



**HAL**  
open science

## SéMo'07: premier atelier sur la Sémantique des Modèles

Benoit Combemale, Xavier Crégut, Marc Pantel, Pierre Michel

► **To cite this version:**

Benoit Combemale, Xavier Crégut, Marc Pantel, Pierre Michel. SéMo'07: premier atelier sur la Sémantique des Modèles. *Revue des Sciences et Technologies de l'Information - Série L'Objet : logiciel, bases de données, réseaux*, 2007, 13 (4), pp.137-144. 10.3166/obj.13.4.137-144 . hal-00369871

**HAL Id: hal-00369871**

**<https://hal.science/hal-00369871v1>**

Submitted on 22 Mar 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

## SéMo'07 : premier atelier sur la Sémantique des Modèles

**Benoît Combemale\*** — **Xavier Crégut\*** — **Marc Pantel\***  
**Pierre Michel\*\***

\* *IRIT CNRS, Université de Toulouse*  
2, rue Charles Camichel, BP 7122  
F-31071 Toulouse Cedex 7  
*prénom.nom@enseiht.fr*

\*\* *ONERA CERT*  
2, av. Edouard Belin - BP 4025  
F-31055 Toulouse Cedex 4  
*prénom.nom@cert.fr*

---

*RÉSUMÉ. Cet article résume les présentations et discussions qui ont eu lieu lors de l'atelier Sé-Mo'2007 à Toulouse en mars 2007 conjointement à la conférence IDM. Le thème principal de cet atelier était la sémantique des modèles et des méta-modèles dans l'objectif de confronter les besoins des utilisateurs de l'IDM et les technologies existantes pour les langages de spécification et programmation.*

*ABSTRACT. This paper summarises the presentations and exchanges that took place in the Sé-Mo'2007 workshop in March 2007 in Toulouse during the IDM conference. The main topic of the workshop was the semantics of models and meta-models in the purpose to confront the MDE users needs and the technologies used for specification and programming languages.*

*MOTS-CLÉS : IDM, Sémantiques des modèles et méta-modèles, SéMo'07*

*KEYWORDS: MDE, Model and meta-models Semantics, SéMo'07*

---

## 1. Introduction

L'atelier SéMo (<http://semo2007.enseeiht.fr>) a eu lieu l'après-midi du 29 mars 2007 à Toulouse, conjointement à la conférence IDM 2007. Son objectif était de réunir les communautés de l'Ingénierie Dirigée par les Modèles (IDM) et de la sémantique des Langages de Programmation (LP), dans le but d'échanger sur les techniques pour exprimer et exploiter la sémantique des LP, et d'élargir l'application de ces techniques aux Langages de Modélisation (LM). Il devient en effet important de disposer de moyens pour exprimer la sémantique comportementale des langages de modélisation, en particulier des langages utilisés pour décrire un système critique (temps-réel, embarqué, fiable, redondant, performant) dans le but de valider les solutions envisagées au plus tôt dans le processus de développement.

En Ingénierie Dirigée par les Modèles, un *modèle* est vu comme une abstraction d'un système pour répondre aux questions que l'on se pose sur ce système. Le modèle est exprimé dans un *Langage de Modélisation* (LM). Quand ce langage est dédié à un domaine particulier, on parle de *Domain Specific (Modeling) Language* (DSL). Les techniques IDM s'appuient sur le *méta-modèle* d'un LM, c'est-à-dire les constructions qui composent ce langage et les relations entre ces constructions, mais offrent peu de moyens généraux pour associer une interprétation sémantique à ces constructions. Actuellement, la sémantique comportementale de chaque DSL est donc généralement décrite de manière ad-hoc, et il est difficile, lorsque plusieurs langages sont utilisés pour concevoir un même système, de corrélérer leurs sémantiques.

La sémantique des Langages de Programmation est par contre formalisée depuis de nombreuses années. La sémantique d'un langage définit de manière précise et non ambiguë la signification des constructions de ce langage. On dit qu'une sémantique est *formelle* lorsqu'elle est exprimée dans un formalisme mathématique. On distingue la *sémantique statique* qui est vérifiée statiquement lors de la compilation des programmes (et exprime des règles, comme le typage, qui ne peuvent pas être exprimées par la syntaxe) et la *sémantique dynamique* (ou *comportementale*) qui permet de décrire le comportement des programmes à l'exécution. Une sémantique comportementale peut être opérationnelle, dénotationnelle ou axiomatique, pour aller de la plus précise à la plus abstraite. Une *sémantique opérationnelle* donne une vision impérative en décrivant un programme par un ensemble de transitions (ou transformations) entre les états du contexte d'exécution (p. ex. de la mémoire). Une *sémantique dénotationnelle* décrit, sous forme de fonctions, l'effet d'un programme et non la manière dont celui-ci est exécuté. Une *sémantique axiomatique* propose une vision déclarative en décrivant des propriétés et l'évolution de ces propriétés lors du déroulement d'un programme. À ces approches se rajoutent les *sémantiques par traduction* qui décrivent un langage par sa transformation vers un autre langage formellement défini.

Divers moyens ont été proposés pour exprimer une ou plusieurs de ces sémantiques dans un cadre IDM : extension d'un langage de méta-modélisation (p. ex. avec Kermeta (Muller *et al.*, 2005)), règles de réécriture (approche grammairale, graphe, QVT relationnel), etc. De plus, de nombreux projets en cours reposent sur l'IDM

pour la conception de systèmes critiques : Topcased ([www.topcased.org](http://www.topcased.org)), SPACIFY ([spacify.gforge.enseeiht.fr](http://spacify.gforge.enseeiht.fr)), GeneAuto ([geneauto.gforge.enseeiht.fr](http://geneauto.gforge.enseeiht.fr)), SPICES ([www.spices-itea.org](http://www.spices-itea.org)), OpenEmbeDD ([openembedd.inria.fr](http://openembedd.inria.fr)), ModelWare ([www.modelware-ist.org](http://www.modelware-ist.org)), ModelPlex ([www.modelplex-ist.org](http://www.modelplex-ist.org)). L'expression de la sémantique est un point majeur de ces différentes initiatives avec des solutions variées. Il devient donc important de capitaliser les solutions envisagées pour construire une ingénierie de la sémantique pour l'IDM, tant au niveau de la méta-modélisation pour décrire la sémantique des langages de modélisation, qu'au niveau des modèles pour proposer des moyens généraux (voire génériques) pour exprimer la sémantique de modèles spécifiques.

SéMo'07 avait pour objectif de contribuer à la convergence des travaux de différentes communautés sur la sémantique. Le but était de partager les expériences de chacun et de faire un tour d'horizon des principales techniques pour exprimer la sémantique des modèles. L'atelier se voulait ouvert à toute proposition, même si quelques thèmes ont été proposés, dans un appel à contributions, pour amorcer le débat :

1) **sémantique des langages et extension comportementale de DSL**. Il faut pouvoir exprimer la sémantique d'un DSL, par exemple pour animer ses modèles ou réaliser des vérifications formelles (vérification de propriétés statiques ou dynamiques). Plusieurs techniques sont envisageables, comme par exemple :

- une *approche axiomatique* pour exprimer sous forme de contraintes les propriétés d'un système et les opérations qui permettent de le faire évoluer. On peut le faire au moyen d'invariants, de préconditions et de postconditions,

- une *sémantique opérationnelle* en équipant par exemple le DSL d'un langage d'action pour décrire l'effet des opérations sur un modèle,

- une *sémantique de traduction* en donnant la sémantique du DSL par traduction dans un autre formalisme bien défini, par exemple les réseaux de Petri.

Dans la mesure où différentes solutions existent pour exprimer la sémantique d'un DSL, il est important de pouvoir comparer les différentes sémantiques définies, par exemple pour en vérifier l'équivalence.

2) **prise en compte des points de variation sémantique** dans un Développement Dirigé par les Modèles (MDD). À l'instar d'UML qui ne définit pas complètement sa sémantique mais laisse des possibilités d'adaptation au travers de la définition de points de variation sémantique, il est envisageable, voire souhaitable que la sémantique d'un DSL puisse être adaptée, voire puisse évoluer au cours d'un développement pour mieux correspondre aux besoins du projet réel. Il est parfois difficile de définir la sémantique de l'ensemble d'un DSL. Il est alors nécessaire d'exprimer rigoureusement ce que signifie l'absence de sémantique : un choix parmi plusieurs possibilités, un paramètre générique qui sera instancié ultérieurement, la séparation du DSL en plusieurs préoccupations tissables dont seules certaines sont entièrement définies, etc. Dans le cas le plus défavorable, la sémantique du modèle peut se limiter à la construction de la donnée qui représente le modèle. Il peut être alors nécessaire d'introduire une relation de raffinement entre les différentes sémantiques d'un même modèle au

fur et à mesure du développement.

3) **application des techniques de langages de programmation à l'IDM**. Après le choix du type de sémantique, il s'agit d'exprimer la sémantique choisie en s'inspirant par exemple des paradigmes utilisés dans les langages existants.

4) **techniques pour la validation et la vérification de modèles (approche formelle, simulation, test)**. Ces techniques exigent une définition rigoureuse des langages de modélisation utilisés. Il est intéressant d'étudier l'impact de l'utilisation de telle ou telle approche sémantique sur l'application de ces techniques de validation.

5) **modèles de calcul (models of computation)**. Il est important de distinguer les différents points de vue abordés dans la définition d'un DSL et de sa sémantique : type de fonctionnalité (flot de contrôle ou flot de données), mode de communication, notions temporelles (temps continu ou discret, dirigé par une horloge ou par événements). Il peut être intéressant, de plus, de situer ce DSL par rapport aux modèles classiques de calcul (événements discrets, langages synchrones, machines à états finis, data flow, etc.).

6) **l'outillage**, enfin, est un aspect important. En effet, l'idée sous-jacente à l'IDM est la possibilité de définir facilement de nouveaux DSL, c'est-à-dire des langages pour couvrir des besoins spécifiques. Il faut alors pouvoir représenter les concepts de ces langages (à l'aide de méta-langages tels que MOF (OMG, 2006) ou Ecore), exprimer une syntaxe concrète, éventuellement graphique (cf. projets GMF (GMF, 2007), Topcased (Farail *et al.*, 2006), TCS (Jouault *et al.*, 2006), Syntaks (Muller *et al.*, 2006)), mais aussi développer les outils qui vont en exploiter les modèles comme par exemple des animateurs, des vérificateurs de propriétés, des générateurs de code.

## 2. Les contributions

Les quatre contributions retenues pour l'atelier couvrent bien les thèmes proposés. Nous indiquons ci-dessous les aspects couverts et les points forts de ces contributions, le contenu technique de chacune étant ensuite décrit dans une sous-section spécifique.

La première contribution (Cuccuru *et al.*, 2007) s'inscrit dans les aspects 2, 3 et 6. Les auteurs proposent d'adapter la technique de généricité des langages de programmation comme une solution à la formalisation des points de variation sémantique.

(Koudri *et al.*, 2007) aborde plutôt les aspects 1 et 3. S'appuyant sur l'infrastructure d'UML2, les auteurs montrent les limites du couple MOF+OCL pour exprimer la sémantique d'un méta-modèle et proposent d'utiliser les techniques de méta-programmation avec Kermeta pour exprimer une sémantique opérationnelle.

La contribution (Combemale *et al.*, 2007) concerne l'axe 1. Les auteurs identifient et comparent les différentes solutions permettant d'attacher une sémantique à un méta-modèle. Ils indiquent quand elles peuvent être utilisées voir combinées et relatent un exemple concret de mise en oeuvre.

Enfin, la contribution (Boniol *et al.*, 2007) porte sur le thème 5 et dans une moindre mesure sur le thème 4. Elle soulève des questions importantes sur la modélisation d'un système complexe, donc nécessairement composé de plusieurs modèles correspondant à autant de vues sur le système et sur le positionnement de la sémantique : en particulier elle pose la question de la possibilité et même de l'intérêt de définir une sémantique globale ou au contraire des sémantiques sur chaque vue du système.

### **2.1. Méta-modèles et Points de Variation Sémantique**

Arnaud Cuccuru *et al.* (Cuccuru *et al.*, 2007) s'intéressent aux points de variation sémantique (PVS) dans les méta-modèles. Ces PVS constituent une information essentielle dans le domaine de l'ingénierie des langages mais sont aussi présents dans l'IDM où ils ne sont pas clairement identifiés et sont décrits le plus souvent informellement dans la documentation associée aux méta-modèles. La spécification d'UML 2 (OMG, 2007a; OMG, 2007b) est un bon exemple de cet état de fait. Le MOF ne propose aucun mécanisme pour identifier ni traiter les PVS.

Les auteurs proposent de s'inspirer du mécanisme de genericité présent dans les langages de programmation pour paramétrer un méta-modèle, aussi bien au niveau des paquetages que des méta-classes. La notation proposée s'appuie sur les templates d'UML 2 et est illustrée sur les machines à états d'UML 2 en s'intéressant à deux points de variation sémantique : la sélection des événements et la sélection des transitions. Chacun de ces deux PVS est décrit par une méta-classe, déclarée comme paramètre de genericité du paquetage des machines à état et de la méta-classe *State* concernée par ces PVS. Deux réalisations concrètes de ces PVS sont proposées correspondant à des politiques respectivement aléatoire et stochastique.

### **2.2. Une sémantique opérationnelle pour une meilleure méta-modélisation**

L'objet de (Koudri *et al.*, 2007) est la définition formelle, complète et consistante des langages de modélisation par une spécification enrichie du méta-modèle. Ce type de définition est nécessaire d'une part, pour permettre l'interopérabilité entre les différents outils (vérificateur de modèles, générateur de code, simulateur...) et pas seulement entre les éditeurs comme c'est le cas pour la définition en MOF, et d'autre part pour faciliter l'utilisation de l'IDM pour les systèmes critiques. Pour cela, les auteurs s'appuient sur l'utilisation d'OCL pour exprimer des contraintes statiques plus précises sur les diagrammes MOF mais aussi sur la possibilité d'associer des opérations et leur dynamique aux concepts du méta-modèle en utilisant un langage d'action comme Kermeta. Les auteurs s'appuient sur des extraits du méta-modèle d'UML2, en particulier les concepts *Element* et *RelationShip*, pour illustrer les limitations d'une représentation en Ecore puis Ecore+OCL, et pour introduire différentes approches pour construire une description plus satisfaisante en s'appuyant sur Kermeta. Il s'agit principalement de remplacer un attribut avec une multiplicité quelconque par des opérations « *getter / setter* » dont la définition en Kermeta est plus précise qu'un simple

attribut équipé de contraintes OCL. Le point essentiel est qu'un *Element* peut contenir d'autres éléments mais ne peut pas se contenir lui-même, y compris transitivement.

### 2.3. Expérimentation pour la définition d'une sémantique dans l'IDM

Benoît Combemale et al (Combemale *et al.*, 2007) visent à définir une sémantique comportementale pour un langage dédié (DSL), soit pour animer ou simuler des modèles écrits dans ce langage, soit pour vérifier des propriétés sur ces modèles.

Dans un objectif d'animation (visualisation du comportement) ou de simulation de modèles, les auteurs penchent pour l'utilisation d'une sémantique opérationnelle, ce qui permet de rester dans un seul cadre technologique. Dans un objectif de vérification, ils proposent par contre d'utiliser une approche par traduction vers un langage cible disposant d'outils de vérification éprouvés (p. ex. un model-checker) pour bénéficier de ces outils. Il est alors nécessaire de garantir la validité de la traduction, par des preuves par bisimulation pour la traduction d'une sémantique opérationnelle ou par des vérifications de consistance pour la traduction d'une sémantique axiomatique.

Les auteurs appuient leur propos sur une expérimentation menée sur SimplePDL, un langage de description de procédé. Dans cette expérimentation, ils ont défini une sémantique opérationnelle du langage (en Kermeta et en ATL), formalisé une transformation des modèles vers les réseaux de Petri et des propriétés vers du LTL, puis vérifié ces propriétés en utilisant le model-checker de Tina (Berthomieu et al, 2004).

### 2.4. Points de vue et sémantiques ad-hoc

Dans le cadre de la conception des systèmes critiques de plus en plus complexes et désormais à logiciels prépondérants, l'article de Boniol et al. (Boniol *et al.*, 2007) prône l'utilisation de la notion de point de vue (fonctionnel, tolérance aux fautes, évaluation des performances, robustesse...) comme un moyen de casser la complexité intrinsèque à la validation et vérification d'un système en découpant cette tâche (de validation et vérification) selon un ensemble de préoccupations transverses. L'utilisation d'une telle approche pose toutefois différentes questions dont celle de la cohérence entre les différents modèles des différents points de vue.

Cet article ouvre une réflexion sur ce problème et propose une approche consistant à voir les points de vue comme des interprétations sémantiques différentes d'un même modèle global et complet du système étudié. Cette approche est d'autant plus naturelle que l'IDM offre un découpage entre la syntaxe (le modèle structurel) et la sémantique (l'interprétation du modèle structurel). L'idée est donc d'offrir : 1) des moyens d'expression d'un modèle structurel global avec une sémantique minimale et 2) des transformations de modèles, chaque transformation encodant en quelque sorte la sémantique d'un point de vue donné, et permettant la génération de squelette vers des langages dédiés aux points de vue.

### 3. Discussions

À la suite de chaque présentation, les questions posées ont donné lieu à des discussions dont les principaux thèmes sont repris ci-dessous.

**Rôle des méta-modèles.** Suite à la première présentation qui propose d'utiliser les techniques de la programmation dans le domaine de la modélisation, en l'occurrence méta-programmation et généricité, le rôle des méta-modèles a été soulevé. Il ressort de la discussion que le méta-modèle n'est pas toujours unique ; qu'il n'est pas forcément le plus approprié pour définir et pour implémenter la sémantique (p. ex. les réseaux de Petri où l'utilisation de matrices est plus efficace pour l'implémentation que la structure de graphe à base de places, transitions et arcs, plus adaptée à la définition) et que son intérêt premier n'est pas l'exécution des modèles, mais la définition de la syntaxe abstraite permettant de manipuler ces modèles. Il existe donc plusieurs points de vue dans la définition d'un langage de modélisation qui pourraient demander l'utilisation de différents méta-modèles coordonnés. Cette remarque conduit également parfois à la définition de méta-modèles spécifiques à certaines syntaxes concrètes.

**Sémantique dans la modélisation.** La prise en compte de la sémantique des langages de modélisation a été évoquée dans le cadre de l'activité de (méta-)modélisation ainsi que la possibilité de définir plusieurs sémantiques pour une même syntaxe abstraite. La discussion a également permis d'identifier les points ouverts suivants : un DSL a-t-il besoin d'une autre description (sémantique) que la simple définition de modèles pour faciliter la compréhension de ces derniers ? Est-ce alors une absence de sémantique ou une sémantique faible ? Un DSL doit-il offrir une certaine souplesse dans la sémantique des constructions exploitées dans un modèle (p. ex. des PVS) ? La sémantique d'un DSL doit-elle évoluer au fur et à mesure du cycle de vie d'un modèle ?

**Système global vs. Points de vue.** Enfin, la dernière présentation a mis en évidence la difficulté de capturer la totalité d'un système complexe dans un unique modèle global. Au delà de la notion de point de vue proposée, la discussion a porté sur la difficulté à définir la sémantique au niveau du modèle global et/ou des points de vue. Se pose alors le problème de la possibilité de définir une sémantique pour un modèle global (quelle granularité ?) et la gestion de la cohérence entre les sémantiques des différents points de vue. Quel équilibre trouver dans la définition de la sémantique du modèle global et pour les différents points de vue ? Cette problématique, largement traitée dans le cadre de la représentation des données, est plus complexe pour le comportement et s'apparente à la séparation et au tissage des préoccupations.

### 4. Synthèse et Perspectives

L'atelier SéMo a regroupé 49 personnes. Les débats relatés ci-dessus ont été animés avec une participation active du public. Ils ont soulevé au moins autant de questions qu'ils ont proposé de solutions. Parmi les questions ouvertes, nous noterons : le rôle du méta-modèle dans la définition de la sémantique, l'existence de variations de la sémantique selon le cycle de vie correspondant à différentes interprétations



d'un même langage, et les préoccupations multiples dans la définition d'un système. D'autres aspects n'ont pas, ou peu, été abordés dans l'atelier mais restent des sujets d'étude essentiels, tels que l'exploitation des approches sémantiques pour la validation et la vérification des modèles, et l'efficacité des contrôles sémantiques dans les outils de manipulation des modèles (éditeurs, animateurs, vérificateurs...).

Le succès de cette première édition de l'atelier SéMo et les nombreuses questions ouvertes nous encouragent à poursuivre l'effort de créer et faire vivre une communauté sur la thématique de la sémantique dans les DSL, associée à l'action IDM ([www.actionidm.org](http://www.actionidm.org)). Un premier pas dans cette voie est la mise en place du site web <http://semo.enseeiht.fr/> contenant déjà les actes et présentations de l'atelier 2007. Il sera enrichi régulièrement de contributions de ce thème de recherche.

## 5. Bibliographie

- Boniol F., Dhaussy P., Pagetti C., « Points de vue et sémantiques ad hoc », in INRIA (ed.), *3<sup>e</sup> journées sur l'Ingénierie Dirigée par les Modèles*, France, p. 235-241, March, 2007.
- Combemale B., Crégut X., Garoche P.-L., Thirioux X., « Expérimentation pour la définition d'une sémantique dans l'IDM », in INRIA (ed.), *3<sup>e</sup> journées sur l'Ingénierie Dirigée par les Modèles*, France, p. 229-234, March, 2007.
- Cuccuru A., Mraidha C., Terrier F., Gérard S., « Métamodèles et Points de Variation Sémantique », in INRIA (ed.), *3<sup>e</sup> journées sur l'Ingénierie Dirigée par les Modèles*, Toulouse, France, p. 217-222, March, 2007.
- Farail P., Gauffillet P., Canals A., Camus C. L., Sciamma D., Michel P., Crégut X., Pantel M., « The TOPCASED project : a Toolkit in OPen source for Critical Aeronautic SystEms Design », *Embedded Real Time Software (ERTS'06)*, France, January, 2006.
- GMF, *Graphical Modeling Framework Tutorial*. April, 2007, available on Eclipse wiki : [http://wiki.eclipse.org/index.php/GMF\\\_Tutorial](http://wiki.eclipse.org/index.php/GMF\_Tutorial).
- Jouault F., Bézivin J., Kurtev I., « TCS : a DSL for the Specification of Textual Concrete Syntaxes in Model Engineering », *5th int. conference on Generative Programming and Component Engineering (GPCE'06)*, Oregon, October, 2006.
- Koudri A., Champeau J., Aulagnier D., « Une sémantique opérationnelle pour une meilleure méta-modélisation », in INRIA (ed.), *3<sup>e</sup> journées sur l'Ingénierie Dirigée par les Modèles*, France, p. 223-228, March, 2007.
- Muller P.-A., Fleurey F., Fondement F., Hassenforder M., Schneckeburger R., Gérard S., Jézéquel J.-M., « Model-Driven Analysis and synthesis of Concrete Syntax », in O. Nierstrasz, J. Whittle, D. Harel, G. Reggio (eds), *9th int. Conference on Model Driven Engineering Languages and Systems (MoDELS'06)*, vol. 4199 of LNCS, Italy, October, 2006.
- Muller P.-A., Fleurey F., Jézéquel J.-M., « Weaving Executability into Object-Oriented Meta-Languages », in S. K. L. Briand (ed.), *MoDELS'05*, LNCS, Jamaica, October, 2005.
- OMG, *Meta Object Facility Core Specification, v2.0*. January, 2006.
- OMG, *Unified Modeling Language (UML) 2.1.1 Infrastructure*. February, 2007a.
- OMG, *Unified Modeling Language (UML) 2.1.1 Superstructure*. February, 2007b.