2, nous avons vu que le graphe de connaissances indique les données d'entrée nécessaires à la simulation du terme (spécifié dans le groupe fonctionnel d'Agent)

tel-00458265, version 1 - 19 Feb 2010

pour conduire un acte en regard d'un autre terme. Pour chaque donnée d'entrée, la classe Sémantique du groupe fonctionnel comporte le sens de cette donnée d'entrée, sens qui est attendu par le terme. Ce sens est précisé dans le vocabulaire somme disjointe. Pour l'action A par exemple (figure 61-A), une couleur identique du texte du marqueur du groupe fonctionnel de Raison et de celui du groupe fonctionnel d'Agent indique que la donnée d'entrée spécifiée par le marqueur du groupe fonctionnel de Raison provient de l'Agent (intrinsèque). Le groupe fonctionnel de Localisation indique que G_raisin est une donnée d'entrée extrinsèque (couleur verte). Le vocabulaire de Localisation indique que le sens de cette donnée d'entrée est un élément topologique qui supporte l'action. Le groupe fonctionnel de Temporalité indique deux données d'entrée émanant du terme vigne. Le sens de ces deux données d'entrée est fourni par le vocabulaire de Temporalité. De façon à permettre l'action du terme *L.botrana* sur le terme vigne, le terme vigne doit donc transmettre les données requises au terme *L.botrana*, dont le sens est défini dans le vocabulaire. L'impossibilité pour le terme vigne de fournir une donnée selon le sens requis impliquerait que l'action en question ne peut être implémentée.
D'autre part, le vocabulaire indique également le sens dans lequel le terme indiqué dans le groupe fonctionnel d'Objet est appréhendé par le terme Agent. Pour l'action A, par exemple, le terme qui subit l'action est vigne. Dans ce cas, l'agent *L.botrana* appréhende l'Objet vigne comme une source de stérol. Le stérol, qualifié d'extrinsèque dans la classe Sémantique du groupe fonctionnel d'Objet (figure 61-A, colonne de droite), est défini dans le vocabulaire du groupe fonctionnel d'Objet (figure 62b) comme nutriment. Ainsi, pour cette action, le terme *L.botrana* appréhende le terme vigne sous le sens d'un nutriment. La mise en place de l'interrelation (effet produit) du terme Agent sur le terme Objet doit par conséquent tenir compte de ce sens particulier.

2 Discussion

L'illustration porte sur un système complexe redevable de trois domaines thématiques : physiologie végétale, entomologie et phytopathologie. Cette illustration a été produite par un agronome ayant accès à ces domaines thématiques. La description du système biologique, i.e. le terme d'assemblage, par un seul individu (l'agronome) permet, dans le cas présent, de lever les problèmes de mises en coïncidence de vocabulaires d'origines scientifiques plurielles.
L'objet de cette illustration est de présenter le mode opératoire permettant d'accéder au sens implicite des termes. Dans le § 1.2, chaque partie de la description littérale du système biologique est transcrite sous la forme d'un graphe de connaissances (figure 61). Cette transcription a consisté à classer les informations disponibles dans la description littérale selon les groupes fonctionnels (Actant, Temporalité, etc.). De cette classification, les classes Sémantiques associées à chaque graphe de connaissances (colonne droite de la figure 61 pour l'Agent et l'Objet) ont été produites en prenant pour base les typologies de la sémantique des groupes fonctionnels (figure 24, 46, 47, 48 respectivement pour les groupes fonctionnels de Localisation, d'Actant, de Temporalité, et de Manière).
Le sens implicite s'obtient à partir du vocabulaire construit par l'agronome. La construction du vocabulaire s'est effectuée par regroupement des classes Sémantiques puis par généralisation successive des concepts. L'intérêt de cette opération de généralisation est qu'elle demande à l'agronome d'énoncer le sens implicite de chaque concept en regard des autres. Par exemple, pour le groupe fonctionnel de l'Agent, l'introduction du concept B_cinerea généralise les concepts Conidie et Mycelium. Cette généralisation indique que, pour le groupe fonctionnel de l'Agent, ces deux concepts héritent du sens du concept B_cinerea. Ce dernier est appréhendé sous la forme du champignon, lui-même appréhendé

tel-00458265, version 1 - 19 Feb 2010
Rapport-Gratuit.com

sous la forme d'un bio-agresseur. Si les concepts Conidie et Mycelium se trouvaient réunis dans un autre groupe fonctionnel et redevables du concept intrinsèque (ou extrinsèque), on pourrait être tenté de réutiliser cette généralisation. Ce mode opératoire sous-tendrait alors que, quelque soit la situation, Conidie et Mycelium sont des bio-agresseurs. Or, le vocabulaire montre que ce n'est pas toujours le cas. Ainsi, dans le vocabulaire du groupe fonctionnel de l'Objet (figure 62b), le concept Stérol est un Produit dans le cas Intrinsèque, et un Nutriment dans le cas Extrinsèque. L'identification des concepts unificateurs ne peut donc pas être résolue par transposition de généralisations produites par ailleurs. La généralisation n'est donc pas une opération automatique. Celle-ci est liée au sens implicite que véhicule chaque concept vis-à-vis de l'usage qui en est fait.
La production du vocabulaire somme disjointe permet de différencier les variations de sens des concepts au sein d'un groupe fonctionnel. Il permet également de comparer les variations de sens des concepts entre groupes fonctionnels. Le concept G_raisin, par exemple, se situe dans les vocabulaires de l'Agent, de l'Objet et de Localisation (figures 62a, 62b et 62c). Dans le vocabulaire de l'Agent, le concept G_raisin hérite des concepts Organe et intrinsèque. Dans celui de l'Objet, G_raisin hérite du concept Hôte, Organisme et Extrinsèque. Dans celui de la Localisation, G_raisin hérite des concepts Siège, Objet_topographique, et Intrinsèque. Dans ces trois vocabulaires, les concepts hérités par le concept G_raisin varient indépendamment du concept Intrinsèque ou extrinsèque. Le concept Stérol se retrouve également dans plusieurs vocabulaires. Dans le vocabulaire de l'Objet (figure 62b), il hérite soit des concepts Nutriment et Extrinsèque, soit des concepts Produit et Intrinsèque. Dans le vocabulaire relatif au groupe fonctionnel de Temporalité (figure 62d), Stérol hérite des concepts Nutriment, Historique et Extrinsèque. De même que G_raisin, les concepts hérités varient indépendamment du concept Intrinsèque ou extrinsèque. L'entendement de ces concepts varie donc selon les cas.
En linguistique, les variations de sens des concepts G_raisin et Stérol peuvent amener à qualifier ces concepts de polysémiques (un mot possède plusieurs sens littéraux) ou de polyphoniques (différents entendements d'un même sens littéral) (Ducrot et Schaeffer, 1999). La question qui se pose est celle de la mise en correspondance de ces formes de variations de sens distinguées par la linguistique avec celles relevées dans les vocabulaires de l'illustration. En corollaire, cette illustration présente l'intérêt de mettre en évidence que, dans le monde du vivant, les notions de Localisation et de Temporalité ne s'expriment pas uniquement en termes de position géographique, pour la localisation, ou en termes de position de la terre par rapport au soleil et de données climatiques (température et humidité notamment), pour la Temporalité. Par exemple, le stérol accélère le développement de la *L.botrana*. Ces notions peuvent également être exprimées en termes d'éléments naturels.

L'illustration, présentée en début de ce chapitre, aurait pu concerner une des plateformes agronomiques analysées dans le chapitre A. Et en particulier la plateforme agronomique APES, dans la mesure où nous avons pris part à son développement. L'illustration proposée présente l'avantage d'aborder une autre forme de complexité. Dans le chapitre A, le système complexe a été défini comme un ensemble d'éléments en interaction. Pour rappel, les plateformes agronomiques DSSAT, APES et SEAMLESS-IF implémentent (en termes de S1) l'action et la rétroaction. Ces plateformes répondent donc positivement à la définition de système complexe (un ensemble d'éléments en interaction). Le langage d'assemblage varie selon la plateforme. La version 4 de la plateforme agronomique DSSAT est le fruit d'un développement ad hoc en regard d'une question particulière. Les plateformes APES et SEAMLESS-IF ont été construites en adoptant un support générique. MODCOM a d'ailleurs dû être modifié pour répondre aux exigences d'APES (la rétroaction). Pour APES et SEAMLESS-IF, la méthode adoptée pour choisir le support générique n'est pas mentionnée

tel-00458265, version 1 - 19 Feb 2010

dans la littérature. Dans la mesure où les modifications de MODCOM ont été produites en cours de projet, on peut se poser la question des raisons qui ont mené au choix d'utiliser ce support générique, pour être modifié.
Le fonctionnement de chaque entité biologique est implémenté sous forme de termes. La question posée par cette illustration est de l'assemblage de ces termes pour représenter un système complexe. La figure 63 présente les interrelations entre les termes *B_cinerea*, *L. botrana* et la vigne. En termes de S1, ce système biologique comporte de l'Action (A), de la rétroaction (B, C, G et I), de l'interaction (F, H, J) et du parallélisme (H et F, et D et E), et tout cela combiné. Les besoins en données d'entrée de chaque Agent sont indiqués, pour chaque Action, dans les groupes fonctionnels de Temporalité, de Localisation, de Manière et de Raison des graphes de connaissances (figure 61). De façon à connaître les interrelations supplémentaires existant entre les termes, il est nécessaire de superposer le graphe exprimant le besoin en données d'entrée de chaque Agent à celui de la figure 63, tout comme nous l'avons fait pour APES (figure 14). Les interrelations, existant entre les termes, dépassent celles requises par les plateformes APES, DSSAT et SEAMLESS-IF. Implémenter un tel système au moyen d'une plateforme logicielle représente un enjeu en regard des questions agronomiques émergeantes en lien avec l'écologie. Placé dans le cadre de la réutilisation, le choix du langage d'assemblage doit tenir compte des interrelations à mettre en place.

Le modèle formel est une proposition de réponse à la question d'accès aux connaissances pragmatiques. Le contexte d'élaboration, adjoint à la description du terme, permet de situer le terme dans l'histoire, en regard des connaissances disponibles et des paradigmes de représentation des connaissances alors en vigueur. Dans le cadre d'une réutilisation extrinsèque, le contexte informe sur les éléments prévalant à la construction du terme (chercheurs, objectifs, etc.). Ce faisant, il permet d'évaluer la compatibilité du terme existant avec l'attendu.
La limite de la méthode proposée réside dans la capacité du chercheur à préciser le sens de chaque concept au sein du vocabulaire. L'apparition de nouvelles connaissances influe également sur la description du sens des concepts, qui de fait s'érode avec le temps. Ainsi, le sens d'un concept ne peut être manipulé indépendamment de son contexte.

tel-00458265, version 1 - 19 Feb 2010

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

tel-00458265, version 1 - 19 Feb 2010

Conclusion

La thèse traite de l'assemblage de programmes au sein de plateformes logicielles selon les points de vue syntaxique, sémantique et pragmatiques. Dans ce paragraphe, nous répondons aux trois questions énoncées dans l'introduction et donnons des pistes pour une suite de ce travail.

Première question de la thèse : La première question porte sur l'identification des aspects implicites du sens dans le langage d'assemblage. Pour répondre à cette question, la première étape du travail a consisté à caractériser ce langage d'assemblage. Dans le §1.1 du chapitre A, les éléments bibliographiques ont montré que la notion de plateforme logicielle n'était pas partagée par la communauté informatique. La réunion des définitions a permis de définir la plateforme logicielle comme un support d'accueil d'éléments exogènes.
La plateforme peut être dotée d'un langage dévolu à l'assemblage des éléments, dont la composition constitue un ensemble opérationnel. Le processus d'assemblage a été décomposé dans le § 1.2 (chapitre A) en trois parties : interrelation entre les termes, mise en coïncidence des entrées/sorties des termes, et connexion physique des entrées/sorties des termes.
La seconde étape (§1.3, chapitre A) a consisté en l'organisation des éléments bibliographiques selon les quatre niveaux de connaissances énoncés par Sabah (1988). Cette organisation a donné lieu à la trame d'analyse DRS. Au niveau morphologique, les termes sont caractérisés par la notion de valence. Au niveau syntaxique, une règle syntaxique est associée à chaque partie du processus d'assemblage : R1 pour l'interrelation des programmes, R2 pour la mise en coïncidence des entrées/sorties (métadonnées et ontologies), et R3 pour l'établissement physique des entrées/sorties des termes. De ces trois règles syntaxiques, seule R2 s'intéresse explicitement au traitement sémantique (S2). Il s'en suit que S2 se situe au niveau sémantique. Les aspects implicites du sens dans le langage d'assemblage sont situés au niveau pragmatique. On y retrouve la sémantique S1 et S3 des règles R1 et R3. S1 octroie une sémantique aux différentes combinaisons d'interrelation entre couple de termes : action, interaction, rétroaction, parallélisme. S3 interprète la modalité d'établissement physique des connexions au moyen de la notion de niveau de couplage.
Dans la trame DRS, la valence est considérée comme un élément morphologique. Lors de la construction de la plateforme agronomique APES, les termes monovalents ont été transformés en termes quadrivalents. La valence du terme véhicule donc également un aspect implicite du sens, dans la mesure où elle signifie que le terme est décomposé en parties élémentaires.

Seconde question de la thèse : Dans la seconde question de la thèse, nous investiguons dans quelle mesure les caractéristiques morphologiques, syntaxiques et sémantiques du langage d'assemblage limitent la capacité à représenter les systèmes complexes étudiés par les sciences du vivant. Par système complexe, nous entendons un ensemble d'éléments en interaction. D'après cette définition, la capacité de représentation des systèmes complexes est par conséquent exprimée au moyen de la sémantique S1 de la règle R1.
Cette question a été approchée en analysant un panel de plateformes logicielles existantes. Une première étude (§ 2, chapitre A) a porté sur trois langages d'assemblage utilisés par les sciences du vivant, appelés supports génériques dans ce document. L'intérêt du choix du panel est de disposer de langages d'assemblage contrastés en termes de valence (mono/multivalent), d'interrelations permises (action/rétroaction/toutes interrelations) et de niveau de couplage (faible/fort). Le passage au crible de la trame d'analyse DRS des supports génériques a montré que les S1 de MODCOM, d'OpenMI et de VLE sont respectivement l'action, la

rétroaction et l'interaction, et la sémantique de toutes les formes d'interrelations envisagées dans la figure 02. Par conséquent, OpenMI et VLE ont la capacité de représenter, en l'état, les systèmes complexes, ce qui n'est pas le cas de MODCOM.

La seconde étude a porté sur la représentation des systèmes complexes par les sciences du vivant (§ 3, chapitre A). Le dispositif d'étude est constitué des plateformes agronomiques DSSAT, APES et SEAMLESS-IF. Ce dispositif présente une gradation, en termes disciplinaire et de domaine thématique, des programmes assemblés : DSSAT réunit des programmes portant sur un même domaine thématique, APES réunit des programmes de domaines thématiques différents mais appartenant à une même discipline, et SEAMLESS-IF réunit des programmes de domaines disciplinaires différents. Ces plateformes agronomiques présentent certaines originalités intéressantes vis-à-vis de notre étude. L'originalité de DSSAT repose sur l'histoire d'une plateforme dont la vocation initiale (version 2.1) était le partage d'outils de gestion de données, et qui, pour mettre en relation les programmes, a finalement (version 4.0) nécessité la restructuration des programmes et l'adoption d'un sous-programme sol identique. L'originalité d'APES est d'utiliser MODCOM pour représenter un système complexe, alors que ce support générique n'est initialement pas prévu pour. Vis-à-vis de la question, c'est la mise en coïncidence des entrées/sorties des termes qui nous intéresse (R2 et S2, § 3.3.2, chapitre A). Pour SEAMLESS-IF, cette opération correspond à l'établissement de la correspondance entre langages scientifiques. Pour APES, il s'agit d'établir la correspondance de vocabulaires redevable d'un même langage scientifique mais relatif à des domaines scientifiques différents, et pour DSSAT la correspondance de vocabulaires de domaines scientifiques identiques. L'étude relative à la constitution du dictionnaire (§4, chapitre A) a mis en exergue que, bien que le sens littéral des entrées/sorties soit identique, leur sens implicite diffère. Dans l'objectif de représenter les systèmes complexes, la limite posée pour l'assemblage est due à l'existence de sens implicite inhérent aux entrées/sorties des termes. L'accès aux connaissances pragmatiques pour renseigner ce sens littéral pose donc question (§ 5, chapitre A).

Troisième question de la thèse : La dernière question porte sur l'identification d'une méthode pour accéder aux connaissances pragmatiques dans le cadre de l'assemblage de programmes. La méthode proposée est le modèle formel de description présenté au chapitre B. Le modèle a été établi en réutilisant les travaux conduits par une autre discipline, la linguistique en l'occurrence.
L'interrogation partielle en mode direct permet de proposer une structure de description basée sur les sept pronoms et adverbes interrogatifs offerts par la langue française. La bibliographie relative à ces pronoms et adverbes interrogatifs nous a permis d'identifier une structure commune de réponses à l'interrogation partielle, qui s'applique à la fois à la description du programme et à celle du contexte d'élaboration (§ 2, chapitre B).
La formalisation de la structure commune de description est la condition *sine qua non* de son utilisation par les plateformes logicielles. La proposition du Modèle Formel a été établie en utilisant des éléments de la Linguistique. Ce modèle formel s'adresse à l'exercice particulier pour lequel il a été développé, c'est-à-dire l'accès aux connaissances pragmatiques dans le cadre de l'assemblage de termes. Cette proposition, qui s'inspire de travaux sur le langage, ne prétend donc pas apporter une cognition dans le domaine linguistique.

Les suites à ce travail : Dans ce travail, nous posons la question de l'accès au sens implicite. La méthode développée a été élaborée à partir des réponses fournies par les pronoms et adverbes interrogatifs Qui, Que, Quoi, Où, Quand, Comment et Pourquoi. Or, dans les

tel-00458265, version 1 - 19 Feb 2010

plateformes logicielles en question, les programmes simulent des phénomènes particuliers (biophysique, économique, etc.) afin d'accéder à des aspects qualitatifs et quantitatifs. Ces aspects ne sont pas pris en compte dans notre proposition. Du point de vue linguistique, ces aspects relèvent des notions de gradation et de scalarité. Dans la suite de notre travail, traiter ces aspects reviendrait à compléter l'étude par l'adverbe « Combien ».
Dans notre étude des supports génériques, l'aspect numérique n'a été qu'effleuré : MODCOM et VLE permettent respectivement l'intégration numérique du temps et des temps/espace/état. OpenMI n'offre quant-à-lui aucune fonction d'intégration. Les méthodes d'intégration du temps adoptées dans MODCOM et VLE sont les mêmes (Runge-Kutta, Euler, etc.). Pour l'espace, VLE emploie des méthodes reposant sur l'opposition discret / continu (Amblard et Phan, 2006). D'autres supports génériques existent pour simuler les systèmes complexes. Pour les concepteurs de Modelica (http://www.modelica.org/), la question de l'intégration numérique spatiale ne peut pas se résoudre à la seule considération discret / continu. Pour autant, aucune solution n'est proposée. Il serait intéressant d'étudier le sens attribué aux notions d'espace et de temps par l'équipe en charge du développement des supports génériques. En effet, l'adoption d'une même méthode d'intégration numérique pour tous les termes insérés dans le support générique présuppose du partage de la notion de Temporalité par les termes, au niveau du sens et du référentiel. La mise en coïncidence des notions de Temporalité et de Localisation des termes fait référence à la notion d'échelle, omniprésente dans la représentation des systèmes complexes. Vis-à-vis de la proposition du Modèle Formel et de son application à l'assemblage de programmes, l'étude du combien conduira à traiter de la notion d'échelle et des mises en correspondance des échelles entre termes.

Par ailleurs, dans le mémoire, nous montrons une application du modèle de description à un système complexe constitué de trois entités biologiques redevables de domaines thématiques différents (physiologie végétale, entomologie et phytopathologie). Pour établir cette application, les connaissances ont été apportées par un agronome familier des trois domaines thématiques. Cette situation, pour laquelle un seul individu est en mesure de décrire plusieurs programmes redevables de domaines thématiques différents, est particulière. Dans le cas où les entités biologiques seraient décrites par des individus différents, les questions se posent de (i) comment confronter les vocabulaires de spécialistes, et (ii) quelle règle adopter pour mettre en coïncidence des vocabulaires établis dans des contextes différents. Ce problème, qui reste à résoudre, conduit à aborder la question de la relation entre la description et l'énonciateur. Que ce soit pour enrichir l'existant ou pour intégrer les aspects qualitatifs et quantitatifs des programmes dans le modèle formel, la suite de ce travail nécessite d'établir une collaboration avec des linguistes, et ceci autour d'applications précises.
D'autre part, le vocabulaire « somme disjointe » permet de distinguer les variations de sens des entités biologiques. La variation de sens attribué au « stérol » par exemple en est l'illustration. Ces variations de sens correspondent à une pluralité de perceptions d'une entité biologique selon l'actant. Cet aspect est une question redevable de la philosophie, et en particulier de la phénoménologie. Badiou (2006), propose de représenter ces variations de perceptions en utilisant la méthode mathématique des topos (théorie des catégories). Selon l'auteur, cette méthode permet d'insérer les diverses perceptions des entités d'un système dans un espace donné, appelé transcendantal, et de les organiser relativement selon leur « degré d'apparaître » dans le système. Dans l'exemple de système complexe que nous avons pris, les différentes natures d'interrelation (action, interaction, parallélisme et rétroaction) sont combinées. La complexité des interrelations laisse présager l'apparition de conflits d'exécution des programmes informatiques (termes) dans le cas où des interrelations opposées ou concurrentes pour un même programme devraient être simulées simultanément. La construction d'un transcendantal pourrait être une solution à l'identification de la priorité

tel-00458265, version 1 - 19 Feb 2010

d'exécution des programmes. Il serait intéressant d'envisager l'applicabilité des travaux de Badiou (2006) au Modèle Formel pour l'assemblage des programmes.

Enfin, au-delà des questions soulevées dans le cadre strict de la thèse, considérer l'assemblage de programmes sous l'angle pragmatique amène à concevoir autrement les plateformes logicielles. Dans notre travail par exemple, nous montrons qu'un programme émetteur doit être en mesure de proposer autant de données de sortie que le vocable a de sens. Du point de vue de l'informaticien, intégrer la pragmatique revient à élargir le concept de plateforme, actuellement circonscrite au rôle de structure d'accueil d'éléments exogènes**,** à celui de structure de composition d'éléments exogènes. Cette fonctionnalité dévolue à la plateforme implique un changement dans la relation entre disciplines informatique et les sciences du vivant, qui d'un contenu technique exogène devra évoluer vers un processus de co-construction dans le cadre d'une réelle interdisciplinarité. Le modèle formel que nous proposons s'inscrit bien dans l'établissement de ces nouvelles collaborations interdisciplinaires.
Hors cadre de la réutilisation, ce travail propose une façon différente d'aborder les systèmes complexes. A l'occasion de l'illustration proposée par exemple, nous montrons que la notion de temporalité pouvait s'exprimer d'une façon élargie par rapport au cadre strict du temps usuel (les stérols comme éléments de temporalité par exemple). Le modèle formel que nous proposons représente une méthode déclarative de description des systèmes complexes. A l'heure actuelle, les systèmes complexes sont simulés sur la base d'équations dynamiques composées. Nous pensons que le modèle formel proposé pourrait représenter un support intéressant pour la formalisation du système complexe et la composition des équations dynamiques. Le travail de thèse ouvre donc un champ de recherche pour (i) décrire le fonctionnement des systèmes complexes dans le cadre des disciplines du vivant, en particulier pour exprimer les notions d'échelle (gradation et scalarité) et (ii) développer les programmes afférents d'un point de vue informatique.

tel-00458265, version 1 - 19 Feb 2010

Glossaire

Dictionnaire : dans le chapitre A, le dictionnaire correspond à la liste des programmes insérés dans la plateforme logicielle. Dans le chapitre B (§ 3.2) et le chapitre C, le dictionnaire rassemble les vocables situés dans la classe Sémantique des groupes fonctionnels. Dans ce dernier cas, trois dictionnaires sont distingués : le lexique, le vocabulaire somme, et le vocabulaire somme disjointe.

Groupe fonctionnel : ensemble de mots formant une seule unité obtenue en réponse à un questionnement partiel en mode direct, c'est-à-dire utilisant un pronom ou adverbe interrogatif Qui, Que, Quoi, Où, Quand, Comment, Pourquoi (§ 1.2, chap. B). L'analyse du groupe fonctionnel à l'usage du Modèle Formel (§ 2.2.1, chap. B), a conduit à décomposer celui-ci selon les classes Organisation, Sémantique et Référentiel.

Niveaux de connaissances : au sein du langage, Sabah (1988) distingue quatre niveaux de connaissance : les connaissances morphologiques des termes (forme canonique du terme, racines et terminaisons, emploi), les connaissances syntaxiques qui expriment les relations entre les termes sous forme de règles, les connaissances sémantiques pour l'expression du sens littéral, et enfin les connaissances pragmatiques qui permettent d'accéder aux aspects implicites du sens. (§ 1.3, chap. A).

Plateforme logicielle : structure d'accueil d'éléments exogènes (§ 1.1, chap. A). Cette structure d'accueil peut être dotée d'un langage dévolu à l'assemblage des éléments, dont la composition constitue un ensemble opérationnel.

Plateforme logicielle agronomique : plateforme logicielle rassemblant des programmes redevables de l'agronomie. A l'objectif final de produire des logiciels, la plateforme agronomique ajoute également la fonction de rassembler et de faire collaborer des équipes d'origine disciplinaire et géographique différentes. De ce fait, les chercheurs amenés à contribuer à la plateforme y amènent leurs méthodes, outils et concepts/paradigmes qu'il s'agit de faire coïncider. Les connaissances pragmatiques, inhérentes à ces méthodes, outils et concepts, sont donc importantes dans ces constructions logicielles.

Principe d'incidence : en grammaire, ce principe met en rapport au sein de la phrase le support, qui est ce dont on parle, et l'apport, qui est ce qui est dit (§ 1.3.3, chap. B). L'incidence externe de premier degré s'adresse aux mots en relation directe avec le support, et l'incidence interne de second degré opère où un mécanisme d'incidence est déjà en place. Dans le modèle Formel, le principe d'incidence est utilisé comme structure syntaxique d'assemblage des groupes fonctionnels (§ 2.2.2, chap. B). Le support est le groupe fonctionnel d'Agent, et les autres groupes fonctionnels sont des apports. L'incidence externe au premier degré est le groupe fonctionnel d'Acte. Les groupes fonctionnels d'Objet, de Temporalité, de Raison et de Manière sont incidents au groupe fonctionnel d'Acte (incidence interne de second degré).

Reformatage : restructuration du code source du programme inséré dans la plateforme logicielle (§ 2.4 et § 4.1, chap. A).

tel-00458265, version 1 - 19 Feb 2010

R1 et S1 : R1 est une règle syntaxique du langage d'assemblage de la plateforme logicielle. R1 comporte l'ordre de juxtaposition des programmes exprimé au moyen des relations série et parallèle (§ 1.3, chap. A). En termes sémantique, S1 est le sens accordé à l'ordre de juxtaposition des programmes formulé par R1. S1 s'exprime en utilisant les notions d'action, d'interaction, de rétroaction et de parallélisme. S1 est un sens implicite redevable du niveau pragmatique (cf. niveaux de connaissances).

R2 et S2 : R2 est une règle syntaxique du langage d'assemblage de la plateforme logicielle. R2 permet la mise en correspondance des entrées/sorties des programmes en utilisant le sens littéral de celles-ci (§ 1.3, chap. A). En termes sémantique, S2 statut de la cohérence du rapprochement du sens des entrées/sorties des programmes produit par R2. S2 est un sens littéral redevable du niveau sémantique (cf. niveaux de connaissances).

R3 et S3 : R3 est une règle syntaxique du langage d'assemblage de la plateforme logicielle. R3 est la méthode physique de transmission des données entre les entrées/sorties des programmes (§ 1.3, chap. A). En termes sémantique, S3 est le sens accordé à la méthode de transmission R3 adoptée. S3 est exprimé en utilisant la notion de niveau de couplage (Faible, Moyen, Fort, etc.). S3 est un sens implicite redevable du niveau pragmatique (cf. niveaux de connaissances).

Structure de description : pour le Modèle Formel, graphe résultant de l'assemblage des groupes fonctionnels au moyen du principe d'incidence (§ 2.2.2, chap. B),

Support générique : plateforme logicielle comportant un langage d'assemblage. Le support générique sert de base technologique à la construction de plateformes logicielles agronomiques.

Terme : utilisé en référence à Sabah (1988) pour désigner une brique ou un programme à assembler.

Typage : adaptation des entrées/sorties du programme inséré dans la plateforme logicielle (§ 2.4 et § 4.2, chap. A).

Valence : propriété morphologique (cf. niveaux de connaissances) d'un terme. La valence indique le nombre de fonctions du programme qui peuvent être exécutées par un autre programme (§ 1.2, chap. A).

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

Références

Les deux chapitres portent sur des thématiques disciplinaires différentes. De façon à faciliter l'accès aux références bibliographiques, celles-ci ont été classées par chapitre. La première section comporte les références bibliographiques de l'auteur du mémoire, produites au cours de son doctorat.

1 Productions de l'auteur du mémoire

1.1 Projet Européen SEAMLESS (2005-2009)

Braudeau E., Mohtar R.H., El Ghezal N., Clouvel P. **Martin P.**, *A multi-scale "soil water structure" model based on the pedostructure concept*, Hydrolology and Earth System Science Discussion, vol. 6(1), p. 1111-1163 (2009).

Donatelli, M., Acutis M., Adam M., Braudeau E., Criscuolo L., Corbeels M., Dupraz C., Duru M., Gary C., Gottshalck P., Kansou K., Ewert F., Leffelaar P., **Martin P.**, Smith J., Trevisan M., Trevisiol P., van Evert F., *SEAMLESS APES - current status*, III CGMS Experts Meeting and GEOLAND Training Workshop, Arlon, Belgium, 23-25 October 2006.

Donatelli M., Russell G., Rizzoli A.E., Acutis M., Adam M., Athanasiadis I., Balderacchi M., Bechini L., Belhoucette H., Bellocchi G., Bergez J.E., Botta M., Braudeau E., Bregaglio S., Carlini L., Casellas E., Celette F., Ceotto E., Charron-Moirez M.E., Confalonieri R., Corbeels M., Criscuolo L., Cruz P., Di Guardo A., Ditto D., Dupra C., Duru M., Fiorani D., Gentile A., Ewert F., Gary C., Habyarimana E., Jouany C., Kansou K., Knapen M.J.R., Lanza Filippi G., Leffelaar P., Manici L., Martin G., **Martin P.**, Meuter E.C., Mugueta N., Mulia R., Noordwijk V.M., Oomen R., Rosenmund A., Rossi V., Salinari F., Serrano A., Sorce A., Vincent G., Theau J.P., Therond O., Trevisan M., Trevisio P., Van Evert F.K., Wallach D., Wery J., Zerourou A., *APES: The Agricultural Production and Externalities Simulator*, In: Brouwer F.M., van Ittersum M.K. (Eds.), Environmental and agricultural modelling: integrated approaches for policy impact assessment, Springer Academic Publishing, book chapter, (sous presse), 2010.

Martin P., Mohtar R.H., Clouvel P., Braudeau E., *Modeling Soil-Water Dynamics for Diverse Environmental Needs*, In: Voinov A., Jakeman A., Rizzoli A. (eds.), iEMSs Third Biennial Meeting: Summit on Environmental Modelling and Software, International Environmental Modelling and Software Society, July 9-12 2006, Burlington, Vermont, USA, pp. 6 (2007a).

Martin P., Libourel T., Clouvel P., Reitz P., *Migration revisited using the Category Theory: Application to Biological Modelling Languages*, Rapport de recherche LIRMM [lirmm-00388313 - version 1], pp. 15. (2008a).

Martin P., « *Soil water 2 – soil model* » *component*, logiciel libre, http://www.apesimulator.it/, 2009a.

Martin P., « *soil water 2 – parameter estimator* » *component*, logiciel libre, http://www.apesimulator.it/, 2009b.

Mohtar R., Braudeau E., Abou Najm M., Salahat M., **Martin P.**, *Kamel*., European Geosciences Union – General Assembly, poster session: Scale, Scaling, and Nonlinearity in the Earth's Surface, Soil and Solid Earth, Vienna, Austria, 2008, Geophysical research abstracts, vol. 10, EGU2008-A-10791 (2008).

1.2 Réseau excellence Endure

Martin P., Nibouche S., Clouvel P., *Simulation of bollworm damages on Cotton*. European Network for the Durable Exploitation of Crop Protection Strategies (ENDURE), Workshop on DSS for crop protection, Flakkebjerg, DK, March 17-19, 2008b.

1.3 Travaux en lien avec le mémoire

Clouvel P., Lecoeur J., Gérardeaux E., Luquet D., Rey H., **Martin P.**, *With sub-optimum conditions the current representation of potential crop development patterns needs to be reconsidered*, European Journal of Agronomy, vol. 28(3), p. 266-272 (2008a).

Martin P., Clouvel P., Luquet D., Dauzat J., *Architectural and geometrical representations of cotton plants to simulate their light interception at low density*, In: Fourcaud T., Zhang X.P. (eds.), Plant growth modelling and applications - Proceedings of PMA06 Los Alamitos, California: IEEE Computer Society, p. 116-123, 2007b.

Dauzat J., Clouvel P., Luquet D., **Martin P.**, *Using virtual plants for analysing the light foraging efficiency of cotton crop at low density*, Annals of Botany, vol. 101(8), p. 1153-1166 (2008).

Clouvel P., Bonvarlet L., Martinez A., Lagouarde P., Dieng I., **Martin P.**, *Wine contamination by Ochratoxin A in relation to vine environment*, International Journal of Food Microbiology, vol. 123(1), p. 74-80 (2008b).

1.4 Autres travaux (dans revues internationales à comité de lecture)

Ndour A., Gueye M., Pereau G., **Martin P.**, Clouvel P., *Développement d'un outil de simulation de la croissance du cotonnier ouvert à l'expertise de l'utilisateur*, Cahiers Agriculture, vol. 15(1), p. 85-91 (2006).

Nibouche S., Guerard N., **Martin P.**, Vaissayre M., *Modelling the role of refuges for sustainable management of dual-gene Bt Cotton in West-African small holder farming systems*, Crop Protection, vol. 26(6), p. 828-836 (2007).

tel-00458265, version 1 - 19 Feb 2010

Rapport-Gratuit.com

tel-00458265, version 1 - 19 Feb 2010

2 Introduction

Baader F., Calvanese D., McGuiness D. L., Nardi D., Patel-Schneider P. F., *The Description Logic Handbook: Theory, Implementation, Applications*, Cambridge University Press, Cambridge, UK, 2003.

Baker D.N., Lambert J.R., McKinion J.M., *GOSSYM: a simulator of cotton crop growth and yield*, S. C. Agricultural Experiment Station Technical Bulletin, No. 1089, S. C. Agricultural Experiment Station, December, 1983.

Bergez J.E., Chabrier P., Garcia F., Gary C., Jeuffroy M.H., Raynal H., Wallach D., *RECORD- Expression des Besoins Fonctionnels*, INRA, Départements EA et MIA, Rapport Projet RECORD, n°2007/1-juin, 2007.

Braudeau E., Mohtar R.H., El Ghezal N., Clouvel P. **Martin P.**, *A multi-scale "soil water structure" model based on the pedostructure concept*, Hydrology and Earth System Sciences Discussions, vol. 6(1), p. 1111-1163 (2009).

Brisson N., Ruget F., Gate P., Lorgeou J., Nicoullaud B., Tayot X., Plenet D., Jeuffroy M.-H., Bouthier A., Ripoche D., Mary B., Justes E., *STICS: a generic model for simulating crops and their water and nitrogen balances. II. Model validation for wheat and maize*, Agronomie, vol. 22, p. 69-92 (2002).

Bouman B., van Keulen H., van Laar H.H., Rabbinge R., *The "school of de Wit" crop growth simulation models: a pedigree and historical overview*, Agricultural systems, vol. 52, p. 171-198 (1996).

Chein M., Mugnier M.L., *Graph-Based Knowledge Representation - Computational Foundations of Conceptual Graphs*, Springer-Verlag, London, 2008.

Courbaud B., Goreaud F., Dreyfus Ph., Bonnet F.R., *Evaluating thinning strategies using a tree distance dependent growth model: some examples based on the CAPSIS software "uneven-aged spruce forests" module*, Forest Ecology and Management, vol. 145, p. 15-28 (2001).

Ferber J., *Les systèmes multi-agents. Vers une intelligence collective*, InterEditions, Paris, 1995.

Gregersen J.P., Gijsbers P.J.A., Westen, S.J.P., *OpenMI: Open modelling interface,* Journal of Hydroinformatics, vol. 9(3), p. 175-191 (2007).

Hendrix G., *Encoding knowledge in partitioned networks*, in associative networks: representation and use of knowledge by computers, Findler, Academic Press, New York, p. 51-92, 1979.

Hillyer C., Bolte J., van Evert F., Lamaker A., *The ModCom modular simulation system*, European Journal of Agronomy, vol. 18(3), p. 333-343 (2003).

IBSNAT, *The IBSNAT (International Benchmark Sites Network for Agrotechnology Transfer) Decade*, Final Report, University of Hawaii, Department of Agronomy and Soil Science, College of Tropical Agriculture and Human Resources, 1993.

tel-00458265, version 1 - 19 Feb 2010

Jones J.W., Hoogenboom G., Porter C.H., Boote K.J., Batchelor W.D., Hunt L.A., Wilkens P.W., Singh U., Gijsman A.J., Ritchie J.T., *The DSSAT cropping system model*, European Journal of Agronomy, vol. 18, p. 235-265 (2003).

Kayser D., *La représentation des connaissances*, Hermès, Paris, 1997.

Martin P., Nibouche S., Clouvel P., *Simulation of bollworm damages on Cotton*. European Network for the Durable Exploitation of Crop Protection Strategies (ENDURE), Workshop on DSS for crop protection, Flakkebjerg, DK, March 17-19, 2008b.

Muetzelfeldt R., Massheder J., *The Simile visual modelling environment*, European Journal of Agronomy, vol. 18 (3-4), p. 345-358 (2003).

Pradal C., Dufour-Kowalski S., Boudon F., Fournier C., Godin C., *OpenAlea: a visual programming and component-based software platform for plant modelling*, Functional plant biology, vol. 35 (9-10), p. 751-760 (2008).

Pressman R.S., *Software Engineering: A Practitioner's Approach, Third Edition*, McGraw-Hill (4ème ed. 1996, 5ème ed. 2001, 6ème ed. 2004, 7ème ed. 2009), New-York, 1992.

Quesnel G., Duboz R., Ramat E., *The Virtual Laboratory Environment – An operational framework for multi-modelling, simulation and analysis of complex dynamical systems*, Simulation Modelling Practice and Theory, vol. 17(4), p. 641-653 (2009).

Ramat E., Preux P., *Virtual laboratory environment (VLE): a software environment oriented agent and object for modeling and simulation of complex systems*, Simulation Modelling Practice and Theory, vol. 11, p. 45-55 (2003).

Sabah G., *L'intelligence Artificielle et le langage - Représentations des connaissances*, Hermès, Paris, 1988.

Schreiber G., *Knowledge engineering*, In: van Harmelen F., Lifschitz V., Porter B. (eds), Handbook of Knowledge Representation, Elsevier, book chapter, p. 929-946, 2008.

Smith M.K., Welty C., McGuinness D.L., *OWL Web Ontology Language Guide. W3C,* http://www.w3.org/TR/owl-guide/, dernière consultation 2009-10-22, 2004.

Sowa J. F., *Conceptual Structures : Information Processing in Mind and Machine*, Addison-Wesley, 1984.

Tuot, T., Le Grenelle Environnement – Rapport Général, Ministère de l'Écologie, de l'Energie, du Développement durable et de l'Aménagement du territoire, http://www.legrenelle-environnement.fr/grenelle-environnement/IMG/pdf/rapporteur_general.pdf, 2007.

Tixier P., Malézieux E., Dorel M., *SIMBA-POP: a cohort population model for long-term simulation of banana crop harvest*, Ecological Modelling, vol. 180, p. 407-417 (2004).

van Ittersum M.K., Ewert F., Heckelei T., Wery J., Alkan Olsson J., Andersen E., Bezlepkina I., Brouwer F., Donatelli M., Flichman G., Olsson L., Rizzoli A.E., van der Wal T., Wien J.E., Wolf J., *Integrated assessment of agricultural systems – A component-*

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

based framework for the European Union (SEAMLESS), Agricultural Systems, vol. 96, p. 150-165 (2008).

3 Chapitre A

Amblard F., Phan D., *Modélisation et simulation multi-agents : applications pour les Sciences de l'Homme et de la Société*, Lavoisier, Paris, 2006.

Athanasiadis I. N., Rizzoli A.E., Donatelli M., Carlini L. *Enriching software model interfaces using ontology-based tools*, In : Proceedings of the iEMSs Third Biannual Meeting "Summit on Environmental Modelling and Software", International Environmental Modelling and Software Society, Burlington, USA (2006).

Bergez J.E., Chabrier P., Garcia F., Gary C., Jeuffroy M.H., Raynal H., Wallach D., *RECORD- Expression des Besoins Fonctionnels*, INRA, Départements EA et MIA, Rapport Projet RECORD, n°2007/1-juin 2007, 2007.

Boote K.J., Jones J.W., Mishoe J.W., Wilkerson G.G., *Modeling growth and yield of groundnut. Agrometeorology of Groundnut*, Proceedings of an International Symposium, ICRISAT Sahelian Center, Niamey, p. 243-254 (1986).

Braudeau E., Mohtar R.H., El Ghezal N., Clouvel P. **Martin P.**, *A multi-scale "soil water structure" model based on the pedostructure concept*, Hydrolology and Earth System Science Discussion, vol. 6(1), p. 1111-1163 (2009).

Brisson N., Ruget F., Gate P., Lorgeou J., Nicoullaud B., Tayot X., Plenet D., Jeuffroy M.-H., Bouthier A., Ripoche D., Mary B., Justes E., *STICS: a generic model for simulating crops and their water and nitrogen balances. II. Model validation for wheat and maize*, Agronomie, vol. 22, p. 69-92 (2002).

Brugali D., Menga G., Aarsten, A., *The framework life span*, Communications of the ACM, vol. 40(10), p. 65-68 (1997).

Campbell R.H., Islam N., Johnson R., Kougiouris P., Madany P., *Choices, Frameworks and Refinement*, In: Cabrera L.F., Russo V., Shapiro M. (eds), Object-Orientation in Operating Systems, IEEE Computer Society Press. Palo Alto, CA, p. 9-15, 1991.

Clouvel P., Lecoeur J., Gérardeaux E., Luquet D., Rey H., **Martin P.**, *With sub-optimum conditions the current representation of potential crop development patterns needs to be reconsidered*, European Journal of Agronomy, vol. 28(3), p. 266-272 (2008a).

Dauzat J., Clouvel P., Luquet D., **Martin P.**, *Using virtual plants for analysing the light foraging efficiency of cotton crop at low density*, Annals of Botany, vol. 101(8), p. 1153-1166 (2008).

Détrie C., Siblot P., Vérine B., *Termes et concepts pour l'analyse du discours - Une approche praxématique*, Honoré Champion, Paris, 2001.

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

Donatelli M., Bellocchi G., Carlini L., Colauzzi M. 2005. *Work Package 3 – Task 3.2. Agricultural Production and Externalities Simulator (APES): the simulation system*. Presentation at the SEAMLESS WP3-WP5 meeting, June 14-17, Milan, Italy, 2005.

Donatelli, M., Acutis M., Adam M., **et al.**, *SEAMLESS APES - current status*, III CGMS Experts Meeting and GEOLAND Training Workshop, Arlon, Belgium, 23-25 October 2006.

Donatelli M., Russell G., Rizzoli A.E., **et al.**, *APES: The Agricultural Production and Externalities Simulator*, In: Brouwer F.M., van Ittersum M.K. (Eds.), Environmental and agricultural modelling: integrated approaches for policy impact assessment, Springer Academic Publishing, book chapter, (sous presse), 2010.

EC, *Règlement (CE) n°11/2008 de la Commission du 8 janvier 2008 portant application du règlement (CE) n°1059/2003 du Parlement européen et du Conseil relatif à l'établissement d'une nomenclature commune des unités territoriales statistiques (NUTS) et concernant les séries chronologiques à transmettre pour le nouveau découpage régional*, Journal officiel de l'Union européenne, 2008.

Ewert F., van Ittersum M.K., Bezlepkina I., Therond O., Andersen E., Belhouchette H., Bockstaller C., Brouwer F., Heckelei T., Janssen S., Jonsson B., Knapen M.J.R., Kuiper M., Louhichi K., Alkan Olsson J., Ruinelli L., Rizzoli A.E., Turpin N., Wery J., Wien J.J.F., Wolf J., *A methodology for enhanced flexibility of integrated assessment in agriculture,* Environmental Science & Policy, vol. 12(5), p. 546-561 (2009).

Fayad M. E., Schmidt D. C., *Object-oriented application frameworks*, Communications of the ACM, vol. 40(10), p. 32-38 (1997).

Gamma E., Helm R., Johnson R., Vlissides J., *Design patterns – Elements of Reusable Object-Oriented Software (1st edition)*, Addison Westley Professional, 2001.

Gijsbers P.J.A., Brinkman R., Gregersen J.B., Hummel S., Westen S.J.P., et al., *The OpenMI Document Series, Part C - the org.OpenMI.Standard interface specification*. http://www.openmi.org/reloaded/about/publications-documents.php, Accès mai 2009, 2005.

Gijsman A.J., Hoogenboom G., Parton W.J., Kerridge P.C., *Modifying DSSAT Crop Models for Low-Input Agricultural Systems Using a Soil Organic Matter–Residue Module from CENTURY*, Agronomy Journal, vol. 94, p. 462-474 (2002).

Gregersen J.P., Gijsbers P.J.A., Westen S.J.P., *OpenMI: Open modelling interface*, Journal of Hydroinformatics, vol. 9(3), p. 175-191 (2007).

Grenievski H., *Le concept d'information et la planification*, In : Cahiers de Royaumont - Le concept d'information dans la science contemporaine, Editions de minuit, Paris, p. 231-240, 1965.

Grüber T.R., *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, vol. 5(2), p. 199-220 (1993).

tel-00458265, version 1 - 19 Feb 2010

Hertel T.W. (Ed.), *Global Trade Analysis: Modeling and Applications*, Cambridge University Press, Cambridge, 1997.

Hillyer C., Bolte J., van Evert F., Lamaker A., *The ModCom modular simulation system*, European Journal of Agronomy, vol. 18(3), p. 333-343 (2003).

Hoogenboom G., Jones J.W., Porter C.H., Wilkens P.W., Boote K.J., Batchelor W.D., Hunt L.A., Tsuji G.Y., *Decision Support System for Agrotechnology Tranfer Version 4.0. Volume 1: Overview*, University of Hawaii, Honolulu, Hawaï, 2003.

Hunt L.A., Boote K.J., *Data for model operation, calibration, and evaluation*, In: Tsuji G.Y., Hoogenboom G., Thornton P.K. (Eds.), Understanding Options for Agricultural Production. Kluwer Academic Publishers, p. 9-39, 1998.

IBSNAT, *The IBSNAT (International Benchmark Sites Network for Agrotechnology Transfer) Decade*, Department of Agronomy and Soil Science, College of Tropical Agriculture and Human Resources, University of Hawaii, Honolulu, Hawaii, 1993.

Janssen S., Andersen E., Athanasiadis I.N., van Ittersum M.K., *A database for integrated assessment of European agricultural systems*, Environmental Science & Policy, vol. 12(5), p. 573-587 (2009).

Jones C.A., Kiniry J.R., *CERES-Maize: A Simulation Model of Maize Growth and Development*, Texas A&M University Press, College Station, Texas, 1986.

Jones J.W., Hunt L.A., Hoogenboom G., Godwin D.C., Singh U., Tsuji G.Y., Pickering N.B., Thornton P.K., Bowen W.T., Boote K.J., Ritchie J.T., *Input and output files*, In: Tsuji, G.Y., Uehara, G., Balas, S. (Eds.), Decision Support System for Agrotechnology Transfer (DSSAT) Version 3, vol. 2. University of Hawaii, Honolulu, HI, p. 1-94, 1994.

Jones J.W., Tsuji G.Y., Hoogenboom G., Hunt L.A., Thornton P.K., Wilkens P.W., Imamura D.T., Bowen W.T., Singh U., *Decision support system for agrotechnology transfer - DSSAT v3*, In: Tsuji, G.Y., Hoogenboom, G., Thornton, P.K. (Eds.), Understanding Options for Agricultural Production. Kluwer Academic Publishers, p. 157-177, 1998.

Jones J.W., Keating B.A., Porter C.H., *Approaches to modular model development*, Agricultural Systems, vol. 70, p. 421-443 (2001).

Jones J.W., Hoogenboom G., Porter C.H., Boote K.J., Batchelor W.D., Hunt L.A., Wilkens P.W., Singh U., Gijsman A.J., Ritchie J.T., *The DSSAT cropping system model*, European Journal of Agronomy, vol. 18, p. 235-265 (2003).

Junker F., Wieck C., Jansson T., Perez I., *Policy description in the CAPRI model*. Working paper, University of Bonn, 2003.

Kim Y.J., Kim T.G., *A heterogeneous simulation framework based on the DEVS BUS and the high level architecture*, In: 1998 Winter Simulation Conference, Washington, USA, p. 421-428 (1998).

Klir G.J., Valach M., *Cybernetic modeling*, (trad. anglaise 1967), SNTL, Prague, 1965.

tel-00458265, version 1 - 19 Feb 2010

Le Moigne J.L., *La théorie du système général*, Presses Universitaires de France, Paris, 1977.

Li H., Louhichi K., Janssen S., Rizzoli A.E., Athanasiadis I., Meuter E., Huberd D., *Development and Application of a Component-based Generic Farm System Simulator Implementing a Semantically Enriched Integrated Modelling Framework*. In: Seventh International Symposium on Environmental Software Systems (ISESS05), May 2007, Prague, Czech Republic (2007).

Martin P., Mohtar R.H., Clouvel P., Braudeau E., *Modeling Soil-Water Dynamics for Diverse Environmental Needs*, In: Voinov A., Jakeman A., Rizzoli A. (eds.), iEMSs Third Biennial Meeting: Summit on Environmental Modelling and Software, International Environmental Modelling and Software Society, July 9-12 2006, Burlington, Vermont, USA, pp. 6 (2007a).

Martin P., Clouvel P., Luquet D., Dauzat J., *Architectural and geometrical representations of cotton plants to simulate their light interception at low density*, In: Fourcaud T., Zhang X.P. (eds.), Plant growth modelling and applications - Proceedings of PMA06 Los Alamitos, California: IEEE Computer Society, 116-123, 2007b.

Martin P., Libourel T., Clouvel P., Reitz P., *Migration revisited using the Category Theory: Application to Biological Modelling Languages*, Rapport de recherche LIRMM [lirmm-00388313 - version 1], pp. 15. (2008a).

Martin P., « *Soil water 2 – soil model » component*, logiciel libre, http://www.apesimulator.it/, 2009a.

Martin P., *« soil water 2 – parameter estimator » component*, logiciel libre, http://www.apesimulator.it/, 2009b.

McNair B.W., Koo J., Walen V.K., Jones J.W., Hoogenboom G., *A web-based data exchange system for crop model applications*, Agronomy Journal, vol. 96, p. 853-856 (2004).

Mens T., Van Gorp P., *A Taxonomy of Model Transformation*, Electronic Notes in Theoretical Computer Science, vol.152, p. 125-142 (2006).

Mohtar R., Braudeau E., Abou Najm M., Salahat M., **Martin P.**, *Kamel*., European Geosciences Union – General Assembly, poster session: Scale, Scaling, and Nonlinearity in the Earth's Surface, Soil and Solid Earth, Vienna, Austria, 2008, Geophysical research abstracts, vol. 10, EGU2008-A-10791 (2008).

Moore R.V., Tindall C.I.., *An overview of the open modelling interface and environment (the OpenMI)*, Environmental Science & Policy, vol. 8, 279-286 (2005).

Muetzelfeldt R., Massheder J., *The Simile visual modelling environment*, European Journal of Agronomy, vol. 18 (3-4), p. 345-358 (2003).

Murray L., Carrington D., Strooper P., *An approach to specifying software frameworks*, In: ACSC '04: Proceedings of the 27th Australasian conference on Computer science, vol. 26, p. 185-192 (2004).

NISO, *Understanding Metadata*, NISO Press (National Information Standards Organization), 2004.

tel-00458265, version 1 - 19 Feb 2010

Pérez Domínguez I, Bezlepkina I, Heckelei T, Oude Lansink A, Romstad E and Kannellopoulos A., *EXPAMOD: A methodological Tool for Linking Farm and Market Models by Means of Econometric Response Functions*, Paper presented at the 12th Congress of the European Association of Agricultural Economists, August 2008, Ghent (2008).

Porter C., Jones J.W., Braga R., *An approach for modular crop model development*, International Consortium for Agricultural Systems Applications, Honolulu, Hawaï, http://www.icasanet.org/modular/index.html, 2000.

Pressman R.S., *Software Engineering: A Practitioner's Approach, Third Edition*, McGraw-Hill (4ème ed. 1996, 5ème ed. 2001, 6ème ed. 2004, 7ème ed. 2009), New-York, 1992.

Quesnel G., Duboz R., Ramat E., Traoré M.K. *VLE – A Multimodeling and Simulation Environment*, In: Proceedings of the Summer Computer Simulation Conference, ``Moving Towards the Unified Simulation Approach" (SCS - ACM), July 15-18, 2007, San Diego, USA (2007).

Quesnel G., Duboz R., Ramat E., *The Virtual Laboratory Environment – An operational framework for multi-modelling, simulation and analysis of complex dynamical systems*, Simulation Modelling Practice and Theory, vol. 17(4), p. 641-653 (2009).

Ramat E., Preux P., *Virtual laboratory environment (VLE): a software environment oriented agent and object for modeling and simulation of complex systems*, Simulation Modelling Practice and Theory, vol. 11, p. 45-55 (2003).

Reynolds J.F., Acock B., *Modularity and genericness in plant and ecosystem models*, Ecological Modeling 94, p. 7-16 (1997).

Ritchie J.T., Otter S., *Description and performance of CERES-Wheat: a user-oriented wheat yield model*, In: National Technical Information Service, ARS Wheat Yield Project, Springfield, Missouri, p. 159-175, 1985.

Sabah G., *L'intelligence Artificielle et le langage - Représentations des connaissances*, Hermès, Paris, 1988.

Rizzoli A.E., Donatelli M., Athanasiadis I.N., Villa F., Huber D., *Semantic links in integrated modelling frameworks*, Mathematics and Computers in Simulation, vol. 78, p. 412-423 (2008).

Tsuji G.Y., Uehara G., Balas S. (Eds.) *Decision Support System for Agrotechnology Transfer (DSSAT) Version 3,* University of Hawaii, Honolulu (Hawaii), 1994.

Van Evert F.K., Leffelaar P.A., Acutis M., Adam M., Ewert F., va, Keulen H., Trevisiol P. *Practical Use of Components in Agro-Ecological Simulation*, In: 2007 Summer Computer Simulation Conference, San Diego, California (USA) (2007).

van Ittersum M.K., Ewert F., Alkan Olsson J., et al., *Integrated assessment of agricultural and environmental policies – towards a computerized framework for the EU (SEAMLESS-IF)*, In: Voinov, A., Jakeman, A., Rizzoli, A. (eds). Proceedings of the iEMSs Third Biannual Meeting “Summit on Environmental Modelling and Software”.

tel-00458265, version 1 - 19 Feb 2010

International Environmental Modelling and Software Society, Burlington, USA, 9-12 July (2006).

van Ittersum M.K., Frank E., Thomas H., Jacques W., Alkan Olsson J., Andersen E., Bezlepkina I., Brouwer F., Donatelli M., Flichman G., Olsson L., Rizzoli A.E., van der Wal T., Wien J.E., Wolf J., *Integrated assessment of agricultural systems – A component-based framework for the European Union (SEAMLESS)*, Agricultural Systems, vol. 96, p. 150-165 (2008).

Verweij P.J.F.M., Knapen M.J.R., Wien J.J.F., *The use of OpenMI in model based integrated assessments*, In: MODSIM 2007 International Congress on Modelling and Simulation, Modelling and Simulation Society of Australia and New Zealand, December 2007, p. 1166-1171 (2007).

von Bertalanffy L., *La théorie générale des systèmes*, Dunod, Paris (1ère édition 1973), 2002.

Ward M.P., Zedan H., *Slicing as a program transformation*, ACM Transactions on Programming Languages and Systems, vol. 29(2), p. 1-52 (2007).

Wilkerson G.G., Jones J.W., Boote K.J., Ingram K.T., Mishoe J.W., *Modeling soybean growth for crop management*, Transactions of the ASAE, vol. 26, p. 63-73 (1983).

Zeigler B.P., *Towards a Formal Theory of Modeling and Simulation: Structure Preserving Morphisms*, Journal of the Association for Computing Machinery, Vol. 19(4), p.742-764, (1972).

Zeigler B.P., Praehofer H., Kim T.G. *Theory of Modeling and Simulation*, Academic Press (2nd ed.), 2000.

4 Chapitre B

Armengaud F., *La Pragmatique*, Presses Universitaires de France, Paris, (5ème éd.), collection Que sais-je, 2007.

Benveniste E., *Problèmes de linguistique générale - 1*, Gallimard, 1966.

Brézillon P., *Représentation des procédures et pratiques par les graphes contextuels*, In : J.M.C. Bastien (Ed.), Actes des Deuxièmes Journées d'Étude en Psychologie Ergonomique - EPIQUE'2003(Boulogne-Billancourt, 2-3 octobre), p. 3-14, Rocquencourt : Inria, 2003.

Charlot B., *Du rapport au savoir - Eléments pour une théorie*, Economica, Paris, 1997.

Chein M., Mugnier M.L., *Graph-Based Knowledge Representation - Computational Foundations of Conceptual Graphs*, Springer-Verlag, London, 2008.

Chomsky N., *Remarks on Nominalizations*, In : Jacobs R., Rosenbaum P. (eds), Readings in English Transformational Grammar, Waltham (Mass.), p. 184-221, 1970.

Declerck R., *When-clauses and temporal structures*, Routledge, London, 1997.

tel-00458265, version 1 - 19 Feb 2010

Détrie C., Siblot P., Vérine B., *Termes et concepts pour l'analyse du discours - Une approche praxématique*, Honoré Champion, Paris, 2001.

Dubois J., Dubois-Charlier F., *Synonymie syntaxique et classification des verbes français*, Langages, vol. 31, n°128, p. 51-71 (1997).

El Hasnaoui Asmahane Rami, *Le circonstant de manière*, Thèse de doctorat, Université Paris VI – Sorbonne, Ecole doctorale V Concepts et langages, 2008.

Fillmore C.J., *Lectures on Deixis*, Center for the Study of Language and Information, 1997.

Francois J., *Changement, causation, action : trois catégories sémantiques fondamentales du lexique verbal français et allemand*, Librairie Droz, Genève-Paris, 1989.

François J., Le Pesant D., Leeman D., *Présentation de la classification des Verbes Français de Jean Dubois et Françoise Dubois-Charlier*, Langue Française, vol. 153(1), p. 3-19 (2007).

Gosselin L., *Les circonstanciels : de la phrase au texte*, Langue Française, vol. 86(1), p. 37-45 (1990).

Gougenheim G., *Les pronoms interrogatifs que et quoi*, Le Français Moderne, Tome 17, p. 85-90 (1949).

Gougenheim G., *Animé et inanimé – A propos de qui interrogatif et qui relatif prépositionnel*, Le Français Moderne, Tome 18, p. 6-16 (1950).

Grenier J.Y., Grignon C., Menger P.M., *Le modèle et le récit*, Ed. de la Maison des sciences de l'homme, Paris, 2001.

Gregersen J.P., Gijsbers P.J.A., Westen, S.J.P., *OpenMI: Open modelling interface,* Journal of Hydroinformatics, vol. 9(3), p. 175-191 (2007).

Guillaume G., *Leçons de linguistiques 1948-1949 (série A, B et C)*, Presses de l'Université Laval, 1973.

Guillaume G., *Leçons de linguistiques 1949-1950 (série A)*, Presses de l'Université Laval, 1974.

Hadermann P., *Etude morphosyntaxique du mot où*, Edition Duculot, Paris-Louvain-la-neuve, 1993.

Hagège C., *La structure des langues*, Paris, Presses Universitaires de France, 6e ed (1re ed. 1982), 2001.

Hendrix G., *Encoding knowledge in partitioned networks*, in associative networks: representation and use of knowledge by computers, Findler, Academic Press, New York, p. 51-92, 1979.

Hobbs J.R., Pan F., *Time Ontology in OWL - Technical report*, Ontology Engineering Patterns Task Force of the Semantic Web Best Practices and Deployment Working Group,

World Wide Web Consortium (W3C), http://www.w3.org/TR/2006/WD-owl-time-20060927/, 2006.

Ilinski K., *Les degrés de l'incidence*, Le Français Moderne, vol. 71(1), p. 52-67 (2003).

Jacquesson F., *Les Personnes : Morphosyntaxe et sémantique*, CNRS, Coll. Sciences du langage, 2008.

Kayser D., *La représentation des connaissances*, Hermès, Paris, 1997.

Kerbrat-Orecchioni C., *La question*, Presses Universitaires de Lyon, 1991.

Kerbrat-Orecchioni C., *Les actes de langage dans le discours*, Nathan, Paris, 2001.

Korzen H., *Pourquoi et l'inversion finale en français : étude sur le statut de l'adverbial de cause et l'anatomie de construction tripartite*, La revue romane, Munksgaard-Danemark, 1985.

Larousse, *Le petit Larousse Grand Format*, 2003.

Lazard G., *L'actance*, Paris, Presses Universitaires de France, 1994.

Leeman-Bouix, D., *Des topoï à la théorie des stéréotypes en passant par la polyphonie et l'argumentation dans la langue : hommages à Jean-Claude Anscombre*, Université de Savoie, Chambery, 2009.

Lenat D., *The dimensions of context space*, Technical report, CYCORP. URL:http://www.cyc.com/doc/context-space.pdf, 28 October 1998.

Maingueneau D., *L'énonciation en linguistique française*, Hachette – les fondamentaux, 1994.

Martin R. *L'interrogation comme universel du langage*, In: Linguistica Palatina Colloquia II - l'interrogation, actes du colloque tenu les 19 et 20 décembre 1983, Presses de l'Université de Paris-Sorbonne, Paris, p. 257-284, 1985.

Martin R., *Comprendre la linguistique*, Presses Universitaires de France, Paris, 2002.

Mascherin L., *Analyse morphosémantique de l'aspectuo-temporalité en Français. Le cas du préfixe RE-*, Mémoire de Doctorat, Université Nancy II, 2008.

Mélis L., *Les circonstants et la phrase*, Presses Universitaires de Louvain, 1983.

Mondy N., Corio-Costet M.F., *The response of the grape berry moth (Lobesia botrana) to a dietary phytopathogenic fungus (Botrytis cinerea): the significance of fungus sterols*, Journal of Insect Physiology, vol. 46, p. 1557-1564 (2000).

Recanati F., *Le sens littéral : langage, contexte, contenu*, Edition de l'éclat, Paris, 2007.

Sabah G., *L'intelligence Artificielle et le langage - Représentations des connaissances*, Hermès, Paris, 1988.

tel-00458265, version 1 - 19 Feb 2010

Rapport-Gratuit.com

tel-00458265, version 1 - 19 Feb 2010

Sagot B., Fiser D., *Building a free French wordnet from multilingual resources*, In: Actes de Ontolex 2008, Marrakech, Maroc, (2008).

Sarfati G.E., *Précis de pragmatique*, Armand Colin, 2002.

Schreiber G., *Knowledge engineering*, In: van Harmelen F., Lifschitz V., Porter B. (eds), Handbook of Knowledge Representation, Elsevier, book chapter, p. 929-946, 2008.

Sénéchal M., Willems D., *Classes verbales et régularités polysémiques : le cas des verbes trivalenciels locatifs*, Langue Française, vol. 153, p. 92-110 (2007).

Sowa J. F., *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, 1984.

Spinoza B. de, *L'éthique*, Folio, 2001.

Stalnaker R.C., *Pragmatics*, Synthese, vol 22, p. 272-89 (1970).

Stalnaker R., *On the Representation of Context*, Journal of Logic, Language, and Information, vol. 7, p. 3–19 (1998).

Tesnière L., *Eléments de syntaxe structurale*, Ed. Klincksieck, Paris, 1988.

TLFi, *Trésor de la Langue Française informatisé* - 1971-1994, http://atilf.atilf.fr/, 2009.

Touratier, C., *Adverbe et Circonstant*, Publications de l'Université de Provence, 2001.

Vandeloise C., *Au-delà des descriptions géométriques et logiques de l'espace : une description fonctionnelle*, Linguisticae Investigationes, vol. IX(1), p. 103-129 (1985).

5 Chapitre C et Conclusion

Amblard F., Phan D., *Modélisation et simulation multi-agents : applications pour les Sciences de l'Homme et de la Société*, Lavoisier, Paris, 2006.

Badiou A., *Logiques des mondes – l'être et l'évènement 2*, Seuil, paris, 2006.

Clouvel P., Bonvarlet L., Martinez A., Lagouarde P., Dieng I., **Martin P.**, *Wine contamination by Ochratoxin A in relation to vine environment*, International Journal of Food Microbiology, vol. 123(1), p. 74-80 (2008b).

Ducrot O., Schaeffer J.M., *Nouveau dictionnaire encyclopédique des sciences du langage*. Editions du Seuil, Paris, 1999.

Mondy N., Corio-Costet M.F., *The response of the grape berry moth (Lobesia botrana) to a dietary phytopathogenic fungus (Botrytis cinerea): the significance of fungus sterols*, Journal of Insect Physiology, vol. 46, p. 1557-1564 (2000).

Sabah G., L'intelligence Artificielle et le langage - Représentations des connaissances, Hermès, Paris, 1988.

tel-00458265, version 1 - 19 Feb 2010

Annexe

Migration revisited using the Category Theory: Application to Biological Modelling Languages

Pierre Martin[1,2], Therese Libourel[2], Pascal Clouvel[3], and Philippe Reitz[2]

[1] CIRAD - PERSYST/DIR – Avenue Agropolis, 34398 Montpellier Cedex 5, France
Pierre.martin@cirad.fr

[2] LIRMM, CNRS and Université de Montpellier 2, 161, rue Ada,
34392 Montpellier cedex 5, France
{libourel, reitz}@lirmm.fr

[3] CIRAD – UPR 102 - Avenue Agropolis, 34398 Montpellier Cedex 5, France
Pascal.clouvel@cirad.fr

Abstract. Since the early seventies, numerous numerical programs have been developed to simulate biophysical processes. The current challenge facing scientific communities is to access any program using their usual language. Migration is a way of dealing with the challenge.
The conventional methodology to build up migration functions is the rule-based method. An alternative consists in using the category theory. This paper illustrates the methodology applied to bidirectional migration of a program written using a language based on the system theory (SIMILE) to a biological modelling language (APES - EU SEAMLESS integrated project) constructed using design patterns. Using the category theory enabled us (i) to provide a mathematical formalism for composition of the program features, i.e. the architecture and semantic function, (ii) to identify aspects which could not be preserved in the migration because of language expressiveness, and (iii) to automate the migration.

Keywords: Preservation, Architecture, Semantic Function, UML, Program Equivalence.

1 Introduction

Since the early seventies, numerous numerical programs have been developed to simulate biophysical processes. Most of those programs were first written using common programming languages, e.g. Fortran, and source code structure specific to each researcher [1]. Later on, in the 90s, a family of languages based on the system theory appeared [2]. Within that range, Modelica [3], Stella [4] and SIMILE [5], for instance, started being used as an assistant framework for program construction. Those languages used the concepts of object, relation between objects, object aggregation and model abstraction level M1. Recently, the new concept of design patterns [6] and ADL [7] started to be used in biological programs [8]. While languages based on the system theory and on the new concepts are simultaneously being used, the biological community today is faced with the problem of re-using the different programs. The current challenge facing the community is to access any program using their

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

usual language, and to be able to integrate program improvements whatever the language. Bidirectional migration [9, 10] is a way of dealing with the challenge.
Rule-based transformation systems [11] and triple graph grammars [12] are commonly adopted for migration. Both methods consist in identifying appropriate transformation rules and the rule matching algorithms [13]. In these methods, the transformation rules are incrementally identified and thus not automated. Moreover, preservation is evaluated *in fine*. Alternatively, other methodologies are quoted facing preservation. Concerning architecture, [14] proposed a method using process algebra. On the other hand, [15] adopted the category theory for preserving program maintainability in regard to the semantic function, i.e. the program computation [16]. Out of preservation, the category theory has already been applied to computer science to solve complex theoretical problems [17, 18].
In the case of biological program migration, our objective was to preserve the architecture of the program and the semantic function. Our question was then to identify a methodology enabling bidirectional preservation of both features. The category theory [19] deals with mathematical structures and relationships between them. Graphs are made of objects, and arrows between objects. In addition to objects and arrows (called morphisms in the category theory), the category theory also considers the composition of arrows, i.e. the composition of morphisms [20]. The advantage of the theory is thus to take into account graph structure through the composition of morphisms. In our case, we assumed that languages could be represented by categories. We hypothesize that studying the relationship between categories could enable us to (i) identify aspects which could not be preserved because of the difference in language expressiveness and (ii) construct the appropriate bidirectional migration function. Once the object correspondence established in the scope of the category theory, the elaboration of translation rules becomes automated by construction.
The aim of this paper is to present the methodology and highlight its merits for bidirectional migration. We illustrate our proposal with the bidirectional migration of a program written using SIMILE [5] and a biological modelling language (APES - EU SEAMLESS integrated project) constructed using design patterns [6].

2 Category theory

2.1 Definition and notation

A category C is defined by:
- A class of objects, noted Ob(C).
- For each pair of objects (X, Y), a set noted $Hom_C(X, Y)$, whose elements are called morphisms of X on Y.
- and for each triplet of objects (X, Y, Z) of Ob(C), an application $Hom_C(X, Y) \times Hom_C(Y, Z) \longrightarrow Hom_C(X, Z)$ called composition of morphisms.

A category should verify the following conditions:
- Condition 1: if a pair of objects (X_1, Y_1) is different from the pair (X_2, Y_2), then $Hom_C(X_1, Y_1) \cap Hom_C(X_2, Y_2) = \varnothing$
- Condition 2: for each object $X \in Ob(C)$, there is an element of $Hom_C(X, X)$, noted id_X which is a neutral element for the composition of morphisms: $f \circ id_X = id_Y \circ f = f$
- Condition 3: composition is associative. Let $f: X \longrightarrow Y$, $g: Y \longrightarrow Z$ and $h: Z \longrightarrow T$, we have: $(h \circ g) \circ f = h \circ (g \circ f)$.

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

2.2 Operations on categories

Several operations can be performed on categories (monads, toposes, etc.). Two of them were used in this study: product of categories and functor. In our case, because the class of objects was a finite set and the class of morphisms was a set, the category is said to be small.

Product of small categories. Let Ob(C) = $\{X_C, Y_C \ldots\}$ and Ob(D) = $\{X_D, Y_D \ldots\}$. The product of C and D, noted C × D, provides a new category E defined as follows:

- Ob(E) is the Cartesian product of Ob(C) and Ob(D). An object of E corresponds to a pair of objects of C and D, e.g. (X_C, X_D).
- Hom(E) is the Cartesian product of Hom(C) and Hom(D). A morphism of E $(X_C, Y_C) \longrightarrow (X_D, Y_D)$ is a pair $<f, g>$ of morphisms of C × D where f: $X_C \longrightarrow X_D$ in C and g: $Y_C \longrightarrow Y_D$ in D.
- The composition of morphisms is defined by: $<f_C, g_C> \circ <f_D, g_D> = <f_C \circ f_D, g_C \circ g_D>$
- and morphism identity by: $id_E = <id_C, id_D>$

Functor. A functor corresponds to a morphism of categories. A functor supports the mapping from objects to objects, morphisms to morphisms, and preserves source, target, identities and composition. The covariant functor F (called functor in this paper) mapping category C to category D is defined as follows:

- Objects: for each $X \in Ob(C)$, $F(X) \in Ob(D)$
- Morphisms: for each $f \in Hom_C(X, Y)$, $F(f) \in Hom_D(F(X), F(Y))$

Two conditions have to be verified:

- Condition 4: relative to identity morphism $F(id_X) = id_{F(X)}$
- Condition 5: let f and g 2 morphisms of the source category: $F(g \circ f) = F(g) \circ F(f)$

3 Application to migration

3.1 Categories

A language is defined by its architecture and a semantic function. According to [5], architectural configurations, or topologies, are connected graphs of constituents and connectors that describe architectural structure. Architecture provides the inclusion relationships between constituents and sets of constituents. Architecture can then be represented by a category (C_{Arch}), where constituents are objects and morphisms the inclusion relationships between constituents. In that case, the composition of morphism is given by the association of morphisms e.g. $(A \subset B)$ and $(B \subset C) \longrightarrow (A \subset C)$. In accordance with this definition, conditions 1 and 3 are verified by construction. For condition 2, the identity morphisms correspond to the inclusion relationship mappings of an object with itself, which is true.

The semantic function deals with the numerical relationship existing between constituents. The semantic function can thus be considered as a calling sequence. It can then be represented by a category (C_{SF}) where constituents are objects and morphisms the calling sequence of the constituents. In that case, the composition of morphism is given by the order relationship e.g. $(A \leq B)$ and $(B \leq C) \longrightarrow (A \leq C)$. In accordance with this definition, condition 1 is verified by construction. Since the composition of morphisms corresponds to the composite of sequence, which is also a sequence, condition 3 is also verified. Finally, the identity morphism for condition 2 corresponds to the empty path.

From a mathematical point of view, the product of categories allows to consider simultaneously the characteristics of both categories. In this paper, we deal with languages whose architecture and semantic function are independent. Thus, we considered the language as the product of the architecture and the semantic function according to equation (1).

$$C_{Lang} = C_{Arch} \times C_{SF} \ . \qquad (1)$$

3.2 Migration function

Migration consists in establishing the correspondence between the source and the target language. A language is described using classes to instantiate in order to build the program. By definition, the functor represents the migration function. Object mapping corresponds to mapping the source language classes to the target ones, and morphism to the translation rules. Translation rules consider simultaneously both the architecture and semantic function. Functor F from source to target language, represented respectively by $C_{SourceLang}$ and $C_{TargetLang}$ categories, is noted:

$$F: C_{SourceLang} \longrightarrow C_{TargetLang} \ . \qquad (2)$$

In order to introduce a common description of the language, we regroup the original language classes using UML. The extra classes created are then used for constructing the categories and the functors. By definition, a functor should always map the source to target objects and morphisms. Aspects having no target correspondence cannot be preserved.

4 Illustration

4.1 SIMILE category (C_S)

SIMILE [5, 21] is a modelling language devoted to the simulation of differential dynamic systems [22]. In the user interface, SIMILE proposes a set of classes that need to be instantiated in order to build up the program. Change in entity status in relation to that of the connected ones is specified using numerical equations also written by users. In [5], the authors describe language functionalities, but not the language itself. From functionalities, we deduced the class diagram presented in Figure 1.

tel-00458265, version 1 - 19 Feb 2010

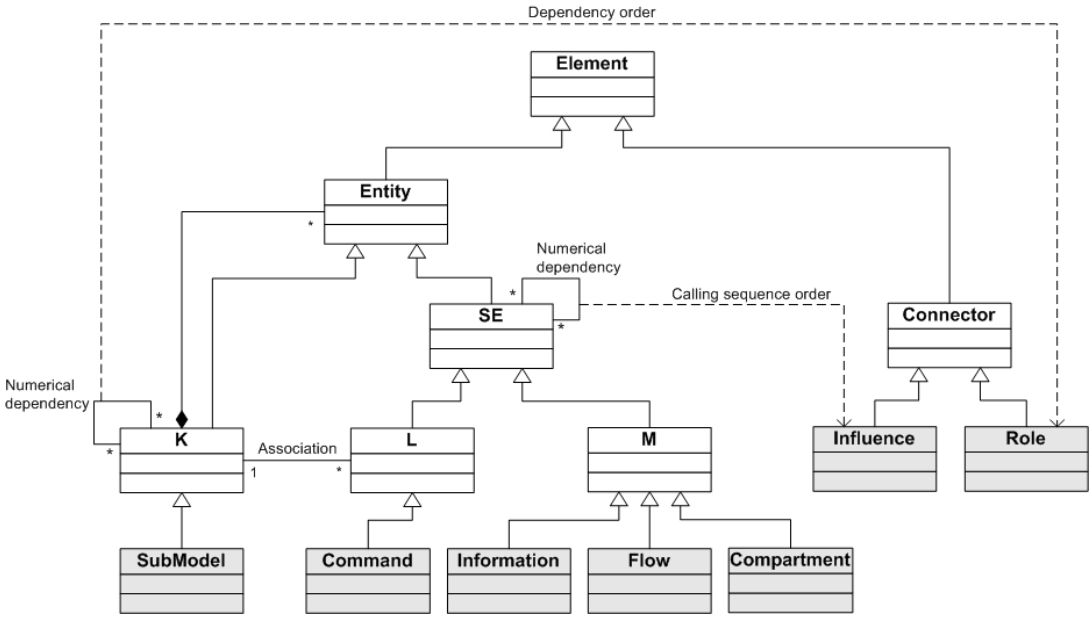


Fig. 1. Class diagram of SIMILE language designed using the UML convention. Classes are represented by rectangles. Classes in grey correspond to those being instantiated by the user. The diagram was inferred from the description of functionalities by [5].

In accordance with UML, the SIMILE language offers 2 types of element class: entity and connector, which correspond respectively to the objects used for the description of the simulated system, and to the characterization of the associative relationship existing between entities. We classified entities depending on their capacity to aggregate the others. Complex entities (K) could aggregate any entity while simple entities (SE) could not. Within the K class, "SubModel" enabled the constitution of groups of entities. Within the SE, we distinguished associative SE (L) which could be associated to K, and basic SE (M) which could not. M was the base class for 3 entity classes, namely:

- "Compartment", representing the state of the system
- "Flow", representing the numerical function responsible for changes in the state of the system
- "Information", representing the information used in the management of the system. Information can be defined either by a parameter, or an input, or an intermediary variable.

L was the base class for the "Command" class associated to K. Commands were orders applied to every entity included in K (K entry condition, loops, etc.).
Two types of oriented connectors are offered. "Influence" is used to establish the numerical relationship between different SE and thus specify the calling sequence. "Role" only applies to K, and deals with the program architecture. It confers the size of the multi-dimensional array of a K target in relation to that of the K sources.
The class of objects of the two feature categories is a set containing the 3 objects of the language: {K, L, M}. Based on the class diagram (Figure 1), we established the morphisms between objects for the two feature categories shown in Figure 2. In the case of the semantic function, morphisms $K \longrightarrow M$ and $M \longrightarrow K$ are not explicit but resulted respectively from the composition of $Hom(K, L) \times Hom(L, M)$ and $Hom(M, L) \times Hom(L, K)$. From a mathematical point of view, the architecture feature graph presents a tree structure where leaves correspond to classes L and M, and internal nodes to class K. The semantic function feature graph presents a lattice structure where all classes are interconnected. These two graphs are summarised in Table 1.

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

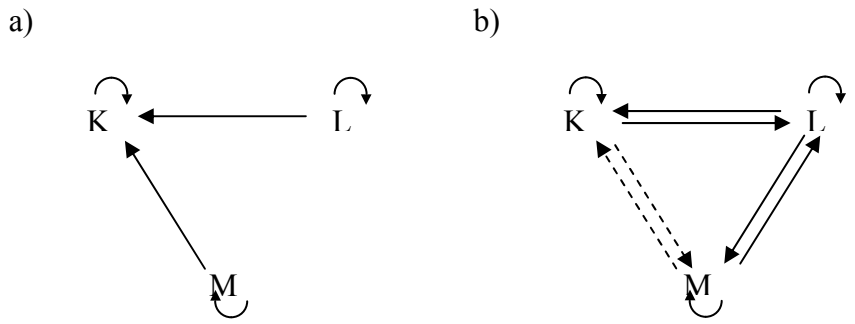


Fig. 2. Graphs of the architecture (a) and semantic function (b) of SIMILE.

Table 1. Morphisms of the architecture and the semantic function categories of SIMILE. A column corresponds to the source and a line to the target. Id_X corresponds to the identity morphism, '*' to the existence and '∅' to the absence of morphism.

Source object	Architecture K	Architecture M	Architecture L	Semantic function K	Semantic function M	Semantic function L
K	id_K	∅	∅	id_K	*	*
M	*	id_M	∅	*	id_M	*
L	*	∅	id_L	*	*	id_L

The language category corresponds to the product of the architecture and the semantic function categories. The resulting class of objects of the SIMILE category is $Ob(C_S)$ = {(K, K), (K, L), (K, M), (L, K), (L, L), (L, M), (M, K), (M, L), (M, M)}. Language category morphisms are obtained by the Cartesian product of architecture and semantic function morphisms (Table 2).

Table 2. Morphisms of the SIMILE language category. The table shows the Cartesian product of morphisms of architecture (column), and semantic function categories (line). The table was drawn up from the definition of the product of categories according to the category theory. Id_X corresponds to the identity morphism, '*' to the existence and '∅' to the absence of morphism.

Hom(x,y)	K,K	K,M	K,L	M,K	M,M	M,L	L,K	L,M	L,L
K,K	id_K,id_K	$id_K,*$	$Id_K,*$	∅,id_K	∅,*	∅,*	∅,id_K	∅,*	∅,*
K,M	$id_K,*$	id_K,id_M	$Id_K,*$	∅,*	∅,id_M	∅,*	∅,*	∅,id_M	∅,*
K,L	$id_K,*$	$id_K,*$	Id_K,id_L	∅,*	∅,*	∅,id_L	∅,*	∅,*	∅,id_L
M,K	*,Id_K	*,*	*,*	id_M,Id_K	$id_M,*$	$id_M,*$	∅,id_K	∅,*	∅,*
M,M	*,*	*,id_M	*,*	$id_M,*$	id_M,id_M	$id_M,*$	∅,*	∅,id_M	∅,*
M,L	*,*	*,*	*,id_L	$id_M,*$	$id_M,*$	id_M,id_L	∅,*	∅,*	∅,id_L
L,K	*,id_K	*,*	*,*	∅,id_K	∅,*	∅,*	id_L,id_K	$id_L,*$	$id_L,*$
L,M	*,*	*,id_M	*,*	∅,*	∅,id_M	∅,*	$id_L,*$	id_L,id_M	$id_L,*$
L,L	*,*	*,*	*,id_L	∅,*	∅,*	∅,id_L	$id_L,*$	$id_L,*$	id_L,id_L

tel-00458265, version 1 - 19 Feb 2010

4.2 APES category (C_A)

APES [23] is a platform devoted to the simulation of agricultural production on a field level. APES was developed under the EU Sixth Framework Research Programme SEAMLESS [24, 25]. APES architecture consists of a set of self-contained components [26] interconnected using the ModCom core [27]. The role of Modcom is (i) to construct a recursive calling chain of the component set and (ii) to transfer output variable values from one component to the others at every time step of the simulation. In APES, the architecture of a component obeys a specific design based on the use of design patterns. The connection of the component with ModCom is based on the "adapter" structural design pattern. Connection consists in adapting the component interface (ClsMyComponent) to that of ModCom (OdeSimObj). OdeSimObj provides four virtual methods which present specific roles in the component running within the components chain. Among them, GetRates is the method dedicated to run the program of the component (Figure 3).

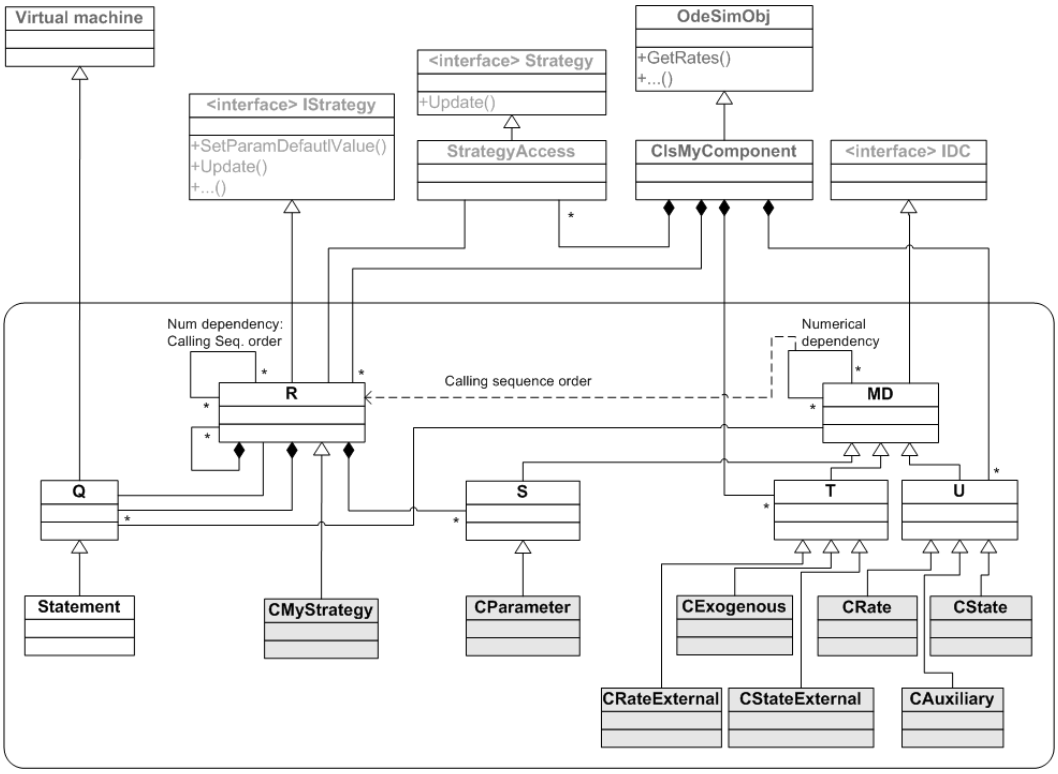


Fig. 3. Class diagram of an APES component designed using the UML convention. Classes in grey correspond to those implementing the component program.

The structure of the component program is based on the object paradigm [28]. In accordance with this paradigm, the state of the program is described using classes of data (Implemented data class = MD) and its behaviour using classes of methods (Implemented method class = R). We established the class diagram of the component in reference to [23] and [29]. The role of data classes is to group variables in accordance with the categorization imposed by APES. A distinction is made between three types of data classes. Basic data (U) correspond to the variables of the program:

- "CState", containing the state variables of the program
- "CRate", containing the variation rate variables of the program
- "CAuxiliary", containing the other variables of the program which cannot be part of the two data classes above.

Input data (T) are provided from the other components of the platform:

- "CRateExternal", containing the input state variables
- "CStateExternal", containing the input variation rate variables
- "CExogeneous", containing the input auxiliary variables

Specific input data (S) relative to a particular method:

- "CParameter", containing the value of the different parameters of the program

In the data classes, the variable types correspond to usual static data types (integer, Boolean, real, string). Arrays of variables are allowed, but not data structures. Classes allow the adjunction of text describing each data item, its initial value and the upper and lower bounds.

The role of the method class is to establish the calling sequence of the methods and to carry out the calculation on the U variables using variables from U, T, and S. The R class is designed using three design patterns [6]: the "strategy" behavioural design pattern consists in defining a unique design of method structure in order to make them interchangeable, the "façade" structural design pattern consists in defining a unified interface, and the "composite" structural design pattern consists in allowing the composite of methods into tree structures to represent part-whole hierarchies of methods. The association of those patterns provides a unique structure to all methods, independently of the program source code.

Finally, we added the "Q" class corresponding to the statement provided by the virtual machine. In APES, the considered statements are "if" and "for".

The class of objects of the two feature categories is a set containing the 5 objects of the language: {R, U, T, S, Q}. Based on the class diagram (Figure 3), we established the morphisms between objects for the two feature categories (Figure 4). The graph of architecture feature shows 3 sub-graphs. Two sub-graphs consist of isolated nodes, i.e. U and T. The third sub-graph consists of a tree graph where leaves correspond to S and Q and the internal node to R. The graph of the semantic function feature shows a lattice structure. These two graphs are summarized in Table 3.

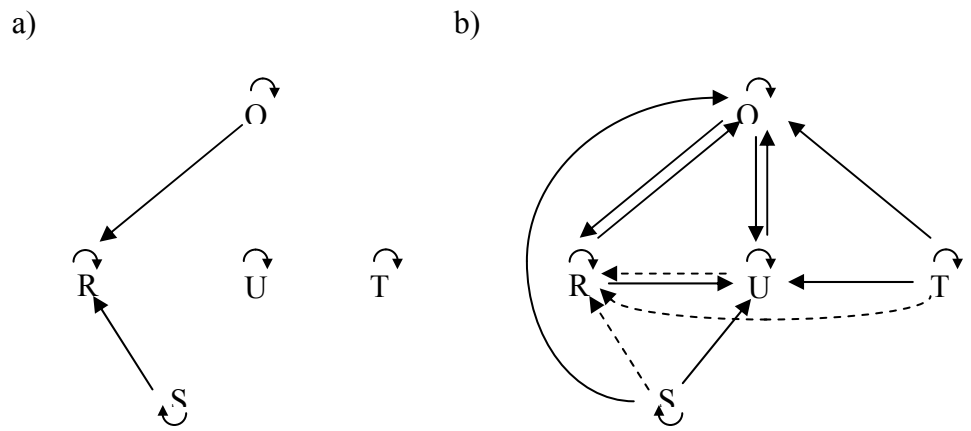


Fig. 4. Graphs of the architecture (a) and semantic function (b) of APES.

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

Table 3. Morphisms of the architecture and the semantic function categories of APES. A column corresponds to the source and a row to the target. Id_X corresponds to the identity morphism, '*' to the existence and '∅' to the absence of morphism.

Source	Architecture					Semantic function				
Object	R	U	T	S	Q	R	U	T	S	Q
R	id_R	∅	∅	∅	∅	id_R	*	∅	∅	*
U	∅	id_U	∅	∅	∅	*	id_U	∅	∅	*
T	∅	∅	id_T	∅	∅	*	*	id_T	∅	*
S	*	∅	∅	id_S	∅	*	*	∅	id_S	*
Q	*	∅	∅	∅	Id_Q	*	*	∅	∅	Id_Q

The resulting class of objects of the APES category is $Ob(C_A)$ = {(R, R}, (R, U), (R, T), (R, S), (R, Q), (U, R), (U, U), (U, T), (U, S), (U, Q), (T, R), (T, U), (T, T), (T, S), (T,Q), (S, R), (S, U), (S, T), (S, S), (S, Q), (Q, R), (Q, U), (Q, T), (Q, S), (Q, Q)}. Language category morphisms are given in Table 4 (Q class not displayed).

Table 4. Morphisms of the APES language category. The table displays the Cartesian product of morphisms of architecture (column), and semantic function categories (row). The table is based on the definition of the product of categories according to the category theory. Id_X corresponds to the identity morphism, '*' to the existence and '∅' to the absence of morphism. To improve legibility, '—' is used in place of '(∅,∅)'.

Hom(x,y)	R,R	R,U	R,T	R,S	U,R	U,U	U,T	U,S	T,R	T,U	T,T	T,S	S,R	S,U	S,T	S,S
R,R	id_R,id_R	id_R,*	id_R,∅	id_R,∅	∅,id_R	∅,*	—	—	∅,id_R	∅,*	—	—	∅,id_R	∅,*	—	—
R,U	id_R,*	id_R,id_U	id_R,∅	id_R,∅	∅,*	∅,id_U	—	—	∅,*	∅,id_U	—	—	∅,*	∅,id_U	—	—
R,T	id_R,*	id_R,*	id_R,id_T	id_R,∅	∅,*	∅,*	∅,id_T	—	∅,*	∅,*	∅,id_T	—	∅,*	∅,*	∅,id_T	—
R,S	id_R,*	id_R,*	id_R,∅	id_R,id_S	∅,*	∅,*	∅,∅	∅,id_S	∅,*	∅,*	—	∅,id_S	∅,*	∅,*	—	∅,id_S
U,R	∅,id_R	∅,*	—	—	id_U,id_R	id_U,*	id_U,∅	id_U,∅	∅,id_R	∅,*	—	—	∅,id_R	∅,*	—	—
U,U	∅,*	∅,id_U	—	—	id_U,*	id_U,id_U	id_U,∅	id_U,∅	∅,*	∅,id_U	—	—	∅,*	∅,id_U	—	—
U,T	∅,*	∅,*	∅,id_T	—	id_U,*	id_U,*	id_U,id_T	id_U,∅	∅,*	∅,*	∅,id_T	—	∅,*	∅,*	∅,id_T	—
U,S	∅,*	∅,*	—	∅,id_S	id_U,*	id_U,*	id_U,∅	id_U,id_S	∅,*	∅,*	—	∅,id_S	∅,*	∅,*	—	∅,id_S
T,R	∅,id_R	∅,*	—	—	∅,id_R	∅,*	—	—	id_T,id_R	id_T,*	id_T,∅	id_T,∅	∅,id_R	∅,*	—	—
T,U	∅,*	∅,id_U	—	—	∅,*	∅,id_U	—	—	id_T,*	id_T,id_U	id_T,∅	id_T,∅	∅,*	∅,id_U	—	—
T,T	∅,*	∅,*	∅,id_T	—	∅,*	∅,*	∅,id_T	—	id_T,*	id_T,*	id_T,id_T	id_T,∅	∅,*	∅,*	∅,id_T	—
T,S	∅,*	∅,*	—	∅,id_S	∅,*	∅,*	—	∅,id_S	id_T,*	id_T,*	id_T,∅	id_T,id_S	∅,*	∅,*	—	∅,id_S
S,R	*,id_R	*,*	*,∅	*,∅	∅,id_R	∅,*	—	—	∅,id_R	∅,*	—	—	id_S,id_R	id_S,*	id_S,∅	id_S,∅
S,U	*,*	*,id_U	*,∅	*,∅	∅,*	∅,id_U	—	—	∅,*	∅,id_U	—	—	id_S,*	id_S,id_U	id_S,∅	id_S,∅
S,T	*,*	*,*	*,id_T	*,∅	∅,*	∅,*	∅,id_T	—	∅,*	∅,*	∅,id_T	—	id_S,*	id_S,*	id_S,id_T	id_S,∅
S,S	*,*	*,*	*,∅	*,id_S	∅,*	∅,*	—	∅,id_S	∅,*	∅,*	—	∅,id_S	id_S,*	id_S,*	id_S,∅	id_S,id_S

4.3 Migration

From SIMILE to APES (F: $C_S \longrightarrow C_A$). As architectural and semantic function categories are composed of the same set of objects, pair-to-pair correspondence can be summarized as object-to-object correspondence: K, L, and M in SIMILE correspond respectively to R, Q, and [U, T and S] in APES (Table 5).

tel-00458265, version 1 - 19 Feb 2010

Table 5. Summarized mapping of the language classes from SIMILE to APES.

$Ob(C_S)$	Simile class			APES class	$Ob(C_A)$
K	Submodel			CMyStrategy	R
M	Compartment			CRate	U
M	Flow			CState	U
		Parameter		CParameter	S
			Connected to a compartment	CStateExternal	T
M	Information	Input	Connected to a flow	CRateExternal	T
			Other	CExogeneous	T
		Intermediary variable		CAuxiliary	U
L	Command			Statement	Q

Establishing the correspondence of morphisms amounts to checking the cell content correspondence between Table 2 (source) and Table 4 (target). As they correspond to input data, S and T could not be the target of the semantic function morphisms (calling sequence). In Table 6, the target cells whose source content are not preserved in the migration are shown in grey. All grey cells correspond to architecture, i.e. (*,-) $\longrightarrow$ (∅,-). For Q, architecture and semantic function are preserved (not shown).

Table 6. Morphism preservation in the SIMILE to APES migration. In grey, cells whose source content is not preserved in the migration. All grey cells correspond to architecture, i.e. (*,-) $\longrightarrow$ (∅, -). To improve legibility, '–' is used in place of '(∅,∅)'.

Hom(x,y)	R,R	R,U	R,T	R,S	U,R	U,U	U,T	U,S	T,R	T,U	T,T	T,S	S,R	S,U	S,T	S,S
R,R	id_R,id_R	$id_R,*$	$id_R,\varnothing$	$id_R,\varnothing$	$\varnothing,id_R$	$\varnothing,*$	—	—	$\varnothing,id_R$	$\varnothing,*$	—	—	$\varnothing,id_R$	$\varnothing,*$	—	—
R,U	$id_R,*$	id_R,id_U	$id_R,\varnothing$	$id_R,\varnothing$	$\varnothing,*$	$\varnothing,id_U$	—	—	$\varnothing,*$	$\varnothing,id_U$	—	—	$\varnothing,*$	$\varnothing,id_U$	—	—
R,T	$id_R,*$	$id_R,*$	id_R,id_T	$id_R,\varnothing$	$\varnothing,*$	$\varnothing,*$	$\varnothing,id_T$	—	$\varnothing,*$	$\varnothing,*$	$\varnothing,id_T$	—	$\varnothing,*$	$\varnothing,*$	$\varnothing,id_T$	—
R,S	$id_R,*$	$id_R,*$	$id_R,\varnothing$	id_R,id_S	$\varnothing,*$	$\varnothing,*$	—	$\varnothing,id_S$	$\varnothing,*$	$\varnothing,*$	—	$\varnothing,id_S$	$\varnothing,*$	$\varnothing,*$	—	$\varnothing,id_S$
U,R	$\varnothing,id_R$	$\varnothing,*$	—	—	id_U,id_R	$id_U,*$	$id_U,\varnothing$	$id_U,\varnothing$	$\varnothing,id_R$	$\varnothing,*$	—	—	$\varnothing,id_R$	$\varnothing,*$	—	—
U,U	$\varnothing,*$	$\varnothing,id_U$	—	—	$id_U,*$	id_U,id_U	$id_U,\varnothing$	$id_U,\varnothing$	$\varnothing,*$	$\varnothing,id_U$	—	—	$\varnothing,*$	$\varnothing,id_U$	—	—
U,T	$\varnothing,*$	$\varnothing,*$	$\varnothing,id_T$	—	$id_U,*$	$id_U,*$	id_U,id_T	$id_U,\varnothing$	$\varnothing,*$	$\varnothing,*$	$\varnothing,id_T$	—	$\varnothing,*$	$\varnothing,*$	$\varnothing,id_T$	—
U,S	$\varnothing,*$	$\varnothing,*$	—	$\varnothing,id_S$	$id_U,*$	$id_U,*$	$id_U,\varnothing$	id_U,id_S	$\varnothing,*$	$\varnothing,*$	—	$\varnothing,id_S$	$\varnothing,*$	$\varnothing,*$	—	$\varnothing,id_S$
T,R	$\varnothing,id_R$	$\varnothing,*$	—	—	$\varnothing,id_R$	$\varnothing,*$	—	—	id_T,id_R	$id_T,*$	$id_T,\varnothing$	$id_T,\varnothing$	$\varnothing,id_R$	$\varnothing,*$	—	—
T,U	$\varnothing,*$	$\varnothing,id_U$	—	—	$\varnothing,*$	$\varnothing,id_U$	—	—	$id_T,*$	id_T,id_U	$id_T,\varnothing$	$id_T,\varnothing$	$\varnothing,*$	$\varnothing,id_U$	—	—
T,T	$\varnothing,*$	$\varnothing,*$	$\varnothing,id_T$	—	$\varnothing,*$	$\varnothing,*$	$\varnothing,id_T$	—	$id_T,*$	$id_T,*$	id_T,id_T	$id_T,\varnothing$	$\varnothing,*$	$\varnothing,*$	$\varnothing,id_T$	—
T,S	$\varnothing,*$	$\varnothing,*$	—	$\varnothing,id_S$	$\varnothing,*$	$\varnothing,*$	—	$\varnothing,id_S$	$id_T,*$	$id_T,*$	$id_T,\varnothing$	id_T,id_S	$\varnothing,*$	$\varnothing,*$	—	$\varnothing,id_S$
S,R	$*,id_R$	$*,*$	$*,\varnothing$	$*,\varnothing$	$\varnothing,id_R$	$\varnothing,*$	—	—	$\varnothing,id_R$	$\varnothing,*$	—	—	id_S,id_R	$id_S,*$	$id_S,\varnothing$	$id_S,\varnothing$
S,U	$*,*$	$*,id_U$	$*,\varnothing$	$*,\varnothing$	$\varnothing,*$	$\varnothing,id_U$	—	—	$\varnothing,*$	$\varnothing,id_U$	—	—	$id_S,*$	id_S,id_U	$id_S,\varnothing$	$id_S,\varnothing$
S,T	$*,*$	$*,*$	$*,id_T$	$*,\varnothing$	$\varnothing,*$	$\varnothing,*$	$\varnothing,id_T$	—	$\varnothing,*$	$\varnothing,*$	$\varnothing,id_T$	—	$id_S,*$	$id_S,*$	id_S,id_T	$id_S,\varnothing$
S,S	$*,*$	$*,*$	$*,\varnothing$	$*,id_S$	$\varnothing,*$	$\varnothing,*$	—	$\varnothing,id_S$	$\varnothing,*$	$\varnothing,*$	—	$\varnothing,id_S$	$id_S,*$	$id_S,*$	$id_S,\varnothing$	id_S,id_S

From APES to SIMILE (G: $C_A \longrightarrow C_S$). Table 7 shows the mapping of the language objects from APES to SIMILE. The same methodology as applied to F enabled us to identify the preservation of G migration. In that direction all morphisms were mapped. All aspects of the 2 features are then preserved.

tel-00458265, version 1 - 19 Feb 2010

Table 7. Summarized mapping of the classes from APES to SIMILE.

$Ob(C_A)$	APES classes	SIMILE classes	$Ob(C_S)$
R	CMyStrategy	Submodel	K
U	CState	Compartment	M
U	CRate	Flow	M
U	CAuxiliary	Information	M
T	CStateExternal	Information	M
T	CRateExternal	Information	M
T	CExogeneous	Information	M
S	CParameter	Information	M
Q	Statement	Command	L

Verification of identity morphism mapping (condition 4) consisted in checking that the identity morphisms of APES corresponds to the identity morphisms of SIMILE. Table 8 shows that correspondence.

Table 8. Identity morphism correspondence between APES and SIMILE.

$Ob(C_A)$	id_{APES}	$Ob(C_S)$	id_{Sim}	$Ob(C_A)$	id_{APES}	$Ob(C_S)$	id_{Sim}
R,R	id_R,id_R	K,K	id_K,id_K	T,R	id_T,id_R	M,K	id_M,id_K
R,U	id_R,id_U	K,M	id_K,id_M	T,U	id_T,id_U	M,M	id_M,id_M
R,T	id_R,id_T	K,M	id_K,id_M	T,T	id_T,id_T	M,M	id_M,id_M
R,S	id_R,id_S	K,M	id_K,id_M	T,S	id_T,id_S	M,M	id_M,id_M
U,R	id_U,id_R	M,K	id_M,id_K	S,R	id_S,id_R	M,K	id_M,id_K
U,U	id_U,id_U	M,M	id_M,id_M	S,U	id_S,id_U	M,M	id_M,id_M
U,T	id_U,id_T	M,M	id_M,id_M	S,T	id_S,id_T	M,M	id_M,id_M
U,S	id_U,id_S	M,M	id_M,id_M	S,S	id_S,id_S	M,M	id_M,id_M

Verification of condition 5 consisted in checking that $G(g \circ f) = G(g) \circ G(f)$ for each pair of morphisms f and g in table 4. For example:

$$\begin{array}{ccc} f & \longrightarrow & G(f) \\ Hom_{Arch}(T,U) \times Hom_{SF}(U,R) & & Hom_{Arch}(M,M) \times Hom_{SF}(M,K) \\ <\varnothing,*> & & <id_M,*> \end{array}$$

$$\begin{array}{ccc} g & \longrightarrow & G(g) \\ Hom_{Arch}(R,T) \times Hom_{SF}(R,U) & & Hom_{Arch}(K,M) \times Hom_{SF}(K,M) \\ <id_R,*> & & <id_K,id_M> \end{array}$$

$$\begin{array}{ccc} g & \circ & f \\ Hom_{Arch}(R,T) \times Hom_{SF}(R,U) & & Hom_{Arch}(T,U) \times Hom_{SF}(U,R) \\ & Hom_{Arch}(R,U) \times Hom_{SF}(R,R) & \\ & <id_R,*> & \end{array}$$

$$\begin{array}{ccc} G(g) & \circ & G(f) \\ Hom_{Arch}(K,M) \times Hom_{SF}(K,M) & & \\ Hom_{Arch}(M,M) \times Hom_{SF}(M,K) & & \\ & Hom_{Arch}(K,M) \times Hom_{SF}(K,K) & \\ & <id_K,*> . & \end{array} \quad (3)$$

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

$$\begin{array}{ccc} g \circ f & \longrightarrow & G(g \circ f) \\ \mathrm{Hom}_{Arch}(R,U) \times \mathrm{Hom}_{SF}(R,R) & & \mathrm{Hom}_{Arch}(K,M) \times \mathrm{Hom}_{SF}(K,K) \\ <id_R,*> & & <id_K,*> . \end{array} \quad (4)$$

Since (3) = (4), $G(g) \circ G(f) = G(g \circ f)$ is verified for this pair of morphisms. Similar calculations made on each pair of morphisms in Table 4 reveals that G verifies the condition 5.

5 Discussion and conclusion

UML enabled us to transcribe biological modelling languages using aggregation and association relationships between classes. This construction was possible on two contrasting languages commonly used in biology. Since our target was program migration, we only considered the terminal classes of the language and not inheritance. We obtained a mathematical representation of the language, where aggregation is expressed as inclusion and association as order relationship.

In our illustration, we established the functorial relationship APES $\longrightarrow$ SIMILE and the application is surjective. By construction, the functor provided both the object and morphism relationships. As each morphism corresponds to a translation rule, the construction of the migration function simply consists in instantiating the functor. According to [16], two programs are equivalent if they provide identical mapping between a set of inputs with a set of outputs. In the category theory, the concept of equivalence is defined using the functor properties. Our construction supported that this definition of the equivalence of programs can be extended to architecture.

Conversely, SIMILE $\longrightarrow$ APES mapping is incomplete because of differences in architecture. With this example, we illustrated the advantages of applying the category theory to identify the aspects of migration which could not be preserved due to language expressiveness. In that case, conferring the properties necessary to verify the condition for functor establishment provides a support for language improvement. This aspect is not covered in this paper. Future work will consist in establishing the functorial bijection enabling bidirectional migration of the improved languages.

Acknowledgments. This publication was partially funded under the SEAMLESS integrated project, EU 6th Framework Programme for Research, Technological Development and Demonstration, Priority 1.1.6.3. Global Change and Ecosystems (European Commission, DG Research, contract no. 010036-2).

6 References

1. Sequeira, R.A., Olson, R.L., McKinion, J.: Implementing generic, object-oriented models in biology. Ecol. Model. 94, 17--31 (1997).
2. Von Bertalanffy, L.: Théorie générale des systèmes. Dunod, 2nd edition (2002).
3. Fritzson, P.: Principles of Object-Oriented Modelling and Simulation with Modelica 2.1. Wiley-IEEE Press (2004).
4. Costanza, R., Voinov, A.: Modelling ecological and economic systems with STELLA: Part III. Ecol. Model. 143, 1--7 (2001).
5. Muetzelfeldt, R, Massheder, J.: The SIMILE visual modelling environment. Europ. J. Agron. 18, 345--358 (2003).

tel-00458265, version 1 - 19 Feb 2010

tel-00458265, version 1 - 19 Feb 2010

6. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design patterns – Elements of Reusable Object-Oriented Software (1st edition). Addison Westley Professional (2001).
7. Medvidovic, N., Taylor, R.N.: A Classification and Comparison Framework for Software Architecture Description Languages. IEEE Trans. Software Eng. 26(1), 70--93 (2000).
8. Donatelli, M., Bellocchi, G., Carlini, L.: Sharing knowledge via software components: models on reference evapotranspiration. Europ. J. Agron. 24 (2), 186--192 (2006).
9. Mens, T., Van Gorp, P.: A Taxonomy of Model Transformation. Electr. Notes Theor. Comput. Sci. 152, 125--142 (2006).
10. Stevens, P.: Bidirectional model transformation in QVT: semantic issues and open questions. In: Proceedings of 10th International Conference on Model Driven Engineering Languages and Systems, pp. 1--15. Springer, Heidelberg (2007).
11. Visser, E.: A survey of strategies in rule-based program transformation systems. J. Symb. Comput. 40(1), 831--873 (2005).
12. Königs, A, Schürr, A.: Tool Integration with Triple Graph Grammars - A Survey. Electr. Notes Theor. Comput. Sci. 148(1), 113--150 (2006).
13. Grunske, L., Geiger, L., Lawley, M.: A Graphical Specification of Model Transformations with Triple Graph Grammars. Lect. Notes Comput. Sci, 3748, 284--298 (2005).
14. Bernardo, M., Bonta, E.: Preserving architectural properties in multithreaded code generation. Lect. Notes Comput. Sci. 3454, 188--203 (2005).
15. Garcia AD., Haeusler, EH.: Code Migration and program maintainability - a categorical perspective. Information Processing Letters. 79 (5), 249--254 (2001).
16. Ward, M.P., Zedan, H.: Slicing as a program transformation. ACM Trans. Program. Lang. Syst. 29(2), n°7 (2007).
17. Barr, M, Wells, C.: Category Theory for Computing Science. Prentice Hall, Englewood Cliffs, NJ (1995).
18. Guo, J.: Using Category Theory to Model Software Component Dependencies. In: 9th IEEE International Conference on Engineering of Computer-Based Systems, pp. 185--194. IEEE Press, New York (2002).
19. Barr, M., Wells, C.: Toposes, Triples and theories. Reprints in Theory and Applications of Categories, 12, 1--288 (2005).
20. Mac Lane, S.: Categories for the Working Mathematician. Springer (1998)
21. Simulistics Ltd, http://www.simulistics.com
22. Forrester, J.W.: World Dynamics. Cambridge Mass. Wright-Allen Press (1971).
23. Agricultural Production and Externalities Simulator. http://www.apesimulator.it
24. Van Ittersum, M.K., Ewert, F., Heckelei, T., Wery, J., Alkan Olsson, J., Andersen, E., Bezlepkina, I., Brouwer, F., Donatelli, M., Flichman, G., Olsson, L., Rizzoli, A., Van der Wal, T., Wien, J.E., Wolf, J. Integrated assessment of agricultural systems - A component-based framework for the European Union (SEAMLESS). Agric. Syst., 98, 150--165 (2008).
25. System for Environmental and Agricultural Modelling; Linking European Science and Society, http://www.seamless-ip.org/
26. Szypersky, C., Gruntz, D., Murer, S.: Component software – beyond object-oriented programming. 2nd ed. Addison-Wesley, London (2002).
27. Hillyer, C., Bolte, J., van Evert, F., Lamaker, A.: The ModCom modular simulation system. Europ. J. Agron. 18, 333--343 (2003).
28. Hill, D.R.C.: Object-Oriented Analysis and Simulation. Addison-Wesley, Boston (1996).
29. Donatelli, M: Unpublished data. Task Leader of APES development, SEAMLESS EU 6 Framework Research Programme , (2007).

tel-00458265, version 1 - 19 Feb 2010

L'assemblage de programmes au sein de plateformes logicielles :
Syntaxe, Sémantique et Pragmatique.
Application aux plateformes dédiées aux simulations en agronomie.

Les systèmes envisagés par les sciences du vivant (génétique, agronomie, écologie etc.) sont pour la plupart des systèmes complexes. Depuis les années 1960, l'informatique permet aux équipes de recherche de représenter les systèmes qu'elles étudient sous la forme de programmes de simulation numérique. L'idée de réutiliser les programmes et de les assembler a donné naissance aux plateformes logicielles. Pour l'agronomie, l'enjeu des plateformes est de répondre aux questions posées par le Développement Durable impliquant durabilité écologique, viabilité économique et équité sociale.
La thèse traite de questions sémantiques générées par l'assemblage de programmes. Pour notre étude, nous avons mis au point une trame d'analyse originale établie sur la base de travaux en cybernétique, ingénierie logicielle et intelligence artificielle. La trame considère les niveaux morphologique, syntaxique et sémantique des langages d'assemblage. Au travers de l'analyse d'exemples (MODCOM, OpenMI, VLE, APES, DSSAT et SEAMLESS-IF), nous montrons que les règles syntaxiques inhérentes aux langages d'assemblages limitent leur aptitude à représenter les systèmes complexes. Nous montrons aussi que l'efficience des métadonnées et ontologies, destinées à faire coïncider les entrées / sorties des programmes, dépend des aspects implicites ayant prévalu à la représentation des systèmes.
Afin d'accéder à l'implicite, nous proposons de décrire le contenu du programme de façon formelle. Un modèle de description a été établi à partir de la linguistique pragmatique et de travaux en morphosyntaxe. L'interrogation partielle en mode direct permet de conférer une structure de description détaillée, basée sur les sept pronoms et adverbes que propose la langue française. Ce travail débouche sur la proposition d'un langage déclaratif de description de systèmes complexes.

Mots clés : Interaction, Couplage, Simulation, Systèmes Complexes, Linguistique.

The assembly of programs in software platforms: syntax, semantic, and pragmatics.
Application to platforms devoted to simulations in agronomy.

The systems envisaged by the life sciences (genetics, agronomy, ecology etc.) are, for the most part, complex systems. In the 1960s, the advent of the computer enabled the capitalization of knowledge gathered in computer programs. The idea of accessing other knowledge by joining programs gave birth to software platforms. The current issue of such software platforms is to answer new questions raised by sustainable development, which involves environmental sustainability, economic viability and social equity.
The thesis deals with semantic issues generated by assembly. For our study, we developed an original analysis framework elaborated on the basis of work in cybernetics, software engineering and artificial intelligence. The analysis framework considers morphological, syntactic, and semantic levels of the assembly language. Three generic assembly media (MODCOM, OpenMI, and VLE) and three agronomic software platforms (APES, DSSAT, and SEAMLESS-IF) were analysed. We show that the syntactic rules inherent to assembly language limit its ability to represent some complex systems. We also show that tacit knowledge inherent to programs construction limits the efficiency of metadata and ontology, both used to establish the relationship between the input and the output of programs.
To clear up the different aspects of sense, we propose a model for program description established using pragmatic linguistic results and morphosyntactic work. The description is based on the seven pronouns and adverbs that the French language offers. A typology of responses is associated with each of these interrogative pronouns and adverbs using a linguistic bibliography. Our work leads to propose of a declarative language that may be used to describe complex systems.

Keywords: Interaction, Coupling, Simulation, Complex Systems, Linguistics.