
Des situations de modélisation pour évaluer les outils de modélisation

Antoine Beugnard* — **Fabien Dagnat*** — **Sylvain Guérin**** — **Christophe Guychard****

* *Télécom Bretagne, IRISA*

Technopole Brest-Iroise, CS 83818, 29283 Brest cedex 3, France

prenom.nom@telecom-bretagne.eu

** *Openflexo*

Technopole Brest-Iroise, 135 rue Claude Chappe, 29280 Plouzané, France

prenom.nom@openflexo.org

RÉSUMÉ. Nous proposons d'identifier des situations de modélisation en mettant en évidence des actions élémentaires sur les artefacts de modélisation que sont les modèles et les méta-modèles. Nous pensons que l'identification de ces situations permettra de mieux comprendre la modélisation et par conséquent les besoins des outils de modélisation. Nous présentons Openflexo, un outil de modélisation libre, et esquissons une comparaison de ses capacités avec divers type d'outils de modélisation qui vont du simple outil dessin à l'atelier de méta-modélisation.

ABSTRACT. We propose to identify modeling situations highlighting elementary actions on modeling artifacts such as models and meta-models. We believe that identification of these situations will help better understand the modeling and therefore modeling tools requirements. We present Openflexo a free modeling tool, and sketch a comparison of its capabilities with various types of modeling tools ranging from simple drawing tool to meta-modeling editors.

MOTS-CLÉS : Modélisation

KEYWORDS: Modeling

1. Introduction

La modélisation est à la base de la plupart des stratégies de résolution de problèmes, en particulier de la démarche scientifique et de l'ingénierie (Morris, 1967 ; Lachapelle et Cunningham, 2007). On modélise en s'appuyant sur des écrits¹ sous formes de phrases ou de dessins. Ces phrases et dessins, artefacts issus de la modélisation, doivent respecter des règles de construction plus ou moins explicites ou formalisées.

Dans « On the art of modeling » (Morris, 1967), W. Morris propose de passer d'un processus de modélisation intuitif à une approche explicite. Il illustre l'article par un problème de planification de transport. Au-delà des étapes de réflexion, il fait apparaître deux étapes qu'on retrouve dans tout processus de modélisation :

- 1) *Consider a specific (...) instance of the problem* ; identifier des exemples.
- 2) *Establish some symbols* ; déterminer leur généralisation ou abstraction, et en définir des représentations, à travers des variables mathématiques par exemple.

Il note que la production de ces artefacts (les exemples et les symboles) suit un processus d'élaboration par enrichissement :

« The process of model development may be usefully viewed as a process of *enrichment* or *elaboration*. One begins with very simple models, quite distinct from reality, and attempts to move in evolutionary fashion toward more elaborate models which more nearly reflect the complexity of the actual management situation. »

Enfin, il met également en évidence le besoin de liens (implicites ou explicites) entre ces artefacts :

« *Analogy* or *association* with previously well developed logical structures plays an important role in the determination of the starting point of this process of elaboration or enrichment. »

Dans la suite de cet article, nous appelons *modèle* une représentation explicite d'un système ou d'un problème et *méta-modèle*, l'ensemble des règles explicites - quel soit leur mode d'explicitation - qu'un modèle doit respecter. Nous verrons que les liens entre modèle et méta-modèle peuvent être connus *a priori*, mais sont parfois élaborés *a posteriori*, comme le résultat d'un choix de modélisation.

La formalisation des outils de modélisation s'est concrétisée, pour les phrases, par la théorie des langages et des grammaires, pour les dessins, par les approches dites dirigées par les modèles. Dans les deux cas, les modèles produits (phrases ou dessins) sont *conformes* aux règles exprimées respectivement dans la grammaire ou le

1. Même si les premiers philosophes discutaient sur l'Agora sans laisser de traces écrites et que nos ancêtres ont dû résoudre bien des problèmes avant l'invention de l'écriture. . .

méta-modèle. Cette relation de conformité est l'une des relations possibles entre modèles et méta-modèles. Nous considérerons donc deux niveaux de modélisation : le modèle (exemple, instance, concrétisation) et le méta-modèle (généralisation, abstraction). Dans le sens où nous l'utilisons, le méta-modèle (Kleppe, 2007) est l'expression des concepts (lexique, ou vocabulaire) utilisables pour élaborer un modèle et de toutes les règles et contraintes (grammaire) d'assemblage de ces concepts. Notre approche peut être considérée comme « catégorique » au sens mathématique du terme. Nous nous intéressons aux relations entre modèles et méta-modèles sans prendre en considération leur structure ou leur contenu.

Nous noterons que dans la plupart des outils et démarches de modélisation existants, la relation de conformité est considérée comme stricte i.e. l'élément modélisé conserve exactement et uniquement les propriétés définies dans son méta-modèle, mais nous ne retenons pas cette hypothèse dans notre démarche. En effet, elle se révèle parfois être une contrainte prescriptive qui appauvrit l'expressivité des modèles. Enfin, on rencontre aussi souvent la relation d'instantiation (*instanceof*) qui relie un modèle créé depuis un méta-modèle à ce méta-modèle. Dans le cadre de cet article et suivant la vision catégorique déjà exposée, nous ne nous intéressons pas à l'intérieur des artefacts de modélisation et donc ne considérons pas cette relation d'instantiation.

Nous pensons que l'identification des usages de ces deux niveaux de modélisation peut servir de base à la définition d'exigences pour les ateliers de modélisation. Nous présentons donc dans la partie 2 des situations rencontrées lors de travaux de modélisation en les illustrant avec des exemples concrets. Nous abordons rapidement la composition de ces situations (3). Puis nous esquissons une comparaison (5) de différents outils comme des applications de dessin, des ateliers de modélisation ou de méta-modélisation avec l'atelier Openflexo (4) qui met en œuvre l'approche de fédération de modèles. Nous comparons notre approche avec d'autres travaux dans la partie 6 avant de conclure.

2. Situations de modélisation

Pour décrire les situations de modélisation, nous avons deux types d'artefacts à considérer : les modèles et les méta-modèles. Les situations varient selon l'ordre dans lequel ces artefacts apparaissent ou sont reliés dans la démarche. Nous simplifions la description en ne considérant qu'un seul acteur dans chaque situation. Une analyse plus fine de ces situations pourraient être intéressante en prenant en compte différents acteurs et donc différentes intentions dans le processus. Les figures 1 et 2 présentent les 11 situations considérées. Chacune des situations est décrite en détail et illustrée dans une des sous sections suivantes par un exemple concret issu de la géométrie.

- 1) Un méta-modèle existe, on cherche produire un modèle [*concrétisation*].
- 2) Un modèle existe, le travail consiste à trouver un méta-modèle [*abstraction*].
- 3) Un modèle existe, il faut trouver plusieurs méta-modèles [*multi-abstraction*].
- 4) Des modèles existent, il faut élaborer un méta-modèle [*co-abstraction*].

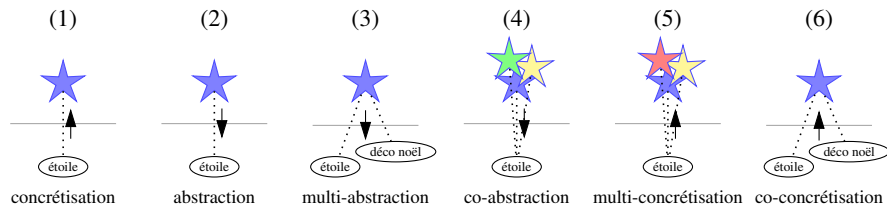


Figure 1. Des situations de modélisation (haut : niveau instance, et bas : niveau méta)

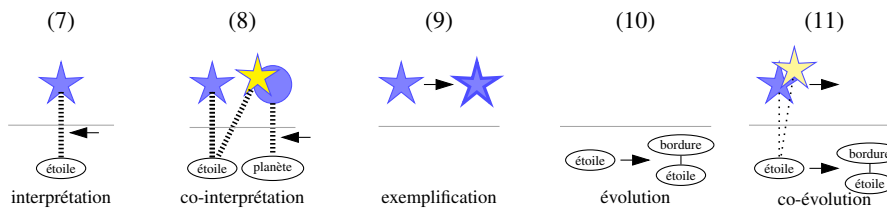


Figure 2. Des situations de modélisation (haut : niveau instance, et bas : niveau méta)

5) Un méta-modèle existe, le travail consiste à construire plusieurs modèles [*multi-concrétisation*].

6) Des méta-modèles existent, le travail consiste à construire un modèle [*co-concrétisation*].

7) Un modèle et un méta-modèle existent, il faut les relier [*interprétation*].

8) Des modèles existent, des méta-modèles existent, le travail consiste à les relier [*co-interprétation*].

9) Un modèle existe, le travail consiste à construire un autre modèle (sans aucun méta-modèle) [*exemplification/extension*].

10) Un méta-modèle existe, le travail consiste à construire un autre méta-modèle (sans aucun modèle) [*évolution/extension*].

11) Un méta-modèle existe avec plusieurs de ses modèles conformes, le travail consiste à faire évoluer le méta-modèle (cas précédent) en adaptant (ou non) ses modèles [*co-évolution*].

Les cas (9) et (10) sont probablement équivalents dans le cadre d'une interprétation multi-niveaux de la modélisation. En effet, à un niveau donné d'abstraction, l'absence de référence à un autre niveau, plus concret ou plus abstrait, rend le travail équivalent. Nous les différencions tout de même car la plupart des outils proposent des moyens de manipulations de ces deux niveaux très différents, liés à leur mode de représentation.

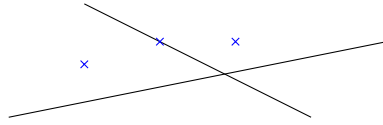


Figure 3. *Des éléments géométriques (points et droites)*

2.1. Concrétisation

La concrétisation est la forme d'utilisation la plus classique d'un outil de modélisation. Les concepteurs de l'outil ont préparé des éditeurs adaptés à un ensemble de concepts (méta-modèle). Les utilisateurs élaborent des modèles en respectant les règles prévues. Cette approche est efficace lorsque le modèle que l'on souhaite construire est adapté au méta-modèle utilisé, i.e. les concepts nécessaires sont présents. Dans le cas où les concepts ne s'alignent pas exactement, les utilisateurs sont parfois amenés à tordre les interprétations ou imaginer des usages des concepts non prévus.

Exemple 1 *La figure 3 illustre un modèle d'éléments géométriques, instances des concepts point et droite.*

2.2. Abstraction

Ce cas, très classique, également dans une démarche de modélisation consiste à partir d'un exemple à identifier les concepts et les règles pour construire un méta-modèle auquel le modèle initial est conforme. Le résultat est exploité par des développeurs d'outil pour construire des éditeurs de modèles.

Exemple 2 *À partir de la figure 3, on peut proposer un méta-modèle géométrie. Pour illustrer et faciliter la compréhension, nous faisons apparaître les concepts (éléments de modèles) de point et de droite. Le méta-modèle pourrait définir une droite comme un ensemble de points, que tous les points situés entre 2 points quelconques d'une droite appartiennent à cette droite. On notera que « situé entre » n'est pas défini, et que l'espace considéré est implicitement un plan.*

2.3. Multi-abstraction

Ce cas, moins classique, généralise la situation d'abstraction, où le but est, à la fois d'abstraire, mais également d'organiser l'abstraction en faisant apparaître des points de vue complémentaires sur le modèle (2 dans la situation 3 de la figure 1). Nous ne différencions pas les méta-modèles, l'un peut pré-exister ou ils peuvent être conçus tous en même temps. L'important est qu'il existe plusieurs méta-modèles.

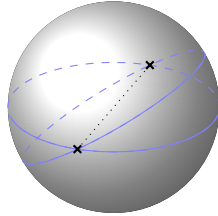


Figure 4. *D'autres points et droites*

Exemple 3 *À partir de la figure 3, il est possible de construire plusieurs méta-modèles. En complément de celui imaginé dans l'exemple précédent, nous pourrions utiliser les concepts de bâtons (les traits) et cailloux (les « points ») dans le but, par exemple, de modéliser un jeu.*

2.4. Co-abstraction

Cette pratique est une généralisation de la deuxième. Plusieurs modèles servent d'exemples pour construire un méta-modèle auxquels tous les exemples seront conformes. Les démarches taxinomiques ou de classification sont des situations de co-abstraction. L'intérêt d'utiliser plusieurs modèles exemples permet de confronter les concepts et de faire apparaître des consensus ou, au contraire, des différences d'interprétation et des conflits conceptuels.

Les situations de co-abstraction et de multi-abstraction sont combinables en une multi-co-abstraction non représentée. Nous discuterons de « composition » des situations à la fin de cette partie (3).

Exemple 4 *À partir des figures 3 et 4, il est possible de construire un méta-modèle plus général de la notion de point et de droite. En effet, sur une sphère, on peut interpréter un grand cercle par le concept de droite et l'intersection de deux grands cercles par le concept de point (qui en serait deux dans l'interprétation classique).*

2.5. Multi-concrétisation

Ce cas est une généralisation du premier. Le même méta-modèle est utilisé à plusieurs reprises pour produire différents modèles. Le risque est que lorsque les interprétations ne sont pas les mêmes pour produire les modèles, des incohérences peuvent apparaître sans être détectables. Le risque de cette approche est d'autant plus important que les interprétations du méta-modèle sont nombreuses comme c'est le cas avec UML et ses multiples points de variation sémantique (OMG, 2007). Par exemple, la sémantique des diagrammes d'états n'est pas définie précisément en UML et peut donner lieu à diverses interprétations (Chauvel et Jézéquel, 2005).

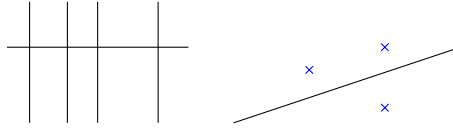


Figure 5. Des modèles produits à partir des concepts point et droite

Exemple 5 À partir des concepts *point* et *droite*, on peut produire les modèles de la figure 5.

2.6. Co-concrétisation

Cette situation se rencontre lorsque plusieurs experts, chacun disposant de son propre méta-modèle, se réunissent pour décrire un système dans lequel leurs points de vue doivent être représentés. Dans cette situation on *souhaite* laisser les abstractions séparées et ne pas construire un méta-modèle commun.

Exemple 6 On peut imaginer, pour représenter un bovin, opter soit pour le point de vue d'un boucher, soit pour le point de vue d'un vétérinaire. Bien que pouvant paraître proche (partage de certains découpages anatomiques), les deux spécialités sont cependant suffisamment éloignées pour que la description du même animal ne puissent pas être partagée (fonctions et propriétés des organes différentes).

2.7. Interprétation

Ce cas, s'il est moins habituel, participe sans aucun doute à la démarche de modélisation, au moins dans la tête des « modélisateurs ». Un modèle exemple existe ainsi qu'un méta-modèle. Cette situation est courante et s'appuie sur la connaissance des concepts et de leur organisation par un expert. Le travail consiste à relier les éléments du modèle exemple à un concept présent dans le méta-modèle dans le but :

- soit de s'assurer que l'exemple respecte les règles du méta-modèle ;
- soit de trouver un contre-exemple (le modèle exemple) au méta-modèle afin de le faire évoluer.

Exemple 7 La figure 6 montre une interprétation d'un dessin existant, pour un méta-modèle (représenté ici dans une version simplifiée par deux boîtes²).

2. On notera qu'un méta-modèle, peut être un objet d'intérêt et devenir lui-même un modèle s'appuyant sur une ou des représentations. La circularité de ces concepts ne simplifie ni leur description, ni leur usage.

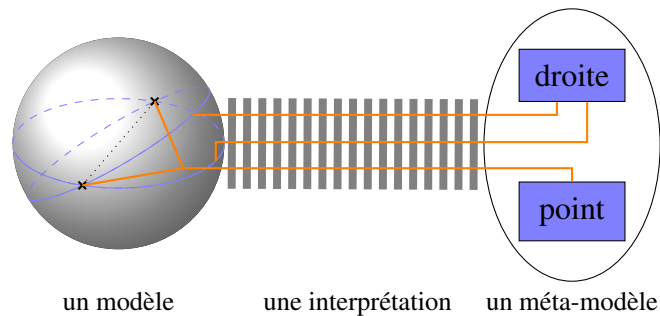


Figure 6. Une interprétation (l'ensemble des traits pointillés) : deux droites et leur point (unique) d'intersection

L'exemple 3 (la multi-abstraction) montre l'importance de l'interprétation. Un grand cercle peut être interprété comme une droite. Mais alors, une droite « classique » n'a plus de sens. Un éditeur de dessin pourra dessiner l'une ou l'autre des interprétations, mais si les deux sont mélangées, il y a des risques d'erreur d'interprétation.

Il faut noter que la relation d'interprétation n'est pas la relation d'instantiation. L'instantiation repose sur un mécanisme de construction d'une représentation particulière. Un même concept peut être instancié de multiples manières. L'interprétation décrit une intention, celle de mettre en relation une représentation et son sens, défini et décrit dans un méta-modèle.

2.8. Co-interprétation

Ce cas est une généralisation du cas précédent. Plusieurs modèles exemples existent ainsi que plusieurs méta-modèles. Cette situation est encore plus courante que la précédente et s'appuie sur la connaissance des concepts et de leurs organisations par un expert. Le travail consiste à relier les éléments des modèles à un (ou plusieurs) concept(s) présent(s) dans les méta-modèles dans le but :

- soit de s'assurer que les exemples respectent les règles des méta-modèles ;
- soit de trouver un contre-exemple aux méta-modèles afin de les faire évoluer.

Exemple 8 Les interprétations de droite dans les géométries proposées en exemple ne sont pas compatibles, alors que celles de droite et baton le sont probablement.

2.9. Exemplification/Extension

Cette situation est souvent la première rencontrée. Il s'agit de produire des exemples de représentation qui servent de base à la réflexion. Pour des dessins ou des dia-

grammes, on choisira des couleurs, des formes, des positions relatives des formes pour représenter le problème. L'excellent article de D. Moody (Moody, 2009) passe en revue de nombreux exemples de représentations graphiques. Une histoire des représentations est présentée par M. Friendly dans (Friendly, 2005 ; Friendly et Denis, 2001). Pour les langages, une variabilité extraordinaire des langues et grammaires est observable dans le dictionnaire des langues (Peyraube *et al.*, 2010).

Exemple 9 *Sur la partie gauche de la figure 5, on pourrait introduire une représentation d'un angle droit pour signifier que la droite coupe perpendiculairement une autre droite. La signification « perpendiculaire » est une intention, non encore identifiée dans un méta-modèle, mais qui vient enrichir la notation.*

2.10. Évolution/Extension

Cette situation, parallèle à la précédente, ne peut être mise en place qu'une fois la conceptualisation réalisée, c'est-à-dire lorsqu'un méta-modèle est disponible. Il s'agit de faire évoluer les concepts ou les règles les gouvernant. Ce travail se réalise souvent - explicitement ou non - en référence à des évolutions des modèles.

Exemple 10 *Parallèlement au cas précédent, on pourrait enrichir le méta-modèle point/droite avec un concept d'angle (ou d'angle droit).*

2.11. Co-évolution

C'est une situation liée à la précédente, où le méta-modèle évolue et était relié à des modèles existants ; qu'advient-il des modèles existants ? L'article (Sprinkle et Karsai, 2004) est l'un des premiers à identifier le problème avec le vocabulaire de l'ingénierie dirigée par les modèles.

Ce cas est très complexe car il peut prendre de nombreuses formes. Par exemple, si un nouveau concept est introduit dans le méta-modèle sans qu'aucune instance de ce concept n'ait été précédemment créée, la solution est triviale. Par contre, si la structure d'un concept est remise en cause et que des instances avaient été créées, la solution pourrait ne pas avoir de solution automatisable et demander l'intervention d'un humain pour guider l'évolution.

Nous identifions cette situation, mais ne l'illustrons pas et la laissons à de futures investigations.

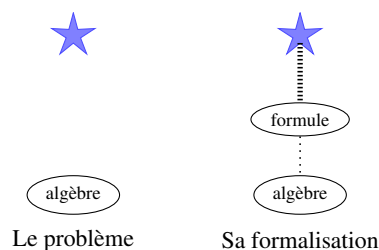


Figure 7. Une composition pour représenter une approche algébrique.

3. Composition

Une fois des situations de modélisation identifiées, la question de leur composition se pose naturellement. Nous ne ferons qu’illustrer quelques exemples relevant de ce large problème :

– *Composition comme séquence de situations* : la situation de multi-concrétisation (respectivement multi-abstraction) peut être vue comme une composition de plusieurs concrétisation (resp. abstraction). Ce n’est pas nécessairement le cas, car le fait de regrouper plusieurs actions dans la même situation ne représente pas exactement la même situation que de reproduire *indépendamment* plusieurs fois l’action.

– *Composition comme empilement* : un modèle peut parfois être interprété comme un méta-modèle et conduire à la création de modèles de « niveaux » différents. Inversement, un méta-modèle peut aussi être considéré comme la concrétisation d’un méta-méta-modèle. Il est donc possible d’empiler les situations de concrétisation, abstraction ou interprétation. La figure 7 montre une formulation de la démarche d’algébrisation en 2 étapes. La première (à gauche) consiste à identifier un problème (le modèle) et une théorie (le méta-méta-modèle). La seconde étape (partie droite de la figure 7) consiste à produire une formule qui représente le problème qui est une concrétisation de la théorie et une interprétation du modèle pour le problème considéré. On voit ici qu’on combine (et empile) deux situations élémentaires : concrétisation et interprétation.

– *Composition de modèles* : l’approche que nous avons choisie ne prend pas en compte la structure des modèles et des méta-modèles. Pourtant pour illustrer l’interprétation (figure 6), nous avons fait apparaître le contenu d’un modèle de géométrie avec point et droite. Ces *éléments de modèle* peuvent, bien entendu, eux-mêmes être considérés comme des modèles. Ceci peut également être appliqué aux méta-modèles ; des éléments de méta-modèle sont des méta-modèles. Cette relation de composition peut amener à de nouvelles situations que nous ne considérons pas ici.

Notons que ces trois formes de composition se composent. Cet article ne présente qu’une ébauche de l’étude des situations de modélisation et de leur composition qu’il conviendra d’approfondir.

4. Openflexo – L'éditeur « Free Modeling »

4.1. L'infrastructure de modélisation Openflexo

Le projet Openflexo vise à construire une infrastructure logicielle dédiée à la construction d'ateliers adaptables. Son architecture modulaire est complétée par une méthode d'assemblage dirigée par les modèles, qui offre une grande souplesse pour la construction d'environnements de modélisation sur mesure. Cette approche, nommée *Diatomée*, s'appuie sur les capacités des composants de l'infrastructure :

- un langage pour la connection dynamique de graphes d'objets (*Connie*) ;
- un *framework* de modélisation supportant l'héritage multiple en Java (*Pamela*) ;
- une API pour la construction d'outils de représentation graphique (*Diana*) ;
- un *framework* de construction et l'interprétation de modèles d'IHM (*Gina*).

Tous ces éléments sont fortement configurables, ce qui permet de construire simplement des ateliers souples et dynamiques. Leur développement est dirigé par la volonté de fournir aux utilisateurs des outils offrant plus de liberté dans la modélisation de systèmes complexes hétérogènes. Tous les composants sont livrés sous license *Opensource* et sont écrits en Java.

4.2. La fédération de modèles

Le cœur de l'infrastructure (*openflexo core*) contient des composants dédiés à la mise en oeuvre de la *fédération de modèles*. Cette technique autorise la construction de nouveaux modèles (représentations graphiques, documents, ...) à partir d'un ensemble hétérogène de sources de données³ (vues comme des modèles). Elle peut être mise en oeuvre pour outiller aussi bien les approches de type *Architecture Framework* (Schekkerman, 2004), que la modélisation multi-paradigmes (Amaral *et al.*, 2010).

Dans sa version actuelle (*Semantics+ 1.6.1*), l'atelier supporte la création conjointe de modèles conceptuels fédérés (syntaxe abstraite) et des représentations graphiques associées (syntaxe concrète). Les diagrammes construits par l'utilisateur peuvent mélanger dessins libres et formes pré-définies. Tous les éléments du niveau méta (modèle conceptuel et formes graphiques prescrites) peuvent être modifiés en cours d'exécution. Cela permet de faire émerger de nouveaux concepts à partir des formes libres (non-associées à un concept) présentes sur les vues. Le détail de l'approche est décrit dans (Guychard *et al.*, 2013).

3. Actuellement possible avec des ontologies (OWL), des modèles EMOF/EMF, des documents XML et des tables Excel.

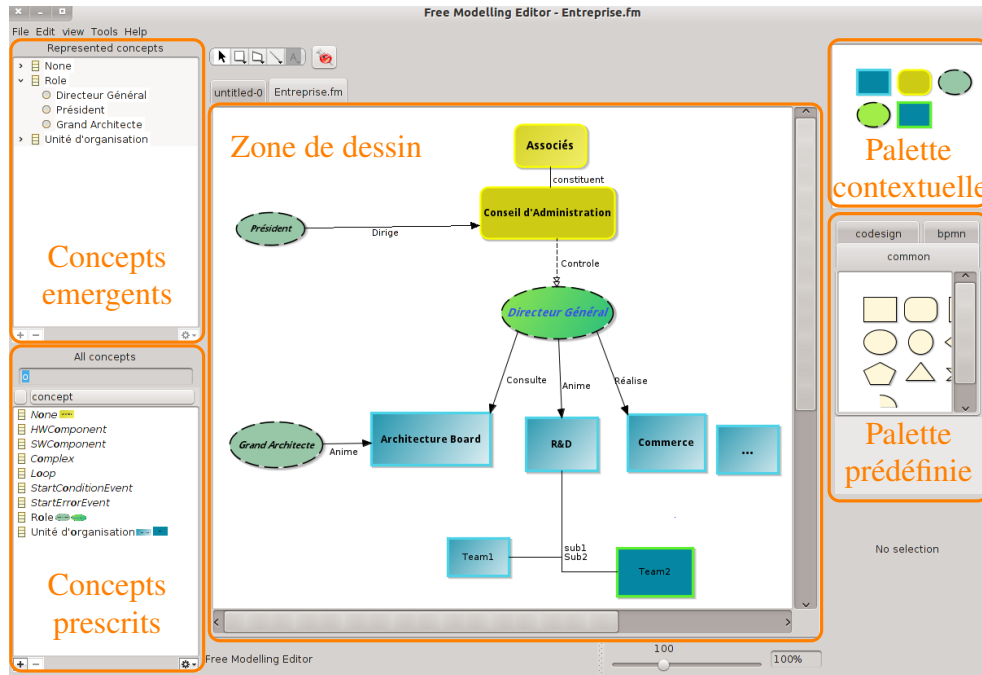


Figure 8. L'organisation du «Free Modeling Editor»

4.3. L'outil de « modélisation libre »

Assemblé selon les principes de *Diatomee* décrits ci-dessus, le « *Free Modeling Editor* » (FME) est un prototype destiné à l'expérimentation de nouveaux modes d'interaction avec les modèles. Il est utilisé pour faciliter l'émergence d'une syntaxe graphique simultanément au modèle conceptuel associé. Son interface, illustrée par la figure 8, est découpée en une partie *outils de dessin* sur la droite et une partie *modèle conceptuel* sur la gauche. L'utilisateur peut librement dessiner (situation d'*exemplification*) dans l'espace au centre en plaçant sur la page de travail des formes graphiques libres sélectionnées depuis une des palettes « métier » ou en ré-utilisant des formes définies dans ce contexte (palette « contextuelle »). Lorsqu'une forme est sélectionnée, il est possible (via un menu contextuel) de l'associer à un élément du modèle conceptuel présenté sur la gauche (*interprétation*). Si le concept correspondant n'existe pas encore, l'utilisateur peut en définir un nouveau (*abstraction*).

Il est facile d'expérimenter les différentes situations de modélisation décrites ci-avant avec le *FME*. La séparation des espaces de travail et des responsabilités conduit à un fonctionnement de l'atelier qui permet de pratiquer les 10 premières situations présentées. Toutefois, le modèle conceptuel utilisé est simpliste et on ne peut pas établir de relations structurelles ou sémantiques entre concepts.

L'outil est actuellement autonome et indépendant de l'infrastructure de fédération de modèles. Un travail est en cours pour les faire coopérer. Cela facilitera la création « à la volée » de vues définies par l'utilisateur au dessus de modèles multi-niveaux, complexes et distribués. Nous pensons que ce fonctionnement conjoint nous permettra d'outiller l'ensemble des 11 situations présentées dans cet article.

5. Évaluation d'outils

Nous considérons comme outil de modélisation des outils de dessin ou de présentation comme dans *Libreoffice*, des outils de modélisation avec méta-modèle comme des éditeurs UML, des ateliers de méta-modélisation comme *MetaEdit+*, et notre outil de fédération de modèle *Openflexo*.

Les situations présentées précédemment servent de critère de comparaison pour chacun des types d'outils. Il s'agit d'une esquisse de comparaison ; des ateliers ou des outils pourraient offrir des capacités spécifiques non étudiées dans cet article.

Outils	1	2	3	4	5	6	7	8	9	10	11	Commentaires
Dessin (Libreoffice, etc)	-	-	-	-	-	-	-	-	+	+	-	sans méta-modèle
Dédié (UML, BPMN, etc)	+	(1)	(1)	(1)	+	-	-	-	+	(1)	-	avec méta-modèle
Meta-Editeur (MetaEdit+, etc)	+	+	(2)	+	+	(2)	(2)	(2)	+	+	+/-	interprétation stricte
Openflexo	+	+	+	+	+	+	+	+	+	+	?	fédération

Commentaires :

(1) Pour UML, les profils s'en rapprochent peut-être...

(2) On peut probablement le faire en programmant...

Les outils de dessins ne permettent pas de travail de méta-modélisation ; ils restent au niveau de la syntaxe concrète. Les outils dédiés sont spécialisés pour traiter les modèles prévus, mais n'offrent pas d'outils pour changer ou élargir le point de vue. Les méta-éditeurs permettent d'enrichir les points de vue mais dans une approche méta vers instance et sans offrir d'outils de composition. *Openflexo* propose, à partir d'une analyse des besoins des situations de modélisation, un atelier plus complet de manipulation de modèles.

6. Travaux connexes

Dans (El Kouhen *et al.*, 2012), les ateliers *Generic Modeling Environment* (GME), *Rational Software Architect*, *MetaEdit+*, *Obeo Designer* et *Eclipse GMF* sont comparés selon 8 critères : niveau de personnalisation, expressivité graphique, complétude graphique, ouverture, utilisabilité, personnel requis, type de license, caractéristique des artefacts. La synthèse sur l'utilisabilité qui est au centre de notre article est :

« The best usability is offered by far by the Obeo Designer editor in terms of efficiency, accessibility, satisfaction and overall number of features. (...)The manipulation of elements is somewhat awkward in RSA and GME. »

Les comparaisons ne font pas référence à des situations de modélisation comme nous proposons de le faire, mais à des capacités ergonomiques. Les situations implicitement rencontrées sont la concrétisation, l'abstraction et la co-abstraction, et la multi-concrétisation.

Dans (Amyot *et al.*, 2006), les ateliers *Generic Modeling Environment* (GME), *Telelogic Tau G2*, *Rational Software Architect* (RSA), *XMF-Mosaic*, *Eclipse EMF+GEF* sont comparés selon 6 critères : complétude graphique, utilisabilité des éditeurs, efforts requis, évolution des langages, intégration et outils d'analyse et de transformation. Implicitement, les cas de concrétisation sont évoqués au travers de l'ergonomie, et la seule situation de modélisation explicitement évoquée est la co-évolution.

Dans (Kirchner et Jung, 2007), les méta-outils *MetaEdit+* et *Cubetto* sont comparés à l'aide d'une analyse multi-critères très simple prenant en compte : coût, dés-installation, capacité de modélisation avec des langages de modélisation prédéfinis, l'utilisation d'approches comme *Event-driven process chains* (EPC), *Petri nets* (PN), *Unified Modeling Language* (UML) et *Entity-Relationship-Model* (ERM), la capacité de simulation et l'usage de métriques. Il n'y a pas de référence explicite à des situations de modélisation autre que les cas de concrétisation et multi-concrétisation.

Un article de synthèse sur la méta-modélisation dans la conception et l'optimisation est réalisé dans (Wang et Shan, 2007). De nombreuses situations de modélisation sont identifiées : l'approximation de modèle, l'exploration de l'espace de conception, la formulation de problèmes, et la résolution de différents types de problèmes d'optimisation (que nous ne considérons pas car elles peuvent être outillées indépendamment). Mais la définition de la méta-modélisation assez déroutante : « The simple model is often called metamodel. »

À aucun moment, la grammaire ou le méta-modèle, au sens où nous l'entendons, n'est intégré dans le processus de modélisation ou de méta-modélisation. Pourtant une table faisant référence à des méta-modèles communément utilisés cite (entre autres) : *Polynomial (linear, quadratic, or higher)*, *Splines (linear, cubic, NURBS)*, *Knowledge Base or Decision Tree* qui sont bien des méta-modèles au sens où nous l'entendons. Les exemples de l'article décrivent des situations de type concrétisation et multi-concrétisation, interprétation et co-interprétation, et exemplification.

Enfin, dans (Muller *et al.*, 2010) les auteurs étudient d'autres relations entre modèles en formalisant différentes intentions dans le but de « faciliter le raisonnement et la discussion sur les méta-modèles et leurs relations » et de servir de base au développement d'ateliers de méta-modélisation « conscients » des intentions.

7. Conclusion

La modélisation est une démarche complexe, avant tout intellectuelle, mais qui peut être outillée. Le papier et le crayon (ou leur équivalent) restent des outils de base grâce à leur flexibilité et à la liberté qu'ils offrent. L'outillage informatique apporte des fonctionnalités supplémentaires qui permettent de vérifier des propriétés telles que la conformité. Pourtant, cet apport se fait au détriment de la liberté. Comment concilier vérification et liberté ? L'identification de situations de référence de modélisation peut servir de spécification d'usage pour des ateliers de modélisation. Nous avons observé :

- qu'il ne faut pas réduire la modélisation au dessin ; la notion de méta-modèle apporte des outils de vérification et peut guider lors de la concrétisation ;
- qu'un méta-modèle se construit et doit pouvoir évoluer ; on doit pouvoir aller au-delà de ce qui est prévu ;
- que méta-modèle et modèle ne sont pas nécessairement liés a priori, mais que c'est une activité de modélisation que d'identifier ces liens.

Modéliser requiert toutes les situations, combinées dans des ordres variés. Par exemple, on peut commencer par exemplifier, puis co-abstraire et interpréter, puis généraliser, pour exemplifier à nouveau afin de valider le méta-modèle. Un atelier de modélisation doit faciliter le travail à tous ces niveaux : intra-niveau et inter-niveaux.

La plupart des outils actuels sont trop contraignants – ils obligent à adopter un méta-modèle (ou un ensemble de méta-modèles) figé – ou trop complexes – et le travail sur le méta-modèle est une activité de programmation qui ramène souvent les concepts aux paradigmes des informaticiens.

L'infrastructure Openflexo propose une approche multi-niveaux par assemblage ou fédération de modèles. Il peut être utilisé comme simple outil de dessin, comme un outil de modélisation classique, mais aussi comme un outil de méta-modélisation pour élaborer un méta-modèle sans programmation et enfin pour mettre en relation – interpréter – des dessins et des méta-modèles.

Pour prolonger ce travail, nous envisageons une analyse plus fine des situations, en prenant en compte par exemple les différents acteurs, mais aussi des situations non pas constructives, mais destructives comme la suppression d'une interprétation ou d'un exemple. D'autres situations sont certainement intéressantes comme le changement de niveau évoqué dans la partie 3, quand un modèle devient un méta-modèle ou réciproquement, ou la composition de modèle évoquée dans cette même partie. Enfin, une étude précise sur des ateliers ou des outils de modélisation devrait être entamée.

8. Bibliographie

- Amaral V., Hardebolle C., Karsai G., Lengyel L., Levendovszky T., « Recent Advances in Multi-paradigm Modeling », *Models in Software Engineering*, vol. 6002 de *Lecture Notes in Computer Science*, p. 220-224, Springer, 2010.

- Amyot D., Farah H., Roy J.-F., « Evaluation of Development Tools for Domain-Specific Modeling Languages », *System Analysis and Modeling : Language Profiles*, vol. LNCS 4320, p. 183–197, Springer, Berlin, Heidelberg, 2006.
- Chauvel F., Jézéquel J.-M., « Code Generation from UML Models with Semantic Variation Points », Briand L., Williams C., Eds., *Model Driven Engineering Languages and Systems*, vol. 3713 de *Lecture Notes in Computer Science*, p. 54–68, Springer, 2005.
- El Kouhen A., Dumoulin C., Gerard S., Boulet P., « Evaluation of Modeling Tools Adaptation », rapport, juin 2012, Laboratoire d'Informatique Fondamentale de Lille (LIFL).
- Friendly M., Denis D. J., « Milestones in the history of thematic cartography, statistical graphics, and data visualization », Web document, <http://www.datavis.ca/milestones/>, 2001.
- Friendly M., « Milestones in the History of Data Visualization : A Case Study in Statistical Historiography », Weihs C., Gaul W., Eds., *Classification : The Ubiquitous Challenge*, p. 34–52, Springer, New York, 2005.
- Guychard C., Guerin S., Koudri A., Dagnat F., Beugnard A., « Conceptual interoperability through Models Federation », *Semantic Information Federation Community Workshop, Models Conference*, 2013.
- Kirchner L., Jung J., « A Framework for the Evaluation of Meta-Modelling Tools », *The Electronic Journal Information Systems Evaluation*, vol. 10, n° 1, 2007, p. 65–72.
- Kleppe A., « A Language Description is More than a Metamodel », *Fourth International Workshop on Software Language Engineering*, Grenoble, France, October 2007, megaplanet.org.
- Lachapelle C., Cunningham C., « Engineering Is Elementary : Children's Changing Understandings of Engineering and Science », *American Society for Engineering Education Annual Conference & Exposition*, Honolulu, HI, June 2007.
- Moody D., « The “physics” of notations : toward a scientific basis for constructing visual notations in software engineering », *IEEE Transactions on Software Engineering*, vol. 35, n° 6, 2009, p. 756–779.
- Morris W. T., « On the art of modeling », *Management Science*, vol. 13, n° 12, 1967, p. B707–B717.
- Muller P.-A., Fondement F., Baudry B., « Modeling modeling modeling », *Software and Systems Modeling*, , 2010.
- « UML 2.0 Superstructure Specification », <http://www.omg.org/cgi-bin/doc?ptc/2003-08-02>, 2007.
- Peyraube A., Bonvini E., Busuttill J., *Dictionnaire des langues*, Presses Universitaires de France, 2010.
- Schekkerman J., « A Comparative Survey Of Enterprise Architecture Frameworks », rapport, 2004, Institute For Enterprise Architecture Development/Capgemini.
- Sprinkle J., Karsai G., « A domain-specific visual language for domain model evolution », *Journal of Visual Languages and Computing*, vol. 15, n° 3, 2004, p. 291–307.
- Wang G. G., Shan S., « Review of metamodeling techniques in support of engineering design optimization », *Journal of Mechanical Design*, vol. 129, 2007, page 370.