

OntoCASE, une approche d'élicitation semi-formelle graphique et son outil logiciel pour la construction d'une ontologie de domaine

Michel Héon

Cotechnoe, 2356 Ch Bourbonnière, Lachute, Québec, Canada
heon@cotechnoe.com; www.cotechnoe.com

Résumé : Toutes activités de gestion des connaissances nécessitent la mise en place d'un mécanisme de représentation de la connaissance. La modélisation semi-formelle graphique est une technique de représentation qui se démocratise de plus en plus grâce à sa convivialité d'utilisation. Cependant, le modèle semi-formel produit est un document informatique non-formel qui contient des éléments de langage polysémiques (possédant plusieurs significations). Cette polysémie, caractéristique des langages semi-formels, ne permet pas le traitement de ce type de modèle par un agent informatique intelligent. Deux besoins sont ainsi confrontés. D'une part, le besoin de l'expert de contenu qui aspire à l'utilisation d'un langage de représentation de connaissances qui soit expressif, simple d'usage et qui stimule sa créativité, et d'autre part, le besoin du cognicien d'utiliser un langage de représentation de la connaissance de degré formel pour la conception de systèmes logiciels intelligents. Cet article présente OntoCASE (acronyme d'*Ontology CASE Tool*), un atelier d'ingénierie ontologique qui facilite l'expression et l'expertise et qui par la suite assiste le cognicien pour la formalisation du modèle semi-formel en une ontologie formelle. OntoCASE se compose d'un éditeur de modèle semi-formel (eLi), d'un système expert à la désambiguïsation, un système expert à la conversion du modèle semi-formel en ontologie OWL ainsi qu'un système expert à la validation de l'ontologie produite.

Mots clés: Modèle semi-formel, Formalisation de modèles, Transformation de modèle, Ingénierie ontologique, Ontologie, Modélisation des connaissances, Mémoire organisationnelle, Semantic Web Rule Language, Ontology Web Language, Modélisation par Objets Typés

INTRODUCTION

Cet article présente OntoCASE (Héon, 2010, 2012), un outil méthodologique et informatique de transformation d'un modèle semi-formel graphique en ontologie formelle dans le formalisme de l'*Ontology Web Language* (W3C OWL, 2004), il apporte une contribution méthodologique et instrumentale au domaine de l'ingénierie ontologique (Gašević *et al.*, 2006 ; Gómez-Pérez *et al.*, 2003 ; Gruber, 1993 ; Uschold et Gruninger, 1996) ainsi qu'en gestion des connaissances. Dans

ses grandes lignes, la construction d'une ontologie nécessite une étape d'élicitation des connaissances du domaine visé, suivie d'une étape de formalisation et d'une étape de validation. La nature des connaissances à représenter, l'hétérogénéité du support des connaissances à formaliser (documents plus ou moins formels, communication orale ou écrite, représentations graphiques, connaissances tacites, etc.) et le manque d'assistance automatisée rendent souvent laborieuse la démarche de conception d'une ontologie.

Gestion des connaissances et systèmes à base de connaissances

La connaissance et sa gestion sont des enjeux qui sont au centre des préoccupations des organisations. Qu'il s'agisse du départ à la retraite d'un nombre important d'employés, d'une restructuration, de l'accréditation à une norme internationale ou du traitement d'une abondance d'informations, les organisations sont plus que jamais conscientes de l'importance de gérer adéquatement leur patrimoine de connaissances. L'émergence des technologies de l'information a permis d'automatiser et de rendre efficaces les processus de *partage*, de *distribution*, de *capture* et de *création* de la connaissance. En comparaison avec les systèmes traditionnels, les systèmes à base de connaissances (SBC) offrent des avantages certains, par exemple, la création automatique de nouvelles connaissances. Le tableau 1 (extrait et adapté de Weber et Kaplan, 2002) présente une liste d'applications informatiques qui assistent le processus de gestion des connaissances selon qu'elles soient issues de l'informatique traditionnelle ou de l'informatique des SBC. La venue des technologies associées au web sémantique a permis la normalisation des outils servant à la construction de SBC dont l'*ontologie* est le cœur.

Tableau 1. Gestion des connaissances et applications informatiques (extrait et adapté de Weber et Kaplan 2002)

	Partager la connaissance	Distribuer, diffuser la connaissance	Capter et codifier	Créer de la connaissance
Systèmes traditionnels	Courriel, Intranet, Web social d'entreprise	Traitement de texte, Répertoire de fichiers, base de données	Tous les systèmes de codification de la connaissance	Logiciels d'analyse de statistique qui assistent lors de séances de remue-méninge
Systèmes à base de connaissances	Référencement sémantique, systèmes à base d'ontologies	Système à base de cas, système à base de règles, système à base de <i>workflow</i> , systèmes experts	Tous les systèmes à base de connaissances qui servent à la codification ou à l'acquisition de la connaissance	Système de création de la connaissance, de forage de données

Modélisation semi-formelle

La construction d'une ontologie est un processus complexe qui demande une expertise poussée. Il est rare qu'un expert de contenu possède les connaissances de programmation informatique nécessaires pour coder sa connaissance dans une ontologie. En revanche, la modélisation semi-formelle de la connaissance est un outil souvent accessible pour qui souhaite représenter des connaissances. Un langage de degré semi-formel dans le sens d'Uschold et Gruninger (1996) est un formalisme d'expression de la connaissance dont la souplesse grammaticale et sémantique rend son utilisation conviviale. Ce type de formalisme, qui fait généralement appel au format graphique, est parfois employé pendant l'étape de conception d'un système (Rumbaugh *et al.*, 1999) ou encore pour favoriser le transfert d'expertise dans les organisations (Basque *et al.*, 2004) ou l'apprentissage dans des situations éducatives (Basque et Pudelko, 2010b ; Kinshuk *et al.*, 2008 ; Paquette, 2002) ainsi que pour susciter les échanges pendant une séance de remue-méninges, ou plus simplement pour représenter graphiquement un énoncé. Bien qu'un modèle semi-formel contienne toujours des éléments d'ambiguïté, sa souplesse d'expression, surtout lorsqu'il fait appel à un format graphique, permet d'accéder plus facilement à l'identification des connaissances tacites des experts. Dans un tel cadre, la spontanéité n'est pas bloquée par une charge cognitive trop lourde associée à une formalisation poussée de la pensée (Basque *et al.*, 2008). Ainsi, le langage semi-formel graphique fournit un certain *guidage représentationnel* (Suthers, 2003) qui structure le processus de modélisation et qui peut servir de première itération d'élicitation pour la conception d'une ontologie formelle (Héon, 2010). Nous pensons que l'usage d'un système de représentation plus convivial que les systèmes de représentation ontologique, tels ceux utilisés dans les éditeurs d'ontologies à la Protégé (Stanford Medical Informatics, 2006), pourrait permettre :

- a) d'élargir le bassin des personnes aptes à contribuer activement aux premières étapes de la représentation de leurs connaissances et ce, sans l'aide d'un cogniticien;
- b) d'économiser du temps lors de l'élicitation des connaissances des experts, les libérant ainsi plus rapidement pour la réalisation de leur travail habituel;
- c) de représenter des types de connaissances autres que déclaratif intrinsèque aux ontologies, notamment les connaissances de type procédural et stratégique.

C'est dans les années 1970 que Tony Buzan (Buzan et Buzan, 1994) a développé une technique de représentation de connaissances, appelée *Mind Mapping*, qui fait appel à un langage graphique combinant du texte, des images et des lignes pour représenter une idée ou un thème. S'apparentant au graphe d'entités-relations de Chen (1976), la technique de la *carte conceptuelle* (*concept mapping*) développée par Joseph Novak en 1972 (Novak et Cañas, 2006) permet de représenter une hiérarchisation de concepts selon une lecture du haut vers le bas. Le langage de modélisation par objets typés (MOT) développé par l'équipe de Gilbert Paquette (Paquette, 2010) est aussi un langage de représentation de connaissances de degré semi-formel. Spécialement conçu pour l'ingénierie pédagogique, ce langage a aussi été utilisé dans des projets visant le transfert d'expertise dans les organisations (Basque *et al.*, 2008) et l'apprentissage dans des situations de formation (Basque et Pudelko, 2010a). Ce langage permet la représentation de connaissances déclaratives, procédurales, stratégiques, abstraites et factuelles ainsi que la mise en relation de ces connaissances par des relations typées (spécialisation, composition, instanciation, intransitif/produit, précédence, régulation et application). L'utilisation d'objets typés ainsi que la propriété de représenter des connaissances selon un double niveau d'abstraction (factuel et abstrait) font de ce langage un système de représentation apte à être traité par un système de raisonnement automatique dans la mesure où la polysémie associée à certaines primitives est levée.

Ingénierie ontologique

En tant qu'artefact informatique, la construction d'une ontologie implique un processus d'ingénierie adéquat. La figure 1 présente le processus d'ingénierie ontologique selon Staab *et al* (2001). Le processus se divise en cinq phases : 1) *l'étude de faisabilité, et la spécification des exigences du système à venir* pour déterminer la viabilité et la rentabilité du système à venir, 2) *l'amorce de la construction d'une ontologie de départ* pour produire l'esquisse et la structure de travail de l'ontologie à venir, 3) *la construction de l'ontologie* pour éliciter et formaliser les connaissances expertes, 4) *l'évaluation de l'ontologie* pour tester l'ontologie dans l'environnement d'exécution cible, et 5) *la maintenance de l'ontologie* pour assurer la mise à jour des connaissances contenues dans l'ontologie. Les phases 3, 4 et 5, qui sont itératives et incrémentales, sont régies par le cognicien et elles nécessitent l'implication de l'expert de contenu.

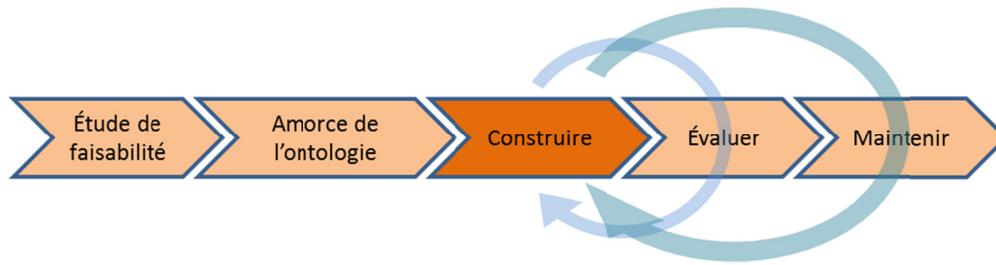
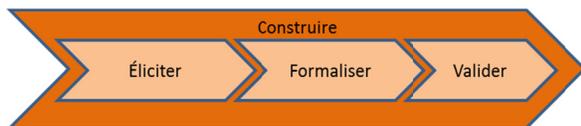


Figure 1. Processus classique d'ingénierie ontologique selon Staab *et al.*, (2001)

Présenté à la figure 2 a) le processus classique de construction d'une ontologie se décompose en trois étapes distinctes qui sont exécutées de façon linéaire : *éliciter la connaissance*, *formaliser la connaissance* et *valider la connaissance*. L'élicitation consiste à interviewer l'expert de contenu afin d'extraire les concepts, les propriétés, les axiomes et les faits concernant un domaine de connaissances. L'étape de formalisation, qui normalement est réalisée par le cogniticien, consiste à traduire la connaissance de l'expert en ontologie informatique. L'étape de validation a pour objectif de valider la véracité de la connaissance exprimée dans l'ontologie.

a) Processus de construction classique



b) Processus de construction avec OntoCASE



Figure 2. OntoCASE dans le processus classique d'ingénierie ontologique

À cause de la linéarité de son déroulement, le processus classique de construction d'une ontologie est un processus chronophage qui demande beaucoup de ressources et de coordination. Après la séance d'élicitation, le cogniticien s'isole pour formaliser en ontologie la connaissance experte. À la fin de l'étape de formalisation, le cogniticien valide le contenu de l'ontologie auprès l'expert. Le cycle complet de construction peut nécessiter plusieurs jours de travail et la coordination de plusieurs rencontres entre l'expert et le cogniticien.

La figure 2 b) présente le processus de construction selon la méthodologie d'OntoCASE. En plus d'être itératif et incrémental, le processus de construction utilise la modélisation semi-formelle comme langage d'élicitation entre l'expert de contenu et le cogniticien. Les principaux avantages de cette approche sont : que les modèles semi-formels produits peuvent servir d'illustration pour expliquer le contenu de l'ontologie; que la construction de l'ontologie est réalisée en présence de l'expert; que l'étape de formalisation est raccourcie puisqu'elle est

instrumentée par un assistant informatique, et que la validation est réalisée, à chaud, pendant les échanges avec l'expert. S'il y a désaccord sur le contenu de l'ontologie, l'élicitation est spontanément reprise pour amorcer une nouvelle itération.

Compte tenu de l'importance de la formalisation des connaissances dans le processus de construction d'une ontologie, quelques auteurs proposent des solutions pour formaliser un modèle semi-formel en ontologie. Eskridge, Hayes et Hoffman (2006) ont conçu l'application « CmapTools Ontology Editor » (COE) pour la formalisation en ontologie de carte conceptuelle. Castro *et al.* (2006) utilisent la carte conceptuelle pour l'élicitation semi-formelle de l'expertise avant de la formaliser en ontologie OWL au moyen de COE. Quant à Park et Hunting (2003), plutôt orientés vers le « TopicMaps », ils proposent un ensemble de méthodes, techniques et technologies pour intégrer le référencement sémantique d'objets du Web par l'intermédiaire d'un modèle de type « ConceptMap ». Il ressort de ces solutions une faiblesse dans la formalisation puisque plusieurs éléments de l'expressivité d'OWL, comme la restriction, sont difficilement représentables dans un langage comme le « TopicMap » ou le « ConceptMap ».

Nous présentons dans cet article, une méthodologie et un outil CASE pour la formalisation de modèle MOT en ontologie OWL. L'outil OntoCASE (Héon, 2010) est un outil méthodologique et logiciel qui soutient le processus d'ingénierie ontologique présenté à la figure 2 b). Cet article débute par la présentation du langage MOT, le langage semi-formel d'OntoCASE, qui sert à l'élicitation. Suivra une section sur la présentation de la méthodologie et une section sur la présentation de l'aspect informatique d'OntoCASE. Finalement, avant de conclure, il sera présenté la section expérimentation, qui permettra de présenter une évaluation d'OntoCASE.

1. LANGAGE DE REPRÉSENTATION SEMI-FORMEL MOT

Le langage de Modélisation par Objets Typés (MOT) et le logiciel qui l'implémente (MOTPlus) ont été conçus par une équipe sous la direction de Paquette (Paquette, 2010). De degré semi-formel, MOT est le langage qui a été utilisé pour développer la méthodologie d'OntoCASE. MOT est une variante du diagramme entité-relation de Chen (1976) dont les entités et les relations sont préalablement typés. Ce langage a été choisi pour la conception de la méthodologie en raison du fait que les entités qui le composent sont regroupées en deux niveaux d'abstraction, le niveau *abstrait* et le niveau *factuel*. Le niveau abstrait renvoie à l'idée de la *Terminological Box* (TBox) en logique de description (DL) (Baader *et al.*, 2007). La TBox est

l'espace de modélisation qui permet de représenter le niveau terminologique d'un sujet. On pourrait aussi dire qu'il fait référence aux abstractions d'un domaine du discours. Par exemple, pour le domaine de la *Famille*, c'est dans le niveau abstrait que se définissent les concepts tels que : *Père, Mère, Enfant, Parent* et les propriétés tel que *estEnfantDe, estFrèreDe, etc.* Quant au niveau factuel, qui renvoi en DL à l'idée de l'*Assertional Box (ABox)*, il permet de représenter les *faits* concernant un domaine du discours. Le fait est une entité observable de la réalité. Ainsi, toujours dans l'exemple du domaine de la *Famille*, l'alphabet du niveau factuel est utilisé pour représenter des faits tels que *ANDRÉ, MARIE, MARC, etc.* Le niveau factuel permet aussi de représenter la mise en relation de deux faits par la déclaration d'un énoncé selon la structure *Sujet/Prédicat/Objet* (par exemple : *MARC estEnfantDe MARIE*). À notre connaissance, MOT est le seul langage semi-formel graphique qui possède cette caractéristique. Au surplus, MOT permet la représentation dans un même diagramme, plusieurs types de connaissance telle que la connaissance déclarative, procédurale, stratégique, factuelle et abstraite. Quant aux relations, elles permettent la représentation d'association de type : composition, sorte de, type de, préséance, intransitif/produit et régulation.

1.1. Vocabulaire et sémantique des entités typés et des relations typées de MOT

Le tableau 2 présente les six éléments du vocabulaire de MOT. Du point de vue de la sémantique, le *concept* représente « le quoi » des choses. Il sert à décrire l'essence d'un objet concret. Il peut être associé à l'idée de classe ou de catégorie. En ce sens, il est l'abstraction d'un objet concret.

Tableau 2. Éléments du vocabulaire des six éléments typés de MOT

Type de connaissance	Connaissance abstraite	Connaissance factuelle
Déclarative <i>Le quoi des choses</i>	Concept 	Exemple 
Action <i>Le comment de choses</i>	Procédure 	Trace 
Stratégique <i>Le pourquoi, le quand, le qui</i>	Principe 	Énoncé 

L'*exemple* représente l'un de ces objets en énonçant un certain nombre de faits qui le décrivent. La *procédure* permet de décrire « le comment » des choses. Elle désigne des opérations, des actions pouvant être accomplies. La *trace* représente l'ensemble des faits concrets obtenus lors de

l'exécution d'une procédure. Le *principe* désigne « le pourquoi », « le quand » ou le « qui » associé à une chose. Il est une connaissance stratégique qui permet de nommer une relation entre des objets, que ce soit des concepts, des procédures ou d'autres principes. Il sert notamment à représenter une condition pouvant s'appliquer à l'exécution d'une action. L'*énoncé* représente l'instanciation d'un principe à propos d'objets concrets.

Chaque type de lien possède une sémantique propre (voir le tableau 3) qui respecte des règles d'intégrité (présentées au tableau 4). Par exemple, un lien de spécialisation (lien S) unit deux connaissances abstraites, qui doivent être de même nature. L'ensemble de ces règles d'intégrité sont décrites dans Paquette (2010).

Tableau 3. Vocabulaire et sémantique des relations typées de MOT

Type de lien	Signification
S	Le lien de spécialisation associe deux connaissances abstraites de même type dont la première est une spécialisation de la seconde. Ce lien est notamment utile dans la description des taxonomies. Le lien de spécialisation est une relation transitive.
I	Le lien d'instanciation associe à une connaissance abstraite un fait qui caractérise une instance de cette connaissance. Le lien d'instanciation n'est pas une relation transitive.
I/P	Le lien intransit/produit sert à associer une connaissance procédurale à une connaissance conceptuelle afin de représenter l'intransit ou le produit d'une procédure. Ce lien est notamment utile dans la description des algorithmes, des processus et des méthodes. Le lien intransit/produit n'est pas une relation transitive.
P	Le lien de précédence associe une connaissance à une autre qui la suit dans une séquence temporelle de procédures ou de règle de décision (principes). Le lien de précédence est une relation transitive.
R	Le lien de régulation associe une connaissance stratégique (un Principe ou un Énoncé) à une autre connaissance afin de préciser une contrainte, une restriction ou une règle qui régit la connaissance. Le lien de régulation qui associe une connaissance déclarative à un principe permet d'associer un domaine à une propriété. Le lien de régulation est une relation non-transitive.
C,C*	Les liens de composition et de composition multiple permettent de représenter l'association entre une connaissance et des connaissances qui la composent. Le lien de composition est une relation transitive.
E	Le lien englobe ne possède pas de symbolique particulière dans le langage MOT original. Il s'agit d'une relation qui unit l'élément d'un modèle aux éléments d'un sous-modèle. Le lien englobe est une relation transitive.

1.2. Règles de la grammaire de MOT

Certaines règles d'association entre des connaissances sources et des connaissances destinations sont appliquées à chacun des types de lien (voir le tableau 4)

Tableau 4. Grammaire du langage MOT

<i>Destination</i>	Connaissances abstraites			Connaissances factuelles		
<i>Origine</i>	Concept	Procédure	Principe	Exemple	Trace	Énoncé
Concept	C, S	I/P	R	I, C		
Procédure	I/P	C, S, P	C, P		I, C	
Principe	R	C, R, P	C, S, P, R			I, C
Exemple				C	I/P	
Trace				I/P	C, P	C, P
Énoncé				R	C, R, P	C, R, P

Ces règles définissent les relations considérées valides entre les différents types de connaissances du point de vue de la sémantique MOT. Plus spécifiquement, il existe un ensemble de règles secondaires qui régissent chacune des relations en fonction de la nature des connaissances d'origine et de destination qu'elles associent.

1.3. Polysémie de MOT

Lorsque qu'un même symbole est utilisé dans une multitude de contextes, il y a surcharge de la sémantique désignée par le symbole (ce que nous avons appelé la *polysémie*). Le tableau 5 présente la polysémie que nous avons identifiée pour chacune des primitives atomiques du langage MOT. Comme il sera vu un peu plus loin, le rôle du processus de désambiguïsation est de permettre à l'utilisateur de choisir le sens approprié au symbole qu'il utilise dans le modèle semi-formel.

Tableau 5. Polysémie de MOT

		ENTITÉ		RELATION		
		Représentation en MOT	Signification (s)	Représentation en MOT	Signification (s)	
Connaissance abstraite	<div style="border: 1px solid black; padding: 2px; display: inline-block;">concept</div> Connaissance déclarative	Classe	Schéma	--C-> Composition	Est composé de	
					Entier	A pour attribut
					Caractère	
		Naturel				
	<div style="border: 1px solid black; border-radius: 50%; padding: 2px; display: inline-block;">procédure</div> Connaissance procédurale	Action	Opération	-IP-> Intrant/Produit	Est l'intrant de	
					A pour produit	
			Est l'instrument utilisé			
	<div style="border: 1px solid black; padding: 2px; display: inline-block;">principe</div> Connaissance stratégique	Agent		--R-> Régulation	Régit	
		Propriété		--P-> Précédence	Puis exécuter	
		Classe de règle			Puis évaluer	
Classe de condition			Permet			
Connaissance factuelle	<div style="border: 1px dashed black; padding: 2px; display: inline-block;">exemple</div> Connaissance déclarative	Objet		Évaluer à partir de		
		Valeur			A pour dépendance	
	<div style="border: 1px dashed black; border-radius: 50%; padding: 2px; display: inline-block;">trace</div> Connaissance procédurale	Acte		--I-> Instance	A pour instance	
		Instruction		--S-> Spécialisation	Est une sorte de	
	<div style="border: 1px dashed black; padding: 2px; display: inline-block;">énoncé</div> Connaissance stratégique	Individu		Icône englobement	Association avec les éléments d'un sous-modèle	
		Assertion				
Condition						
	Règle					

2. MÉTHODOLOGIE ONTOCASE

Nous pensons que le processus de construction d'une ontologie doit être décomposé en trois phases distinctes (Héon, 2010 ; Héon et Paquette, 2010): une phase *d'élicitation* de la connaissance dans un langage de degré semi-formel relativement évolué, une phase de *formalisation* des connaissances où le modèle semi-formel est transformé dans un langage ontologique, puis une phase de *validation* permettant d'harmoniser la sémantique du modèle semi-formel issue de l'élicitation avec la sémantique formelle de l'ontologie produite.

Schématisé à la figure 3 la méthodologie de conception d'une ontologie formelle à partir d'une élicitation semi-formelle se répartie en trois phases: *concevoir un modèle semi-formel*, *formaliser en ontologie du domaine* et *valider l'ontologie du domaine* produisant le *modèle semi-formel du*

domaine et l'ontologie du domaine. La méthodologie est itérative, c'est-à-dire que le résultat de l'itération précédente sert d'intrant à l'itération en cours de réalisation.

Cinq acteurs interviennent. Le premier est l'*expert de contenu*. Son rôle principal est d'extérioriser l'expertise qu'il détient par la production d'un modèle semi-formel du domaine valide et cohérent face à son domaine d'expertise. Il est donc directement impliqué dans la réalisation des méthodes de conception du modèle semi-formel du domaine et dans la méthode de validation de l'ontologie du domaine.

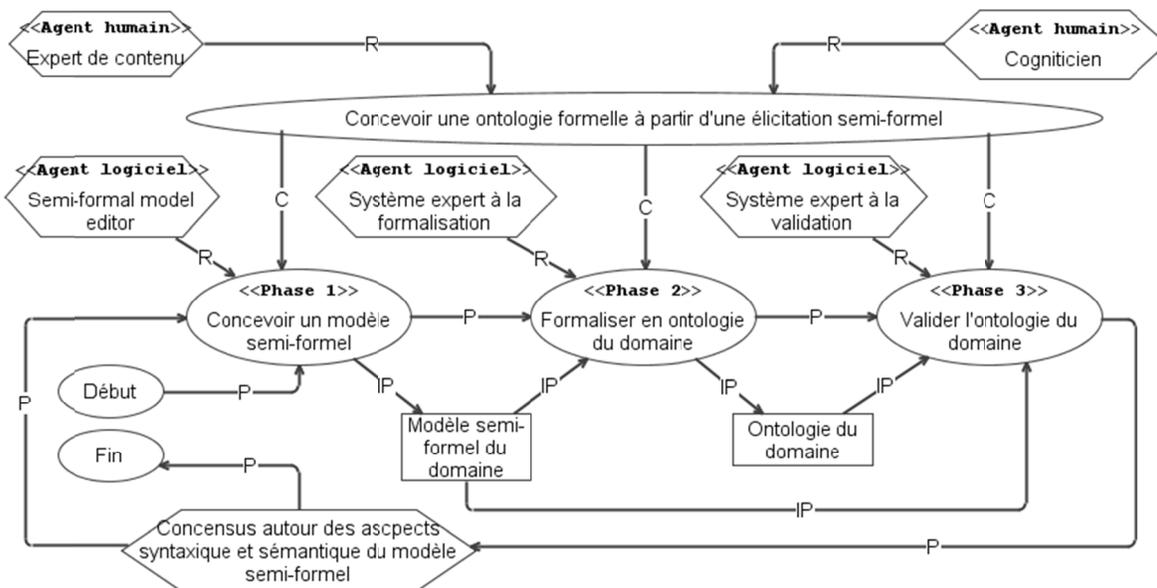


Figure 3. Méthode incrémentale de conception d'une ontologie à partir d'une élicitation semi-formelle

Le deuxième acteur, le *cogniticien*, a un rôle de support à la formalisation auprès de l'expert de contenu. Le cogniticien guide l'expert pendant la conception du modèle semi-formel; il réalise l'exécution de la méthode de formalisation, puis il travaille en collaboration avec l'expert pour la validation la sémantique de l'ontologie. Les troisième, quatrième et cinquième acteurs sont les outils qui forment le volet computationnel d'OntoCASE, soit l'éditeur de modèle MOT eLi, le système expert à la formalisation et le système expert à la validation.

2.1. Phase 1 L'élicitation

La modélisation, en vue de concevoir un système à base de connaissances ou une ontologie, passe par les étapes d'identification, d'explicitation, de structuration et de représentation de la

connaissance avant qu'elle ne soit utilisée pour la conception du système à base de connaissances (Hart, 1986). À sa plus simple expression, la modélisation est assumée par un cogniticien et un expert de domaine. Une méthode particulièrement prometteuse appelée *comodélisation* (Basque *et al.*, 2004) est proposée dans le cadre de la méthodologie d'OntoCASE. Cette méthode implique la participation d'un ou plusieurs experts et éventuellement de personnes plus novices¹ dans le domaine de connaissances ciblé, qui sont dirigées par le cogniticien. La *comodélisation* vise à stimuler l'externalisation des connaissances par la mise en présence de plusieurs intervenants (cogniticien, expert de contenu et novice) dans l'activité de modélisation. Les échanges entre les acteurs et les accords qu'ils établissent entre eux permettent la modélisation de connaissances consensuelles. Les connaissances consensuelles ainsi exprimées sont représentées dans un modèle semi-formel graphique, qui, par la suite, sert de nouvelle base de discussion pour la production de nouvelles connaissances consensuelles. Le cycle d'expression et de modélisation se poursuit jusqu'à ce que les acteurs humains décident par consensus de la complétude du modèle.

2.2. Phase 2 La formalisation

La méthode *formaliser en ontologie du domaine* (schématisée à la figure 4) est contrôlée par le cogniticien. Trois processus composent la méthode, soit (1) *importer dans l'espace de modélisation ontologique*, (2) *désambiguïser* et (3) *convertir en ontologie du domaine*.

Le processus d'importation « Étape 2.1 » sert à la traduction d'un modèle semi-formel employant le formalisme de l'éditeur de modèles semi-formels (par exemple le *XML Metada Interchange (XMI)*) vers un modèle employant le formalisme ontologique utilisé par le modèle cible (par exemple le *OWL*). Il s'agit d'une représentation en *OWL* du modèle source.

Le processus de désambiguïstation « Étape 2.2 », qui dans certains cas est assisté par l'expert de contenu, consiste à supprimer les ambiguïtés contenues dans *l'ontologie du modèle semi-formel* afin de produire *l'ontologie du modèle semi-formel désambiguïsé*. Pour ce faire, le processus de désambiguïstation utilise la sémantique du langage source pour identifier le type de désambiguïstation appropriée du modèle semi-formel. Trois catégories de désambiguïstation sont réalisées : *typologique*, *topologique* et *sémantique*. La *désambiguïstation typologique* est la plus

¹ On entend par novice toutes personnes en connaissance d'un domaine sans en être un expert. Par exemple, un nouvel employé qui vient tout juste de terminer ses études pourrait être considéré comme étant un novice.

automatique des désambiguïisations. Elle associe le type d'un composant du modèle semi-formel à un élément ontologique. Par exemple, un LienS sera directement traduit en `owl:subClassOf`. La *désambiguïisation topologique* est plus complexe et peut, dans quelques cas, n'être que semi-automatisée. Cette désambiguïisation s'établit en caractérisant les composants du modèle par l'identification d'un patron de disposition. Du point de vue de l'ingénieur de la connaissance, l'étape de *désambiguïisation selon la sémantique du domaine* est délicate, car elle nécessite une compréhension du domaine qui est modélisé, ce qui implique la collaboration de l'expert de contenu afin de répondre aux questions du cogniticien.

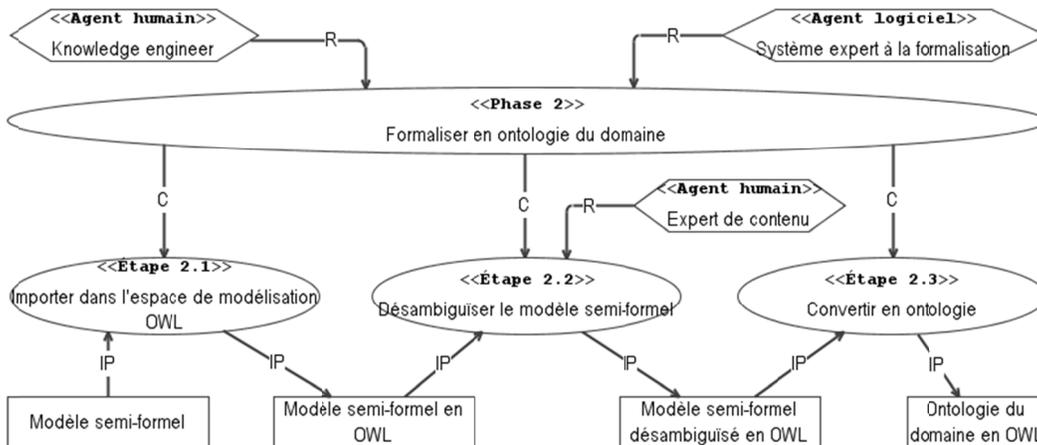


Figure 4. Méthode de formalisation d'un modèle semi-formel en ontologie du domaine

Finalement, à partir de l'ontologie du modèle semi-formel désambiguïisé, le *processus de conversion en ontologie du domaine* « Étape 2.3 » produit une nouvelle ontologie du domaine en interprétant chacun des éléments de l'ontologie désambiguïisée en élément correspondant dans l'ontologie du domaine. Par exemple, un concept dans le modèle semi-formel sera transformé en classe dans l'ontologie du domaine.

2.3. Phase 3 La validation

En ingénierie ontologique, la validation est une étape essentielle à la production d'une ontologie de domaine. Gómez-Pérez *et al.* (2003) présentent les critères selon lesquels l'ontologie devrait être validée. La *consistance* concerne l'absence de contradictions entre les éléments ontologiques. La *complétude* assure que tous les éléments ontologiques sont soit explicitement déclarés, soit inférables. La *concision* est le principe qui stipule que seuls les éléments à être définis doivent l'être. L'*expansibilité* est la capacité d'ajouter de nouvelles connaissances sans

modifier les anciennes. Finalement, la *sensibilité* est la capacité de l'ontologie à réagir à des modifications. L'application des critères de validation assiste le cogniticien dans la validation de l'ontologie. Pendant la formalisation d'un modèle semi-formel, certaines opérations peuvent modifier des éléments de vocabulaire (le critère syntaxique) ou altérer le sens de la représentation (le critère sémantique). Il importe donc d'implanter un mécanisme de validation qui assure qu'aucune altération de la syntaxe et de la sémantique du modèle ne soit induite par le processus de transformation.

Le but de la *validation syntaxique* est de fournir au cogniticien un mécanisme de détection des erreurs informatiques qui peuvent survenir pendant la production de l'ontologie du domaine. Comme son nom l'indique, la validation syntaxique concentre les efforts sur l'inspection des composants du vocabulaire et de la grammaire du modèle semi-formel. L'idée soutenant ce processus est que tous les éléments et relations du modèle d'origine doivent être représentés dans l'ontologie du domaine. Ainsi un modèle reconstruit à partir de l'ontologie du domaine devrait être identique au modèle d'origine en matière d'entités et de relations. Le sous-processus, de *génération d'un modèle semi-formel* produit un modèle semi-formel de domaine qui est reconstruit à partir de l'ontologie du domaine. Le sous-processus de comparaison du modèle semi-formel d'origine avec le modèle semi-formel reconstruit produit un rapport de validation syntaxique qui exprime les éventuelles anomalies syntaxiques entre les modèles. Le rapport de validation sert de guide au cogniticien pour apporter les ajustements nécessaires aux règles de transformation.

La *validation sémantique* est le sous-processus de la validation qui s'intéresse à la préservation de la signification du modèle (le modèle représente-t-il bien la réalité qu'il désigne?). À l'issue de la validation sémantique, un rapport de validation est produit de la manière suivante: l'expert produit une *interprétation du modèle semi-formel*. Le cogniticien produit, grâce à OntoCASE une *interprétation automatique de l'ontologie du domaine*. Pour ce faire, OntoCASE fait appel à un moteur d'inférence pour générer de nouvelles conclusions qui serviront d'intrants à la *comparaison des interprétations* pour la production du rapport d'interprétation. C'est sur la base de ce rapport que la décision de poursuivre ou non la démarche d'élicitation par une nouvelle itération de formalisation sera prise.

3. OUTIL ONTOCASE

La figure 5 présente l'interface utilisateur de l'application OntoCASE. L'édition des modèles semi-formels et des ontologies est assuré par deux modules : le module d'édition de modèles semi-formels « eLi » qui est une application *Eclipse* de type EMF (Eclipse Foundation, 2009b) et GMF (Eclipse Foundation, 2009a) et le module d'édition d'ontologies OWL qui est assuré par l'intégration des plug-ins *TopBraid Composer™ free edition* (TopQuadrant, 2009) à l'environnement d'OntoCASE.

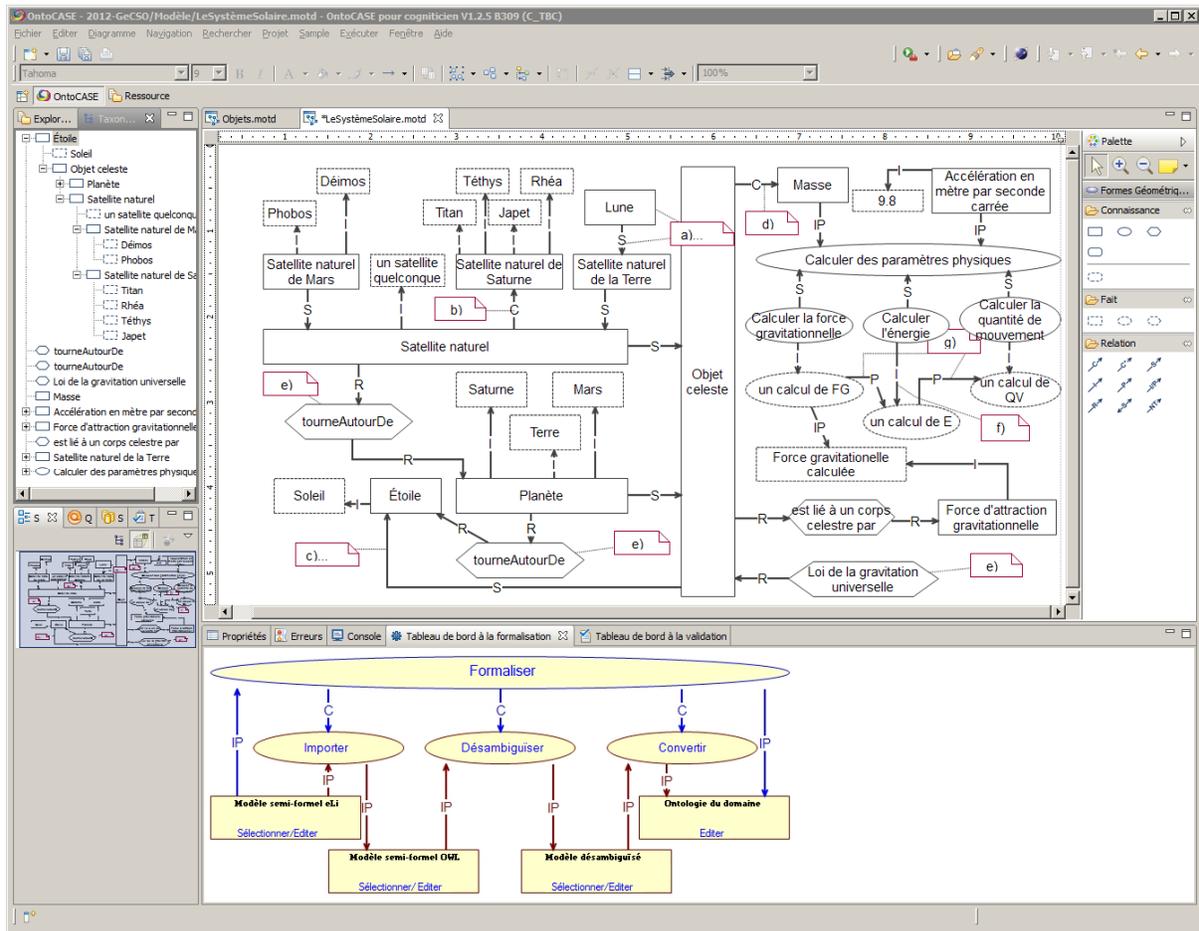


Figure 5. Interface utilisateur de l'application OntoCASE

Au centre de la moitié supérieure de la figure 5, se trouve le schéma du modèle semi-formel, alors qu'à gauche, la vue en graphe de composites en présente la taxonomie. La moitié inférieure droite présente un tableau de bord (sous forme d'un modèle procédural) permettant au cognicien de lancer les processus nécessaires à la formalisation (importer, désambiguiser et convertir) et d'éditer au besoin chaque modèle après le déclenchement du processus. Sont aussi accessibles :

une vue permettant le contrôle de la validation, une console de commandes exécutées, un volet des propriétés des objets ainsi qu'un calepin des messages d'erreurs. Dans la section inférieure gauche, l'utilisateur a accès à des utilitaires tels qu'un calepin de tâches à faire, une structure SVN (Apache Software Foundation, 2001) pour le contrôle de version des ontologies, une vue de manipulation du serveur Web ainsi qu'une vue de structure facilitant la navigation dans de grands modèles.

Le module de formalisation dont l'interface est un tableau de bord (voir la figure 6) est un module qui assiste le cogniticien dans la tâche de formalisation du modèle semi-formel en ontologie. Les trois étapes d'importation, de désambiguïsation et de transformation qui composent la réalisation de cette tâche y sont représentées. À chacune des étapes, le cogniticien a l'opportunité d'éditer l'ontologie résultante.

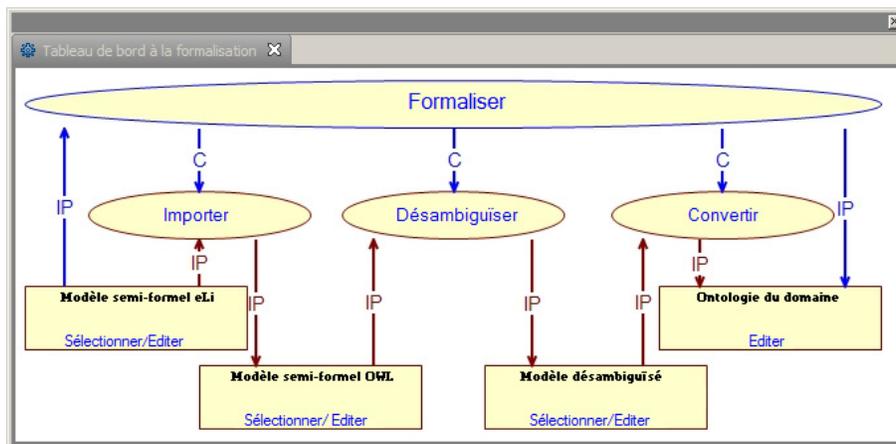


Figure 6. Tableau de bord à la formalisation disponible pour assister le cogniticien

Le module de validation (voir la figure 7) permet à l'ingénieur et à l'expert de contenu d'accéder aux outils nécessaires à la réalisation d'une validation syntaxique et sémantique. Ainsi, les acteurs de cette activité ont accès à des outils permettant la production de conclusions automatiques et nécessaires à l'évaluation sémantique de l'ontologie. Ils ont aussi accès à des outils de génération d'un modèle semi-formel à partir de l'ontologie du domaine ainsi qu'à des outils de comparaison entre le modèle d'origine et le modèle généré afin d'assumer la validation syntaxique de l'ontologie du domaine. L'appel à ces outils est assuré par le déploiement d'un menu contextualisé qui offre aux acteurs l'opportunité de sélectionner l'outil approprié à la réalisation de la tâche en cours.

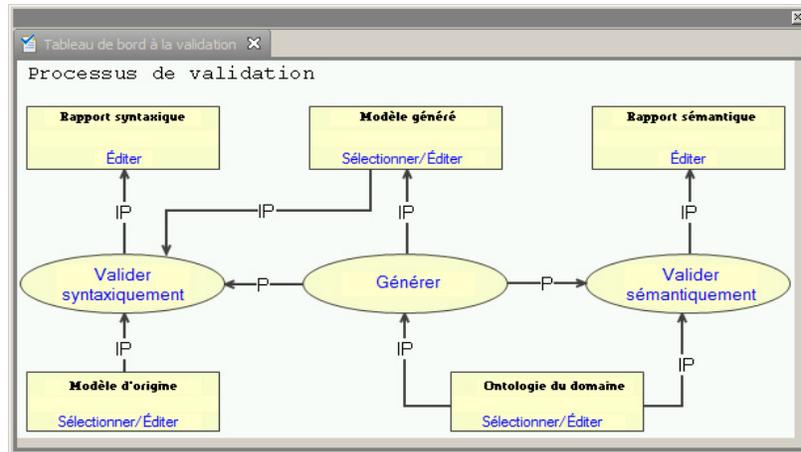


Figure 7. Tableau de bord à la validation

Les systèmes experts à la formalisation et à la validation sont des applications *Java V1.5* qui utilisent la bibliothèque de codes de *Protégé* version 3.4.4. Cette bibliothèque fournit une interface au moteur *Pellet* (Clark & Parsia, 2009) pour le raisonnement logique et au moteur *Jess* (Sandia National Laboratories, 2008) pour le raisonnement à base de règles. Le *Semantic Web Rule Language* (SWRL) est un langage à base de règle structuré en *antécédent* qui implique un *conséquent*. L'édition des règles SWRL est réalisée grâce aux fonctionnalités du *SWRLTab*². *Protégé* offre aussi une bibliothèque d'implantation *Java* dont le *SWRLBuitlinBridge* (O'Connor, 2009), qui permet d'associer une commande *SWRL-BuiltIn* (Horrocks *et al.*, 2004) à son implantation *Java*. Comme son nom l'indique, le *SWRLBuitlinBridge* est un mécanisme qui sert de pont entre l'atome SWRL et son implantation en *Java*. Le détail de cette implémentation est traité dans Héon (2010, 2011).

4. EXPÉRIMENTATION AVEC UN MODÈLE COMPLEXE

Dans cette section, nous illustrons un exemple d'utilisation d'OntoCASE au moyen du traitement d'un modèle complexe contenant des connaissances déclaratives, procédurales et stratégiques, abstraites et factuelles en vue de les formaliser dans l'ontologie.

4.1. Étape 1: Production d'un modèle semi-formel de connaissances

Cette étape permet d'élaborer un modèle de connaissances semi-formel. Aux fins d'illustrer le processus d'assistance et sa capacité de détection des erreurs de nature sémantique, nous avons

² <http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTab>

introduit dans le modèle (présenté à la figure 8) différents types d'erreurs ou d'imprécisions sémantiques. Les erreurs de modélisation introduites dans le modèle sont inspirées du concept d'antipattern proposé par Allemang *et al.*(2008). Par exemple, l'inversion du LienS, la confusion entre la subsomption et la composition ou encore, le niveau de granularité à atteindre pour représenter une réalité dans un modèle. Les étiquettes dans la figure désignent les points de discussion qui seront traités ultérieurement dans cette section.

Voici en quelques lignes la pensée qui aurait pu conduire à une telle modélisation. *Le corps céleste est régi par le principe de la loi de la gravitation universelle. Il a pour attribut une masse et est lié à d'autres corps célestes par la force d'attraction gravitationnelle(FG). Cette force est calculée à partir de la masse et de l'accélération gravitationnelle (G) qui est de 9.8 m/s². D'autres paramètres physiques peuvent être calculés tels que l'énergie (E) et la quantité de mouvement (QV). Un calcul de la FG suivi d'un calcul de E et d'un calcul de QV sont des exemples spécifiques de calcul de paramètres physiques. Les étoiles, les planètes et les satellites naturels sont des corps célestes. Les planètes tournent autour des étoiles et les satellites naturels tournent autour des planètes. Saturne, Mars et la Terre sont des planètes, Déimos, Phobos sont des satellites naturels de Mars; Titan, Téthys, Japet et Rhéa sont des satellites naturels de Saturne. Finalement, la Lune est un satellite naturel de la Terre. Il existe aussi, quelque part, un satellite quelconque.*

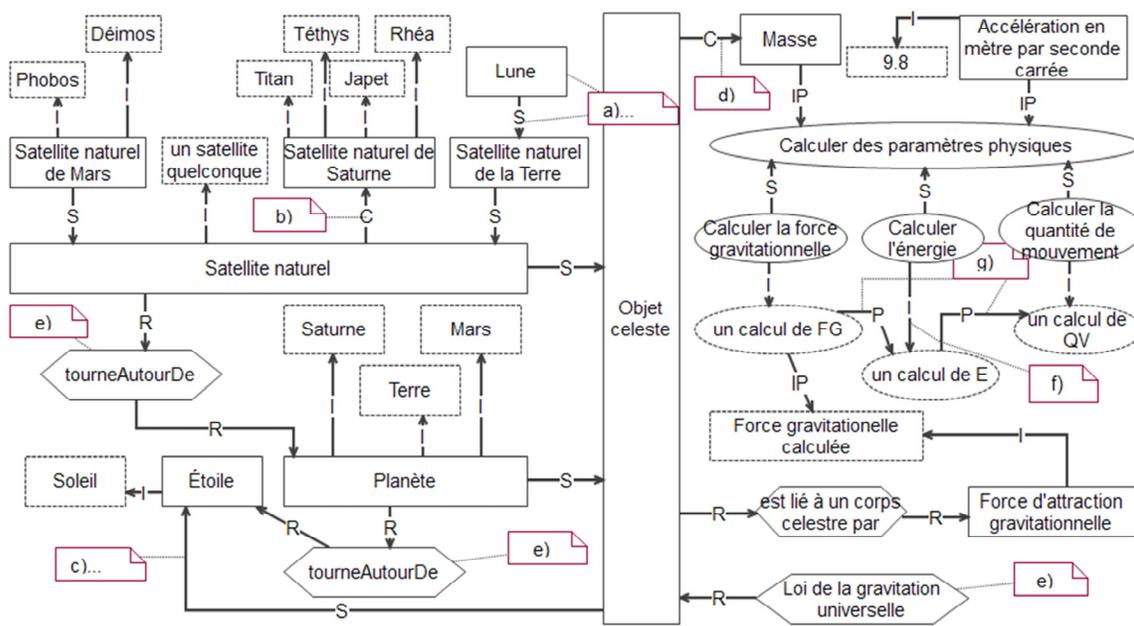


Figure 8. Exemple de modèle semi-formel à formaliser

Avant d'aller de l'avant dans la formalisation du modèle expérimental, nous désirons souligner quelques ambiguïtés ou imprécisions concernant le modèle présenté à la figure 8. L'étiquette **a** illustre un exemple d'erreur de modélisation couramment observée. Il s'agit d'une erreur de classification du niveau d'abstraction des "choses" représentées. Il arrive qu'un modélisateur confonde la représentation d'un objet concret avec la classe abstraite à laquelle appartient l'objet. Il est vrai que le concept de *Lune* est une sorte de *Satellite naturel de la Terre*, mais ce qui est représenté ici devrait plutôt être l'objet de type *Lune* qui est un exemple du concept *satellites naturels de la Terre*, et non une sous-classe de ce concept. Cette erreur affecte la concision et la sensibilité de l'ontologie (selon la définition donnée à la section 3.3).

L'étiquette **b** illustre une autre erreur de représentation, soit la confusion dans l'utilisation du LienS et du LienC. Dans le modèle, on affirme que les *Satellites naturels* se composent de *Satellite naturel de Saturne*. Pour exprimer la relation entre ces deux concepts, il s'agit plutôt d'une relation de spécialisation. Nous verrons à l'étape de validation comment cette erreur pourra être relevée et présentée à l'expert. Cette erreur affecte la consistance et complétude de l'ontologie.

Finalement, une autre erreur commise est l'inversion de l'orientation du LienS (voir l'étiquette **c**). Étant donné la transitivité du lien S, on devrait en conclure que tous les *satellites* et toutes les *planètes* sont des *étoiles*. Nous en verrons les conséquences à l'étape de validation. Les étiquettes **d** à **g** soulignent, quant à elles, des points de précisions qui seront traités plus loin. Cette erreur affecte la consistance de l'ontologie.

4.2. Étape 2: Formalisation

La formalisation est l'étape de construction de l'ontologie du domaine. Après avoir importé le modèle semi-formel dans l'espace de modélisation ontologique, le cognicien désambiguïse les éléments du modèle. Certaines entités (comme le LienI) sont univoques, c'est-à-dire qu'une entité semi-formelle correspond à une seule entité formelle. Il s'agit alors d'une désambiguïstation automatique de type typologique.

D'autres ambiguïtés nécessitent un traitement automatique basé sur des patrons de modélisation de type topologique. Par exemple, les principes indiqués par l'étiquette **e** subissent ce type de désambiguïstation puisqu'un principe entre deux concepts est interprété comme une `owl:ObjectProperty` alors qu'un principe lié à un concept par un LienR sera interprété en tant

que `owl:Class` de catégorie `AGENT_CONTRAINTE_NORME`. Il en va de même pour les liens IP qui sont désambiguïsés en `owl:ObjectProperty` de catégorie `INTRANT` ou `PRODUIT` selon l'orientation des liens.

Lorsqu'un objet ne peut être désambiguïsé de façon typologique ou topologique, alors il doit l'être de façon manuelle selon la sémantique du domaine. Dans l'exemple, le LienC de l'étiquette **b** et celui de l'étiquette **d** n'ont pas la même signification. Dans le premier cas, il est exprimé comme une relation d'agrégation (les *satellites naturels* se composent de *satellite de la Terre*), alors que celui de l'étiquette **d** s'interprète comme une relation d'attribution : le *corps céleste* a pour attribut une *masse*.

La transformation en ontologie du domaine est le dernier processus de l'étape de formalisation. Étant donné que les connaissances du modèle semi-formel intègrent des connaissances procédurales et stratégiques, il est nécessaire de procéder à une transformation qui mènera à une représentation de ces connaissances selon un format déclaratif dans l'ontologie du domaine. L'ontologie de référence permet de regrouper les connaissances formalisées sous trois grandes catégories: `MD_Declarative`, `MD_Procedurale` et `MD_Strategique`. Les propriétés du domaine sont regroupées sous six groupes de méta-propriétés soit: `A-POUR-COMPOSANT`, `A-POUR-DÉPENDANCE`, `EST-LA-PARTIE-DE`, `INTRANT-PRODUIT`, `PERMET-DE`, `A-POUR-ATTRIBUT`, `REGIT` et `EST-REGI-PAR`. Certaines méta-propriétés sont les inverses les unes des autres; par exemple, `A-POUR-COMPOSANT` est une propriété inverse de `EST-LA-PARTIE-DE`. Nous verrons à l'étape de validation que cette attribution de propriété inverse sera fort utile pour l'interprétation des connaissances de l'ontologie.

4.3. Étape 3: Validation

La validation syntaxique a pour objectif de s'assurer que chaque élément du modèle semi-formel retrouve sa représentation dans l'ontologie du domaine. Pour mesurer cette concordance, un modèle semi-formel est généré à partir de l'ontologie du domaine. Le décompte des éléments de chacun des modèles et la comparaison des résultats permettent de vérifier la représentation de chaque élément du modèle semi-formel dans l'ontologie du domaine. Ainsi, pour le modèle de la figure 8, le système identifie que le modèle se compose de 11 concepts, 4 procédures, 4 principes, 13 exemples, 3 traces, 3 lienIP, 7 lienR, 5 lienC, 9 lienS, 16 lienI, 2 lienP et zéro lienA, LienCm, et énoncés.

L'idée maîtresse de la validation sémantique est de comparer les conclusions inférées humainement à partir du modèle semi-formel avec les conclusions inférées automatiquement à partir de l'ontologie du domaine (voir la figure 9) La première phase de la validation est l'analyse de l'ontologie. Dans la taxonomie de cette ontologie, le concept d'*Objet-céleste* est subsumé par celui d'*Étoile* (voir la ligne 03), ce qui voudrait dire que la Terre, qui est une sorte de Planète, serait considérée comme une Étoile (voir la ligne 20). Cette erreur se répète aussi pour les lignes 11, 15, 18 et 33. Cette erreur de classification provient de l'inversion de la direction du LienS Dans le cas de figure représenté par l'étiquette **b**, on pourrait interpréter que *Phobos*, en tant que *satellite naturel* se compose d'un *satellite naturel de Saturne* tel que *Titan*. Or, une inférence automatique nous indiquera que *Titan* EST-LA-PARTIE-DE d'un *satellite naturel* alors qu'il est

-----interprétation partielle avant inférence-----	
01.	Objet celeste a pour attribut Masse
02.	Objet celeste est lié à un corps celestre par Force d'attraction gravitationnelle
03.	Objet celeste est une sorte de Étoile
04.	Planète est une sorte de Objet celeste
05.	Planète tourneAutourDe Étoile
06.	Satellite naturel de Mars est une sorte de Satellite naturel
07.	Satellite naturel de Saturne est une sorte de (metaDom:MD_Declarative_Concept)
08.	Satellite naturel de la Terre est une sorte de Satellite naturel
09.	Satellite naturel est une sorte de Objet celeste
-----intepretation partielle après inférence-----	
10.	[9.8] est un (:Accélération en mètre par seconde carrée (G))
11.	[Déimos] est un (:Satellite naturel de Mars :Étoile :Objet celeste :Satellite naturel)
12.	[Force gravitationnelle calculée] est le produit de [un calcul de FG]
13.	[Force gravitationnelle calculée] est un (:Force d'attraction gravitationnelle)
14.	[Japet] est un (:Satellite naturel de Saturne)
15.	[Mars] est un (:Planète :Étoile :Objet celeste)
16.	[Phobos] est un (:Satellite naturel de Mars :Étoile :Objet celeste :Satellite naturel)
17.	[Rhéa] est un (:Satellite naturel de Saturne)
18.	[Saturne] est un (:Planète :Étoile :Objet celeste)
19.	[Soleil] est un (:Étoile)
20.	[Terre] est un (:Planète :Étoile :Objet celeste)
21.	[Titan] est un (:Satellite naturel de Saturne)
22.	[Téthys] est un (:Satellite naturel de Saturne)
23.	[un calcul de E] a pour dépendance [un calcul de FG]
24.	[un calcul de E] est un (:Calculer l'énergie :Calculer des paramètres physiques)
25.	[un calcul de E] permet [un calcul de QV]
26.	[un calcul de FG] est un (:Calculer la force gravitationnelle :Calculer des paramètres physiques)
27.	[un calcul de FG] permet [un calcul de E]
28.	[un calcul de FG] permet [un calcul de QV]
29.	[un calcul de FG] puis exécuter [un calcul de QV]
30.	[un calcul de QV] a pour dépendance [un calcul de E]
31.	[un calcul de QV] a pour dépendance [un calcul de FG]
32.	[un calcul de QV] est un (:Calculer la quantité de mouvement :Calculer des paramètres physiques)
33.	[un satellite quelconque] est un (:Satellite naturel :Étoile :Objet celeste)

Figure 9. Exemple d'interprétation automatique à partir du modèle semi-formel

plus exact de dire que *Titan* et *Phobos* sont tous les deux des *satellites naturels*.

La formalisation du cas de figure présenté à l'étiquette **a**, indique que la classe *Lune* est une sorte de *satellite naturel* de la *Terre*. Mécaniquement, nous pourrions instancier des individus de type *Lune*, ce qui, en termes représentationnels, est faux puisque la *Lune* est un objet et non une classe d'objets.

L'étiquette **d** présente le cas où un lien C est formalisé en tant qu'attribut. Ce lien sera désambiguïté grâce à la distinction entre les métaliens A-POUR-COMPOSANT et A-POUR-ATTRIBUT.

L'étiquette **e** présente deux cas d'utilisation du mot:Principe. Le premier des cas, *Loi de la gravitation universelle*, est désambiguïté sous la forme owl:Class de catégorie Agent_Contrainte_Norme. Le deuxième cas, *tourneAutourDe*, est formalisé en owl:ObjectProperty dont le domaine et l'image correspondent à la classe source et cible du Principe.

L'étiquette **f** indique que les entités *un calcul de FG*, *un calcul de E* et *un calcul de QV*, sont des traces des connaissances procédurales *calculer la force gravitationnelle*, *calculer l'énergie* et *calculer la quantité de mouvement*. L'étiquette **g** présente un cas de figure fort intéressant puisqu'il concerne le traitement ontologique de connaissances procédurales. Après l'inférence, grâce aux méta-propriétés définies dans l'ontologie de référence, on déduira que le calcul de la quantité de mouvement A-POUR-DÉPENDANCE l'exécution des étapes *calculer la force gravitationnelle* et *calculer l'énergie*, dans l'ordre. (étiquette **c** de la figure 8). Le moteur d'inférence appliqué à l'ontologie fera apparaître automatiquement cette incohérence, ce qui pourrait amener à corriger le modèle semi-formel initial.

5. CONCLUSION

Dans cet article, nous avons présenté OntoCASE une méthodologie dédiée à la conception d'ontologies et instrumentée d'un assistant logiciel. Nous avons présenté la structure du langage MOT qui est utilisé par l'éditeur eLi, composante d'OntoCASE. Nous avons aussi traité des aspects méthodologique et informatique d'OntoCASE pour finir par la présentation d'un exemple de formalisation d'un modèle semi-formel en ontologie de domaine.

Dans le domaine de la représentation des connaissances, certains éléments d'expressivité des ontologies ou d'autres types de langages ne peuvent pas être représentés dans le langage MOT. Par exemple, la propriété owl:inverseOf n'a pas de correspondant en langage MOT. Certaines

fonctionnalités de "stéréotypage" pourraient être implantées dans OntoCASE pour permettre la représentation de l'ensemble des éléments d'expressivité d'OWL à partir de l'édition semi-formelle de la conceptualisation d'origine.

En gestion des connaissances notamment en élicitation de connaissances, la capacité que possède OntoCASE de produire une interprétation objective et normalisée en langage naturel d'un modèle semi-formel pourrait être exploitée pour stimuler la créativité des experts à exprimer leurs connaissances. De plus, la simplicité d'utilisation du langage MOT permet à des experts de contenu de construire une mémoire organisationnelle semi-formelle sans l'intervention d'un cognicien. La mémoire semi-formelle peut ensuite être réutilisée pour être formalisée en ontologie avec OntoCASE sans le recours à l'expert de contenu. Cet avantage procure de la souplesse pour la coordination de projets de construction de systèmes à base de connaissances.

En considérant OntoCASE comme un Environnement Informatisé pour l'Apprentissage Humaine (EIAH) pour l'apprentissage de la modélisation, les dispositifs de rétroaction (les menus contextuels qui limitent le choix des objets de modélisation à des possibilités valides, les messages liés à la désambiguïsation ou à la gestion des erreurs de modélisation et l'utilisation des outils de validation syntaxique et sémantique), offrent à l'apprenant en modélisation un outil efficace d'interprétation automatique du modèle semi-formel. L'interprétation automatique de modèle à travers un processus de désambiguïsation et de formalisation ontologique procure les avantages suivants: elle guide l'activité de modélisation pendant l'étape d'explicitation de la connaissance; elle permet une auto-évaluation de la démarche de modélisation; elle permet une validation objective du modèle conçu. L'usage d'OntoCASE en tant qu'EIAH pourrait avoir des effets sur la conceptualisation interne (mentale) des sujets, mais cela serait à vérifier avec des recherches faisant appel à des mesures cognitives.

BIBLIOGRAPHIE

- Allemang, Dean, et Jim Hendler. 2008. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Coll. «Semantic Web / Web programming». Burlington MA: Morgan Kaufmann, 330 p.
- Apache Software Foundation. 2001. «Subversion: Open Source Software Engineering Tools». Tigris.org. En ligne. <<http://subversion.apache.org/>>. Consulté le 1 avril 2012.
- Baader, Franz, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi et Peter F. Patel-Schneider. 2007. *The Description Logic Handbook*: Cambridge University Press.

- Basque, Josianne, Clément Imbeault, Béatrice Pudelko et Michel Léonard. 2004 «Collaborative knowledge modeling between experts and novices: A strategy to support transfer of expertise in a organization», *Conference on Concept Mapping* (September 14-17). J.D. Novak A.J. Canas, F.M. Gonzalez Pamplona, pp. 75-81 p.
- Basque, Josianne, Gilbert Paquette, Béatrice Pudelko et Michel Léonard. 2008. «Collaborative Knowledge Modeling with a Graphical Knowledge Representation Tool: A Strategy to Support the Transfer of Expertise in Organizations». Dans *Knowledge Cartography. Mapping Techniques and Software Tools*, Alexandra Okada, Simon Buckingham Shum et Tony Sherborne, p. 357-382. London: Springer-Verlag.
- Basque, Josianne, et Béatrice Pudelko. 2010a. «Intersubjective Meaning-Making in Dyads Using Object-Typed Concept Mapping». Dans *In Handbook of Research on Collaborative Learning Using Concept Mapping*, P.L. Torres et R.C.V. Marriott: IGI Global.
- Basque, Josianne, et Béatrice Pudelko. 2010b. «Modeling for Learning». Dans *Visual Knowledge and Competency Modeling - From Informal Learning Models to Semantic Web Ontologies*, Gilbert Paquette. Hershey, New York: IGI Global.
- Buzan, Tony, et Barry Buzan. 1994. *The Mind Map Book: How to Use Radiant Thinking to Maximize Your Brain's Untapped Potential*: E P Dutton.
- Castro, Alexander, Philippe Rocca-Serra, Robert Stevens, Chris Taylor, Karim Nashar, Mark Ragan et Susanna-Assunta Sansone. 2006. «The use of concept maps during knowledge elicitation in ontology development processes - the nutrigenomics use case». *BMC Bioinformatics*. vol. 7, no 1, p. 267. En ligne. <<http://www.biomedcentral.com/1471-2105/7/267>>.
- Chen, Peter Pin-shan. 1976. «The Entity-Relationship Model: Toward a Unified View of Data». *ACM Transactions on Database Systems*. vol. 1, p. 9-36.
- Clark & Parsia. 2009. «Pellet: The Open Source OWL Reasoner». Clark & Parsia. En ligne. <<http://clarkparsia.com/pellet>>. Consulté le 2009.
- Eclipse Foundation. 2009a. «Eclipse Graphical Modeling Framework (GMF)». The Eclipse Foundation. En ligne. <<http://www.eclipse.org/modeling/gmf>>.
- Eclipse Foundation. 2009b. «Eclipse Modeling Framework Project (EMF)». The Eclipse Foundation. En ligne. <<http://www.eclipse.org/modeling/emf/>>.
- Eskridge, Thomas, Pat Hayes et Robert Hoffman. 2006 «Formalizing The Informal: A Confluence Of Concept Mapping And The Semantic Web», *Second Int. Conference on Concept Mapping* (San José, Costa Rica). A. J. Cañas et J. D. Novak.
- Gašević, Dragan, Dragan Djurić et Vladan Devedžić. 2006. *Model Driven Architecture and Ontology Development*. New York, Inc.: Springer-Verlag. En ligne. <<http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/3540321802>>.
- Gómez-Pérez, Asunción, Mariano Fernández-López et Oscar Corcho. 2003. *Ontological Engineering : with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*, First edition. New York: Springer.
- Gruber, Thomas. 1993. «Toward Principles for the Design of Ontologies Used for Knowledge Sharing». *Formal Ontology in Conceptual Analysis and Knowledge Representation*, p. 23.
- Hart, Anna. 1986. *Knowledge acquisition for expert systems*. New York: McGraw-Hill.
- Héon, Michel. 2010. «OntoCASE: Méthodologie et assistant logiciel pour une ingénierie ontologique fondée sur la transformation d'un modèle semi-formel». Thèse, Montréal, Informatique Cognitive,

- Université du Québec à Montréal, 419 p. En ligne. <http://hal.archives-ouvertes.fr/tel-00568936_v1/>.
- Héon, Michel. 2011. «Implantation d'un moteur de transformation intelligent dans le contexte de l'Architecture Conduite par les Ontologies». Dans *4èmes Journées Francophones sur les Ontologies* (22 – 23 Juin 2011).
- Héon, Michel. 2012. «OntoCASE un outil CASE d'élicitation et de conception d'une ontologie OWL». Cotechnoe inc. En ligne. <<http://www.cotechnoe.com/ontocase/>>. Consulté le 20 février 2012.
- Héon, Michel, et Gilbert Paquette. 2010. «From semi-formal Models to Formal Models». Dans *Visual Knowledge and Competency Modeling - From Informal Learning Models to Semantic Web Ontologies*, Gilbert Paquette. Hershey, New York: IGI Global.
- Horrocks, Ian, Harold Boley, Said Tabet, Benjamin Grosf et Mike Dean. 2004. «SWRL: A Semantic Web Rule Language Combining OWL and RuleML». W3C. En ligne. <<http://www.w3.org/Submission/SWRL/>>. Consulté le 29 mai 2010.
- Kinshuk, Heimo H. Adelsberger, Jan Martin Pawlowski et Demetrios Sampson. 2008. *Handbook for Education and Training on Information Technologies*, Second edition. Coll. «International Handbooks on Information Systems». Heidelberg: Springer-Verlag.
- Novak, Joseph D, et Alberto J Cañas. 2006. «The origins of the concept mapping tool and the continuing evolution of the tool». *Information Visualization*. vol. 5, p. 175–184.
- O'Connor, Martin. 2009. «SWRLBuilt In Bridge». Stanford Center for Biomedical Informatics Research. En ligne. <<http://protege.cim3.net/cgi-bin/wiki.pl?SWRLBuiltInBridge>>.
- Paquette, Gilbert. 2002. *L'ingénierie pédagogique*. Sainte-Foy (Québec): Presses de l'Université du Québec.
- Paquette, Gilbert. 2010. *Visual Knowledge and Competency Modeling - From Informal Learning Models to Semantic Web Ontologies*. Hershey, PA: IGI Global.
- Park, Jack, et Sam Hunting. 2003. *XML Topic Maps: Creating and Using Topic Maps for the Web*. Addison Wesley, 605 p.
- Rumbaugh, James, Ivar Jackson et Grady Booch. 1999. *The Unified Modeling Language Reference Manual*. Coll. «Object Technology».
- Sandia National Laboratories. 2008. «Jess: the Rule Engine for the Java Platform». Sandia National Laboratories. En ligne. <<http://www.jessrules.com/jess/index.shtml>>. Consulté le 27 mars 2008.
- Staab, Steffen, Rudi Studer, Hans-Peter Schnurr et York Sure. 2001. «Knowledge Processes and Ontologies». *IEEE Intelligent Systems*. vol. 16, no 1, p. 26-34.
- Stanford Medical Informatics. 2006. «Protégé home page». Stanford Medical Informatics. En ligne. <<http://protege.stanford.edu/>>. Consulté le 29 Mai 2006.
- TopQuadrant. 2009. «TopBraid Composer (TM)». TopQuadrant. En ligne. <http://www.topquadrant.com/products/TB_Composer.html>. Consulté le 27 sept. 2009.
- Uschold, Mike, et Micheal Gruninger. 1996. «Ontologies: Principles, Methods and Applications». *Knowledge Engineering Review*. vol. 11, no 2, p. 93-136.
- W3C OWL. 2004. «Ontology Web Language». World Wide Web Consortium,,. En ligne. <<http://www.w3.org/2004/OWL/>>. Consulté le 30 oct. 2009.
- Weber, R., et R. Kaplan. 2002. «Knowledge-based knowledge management». Dans *Innovations in Knowledge Engineering*, C Faucher, L Jain et N Ichalkaranje. Heidelberg.