HAL
open science

Ingénierie des processus : une approche à base de patrons

Charlotte Hug, Agnes Front, Dominique Rieu

To cite this version:

Charlotte Hug, Agnes Front, Dominique Rieu. Ingénierie des processus : une approche à base de patrons. Inforsid, May 2007, Perros-Guirec, France. pp.471-486. hal-00671906

HAL Id: hal-00671906

https://hal.science/hal-00671906v1

Submitted on 19 Feb 2012

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.

Ingénierie des processus : une approche à base de patrons

Charlotte Hug — Agnès Front — Dominique Rieu

LIG, équipe SIGMA – Université de Grenoble
BP 72 38402 Saint Martin d'Hères Cedex
{Prenom.Nom}@imag.fr

[Chercheurs]

RÉSUMÉ. *Il existe de nombreux modèles de processus dédiés à l'ingénierie des systèmes d'information. Ils sont basés sur des concepts (activité, produit,...), et donc sur des méta-modèles de processus différents mais complémentaires. En ce sens, ces méta-modèles ne permettent qu'une vision partielle des processus d'ingénierie. Nous proposons dans cet article un patron de conception qui permet de modéliser la plupart des concepts de méta-modèles de processus. Ce patron constitue une première étape vers la construction de méta-modèles de processus unifiés et multi-vues.*

ABSTRACT. *A lot of process models for information systems engineering exist. They are based on different concepts (activity, product...) and then on different process metamodels. Those metamodels allow a partial vision of the process engineering. This paper presents a design pattern to model concepts of various process metamodels. This pattern is a first step towards the construction of unified and multi-views process metamodels.*

MOTS-CLÉS : *modèle de processus, ingénierie des processus, patron, méta-modélisation.*

KEYWORDS : *process model, process engineering, pattern, meta-modelling .*

1. Introduction

Les modèles de processus proposent des démarches de développement de systèmes logiciels. Ils prescrivent une démarche méthodologique pour atteindre la cible que constituent les produits (Rolland, 2005). Il est essentiel de permettre aux ingénieurs de créer ou d'adapter des modèles de processus en fonction des contraintes et caractéristiques des projets. De plus, la définition d'un modèle de processus doit être dirigée par des concepts, des règles et des relations, un méta-modèle est donc nécessaire pour l'instanciation de modèles.

Un modèle de processus, comme un modèle de produit, doit pouvoir être perçu suivant différents points de vue et différents niveaux d'abstraction. Or la plupart des méta-modèles de processus ne permettent d'avoir qu'une modélisation monolithique des modèles qu'ils décrivent. Ils sont, de plus, souvent trop précis ou spécifiques pour être adaptables au vocabulaire d'une organisation. La majorité d'entre eux ne propose pas de mécanismes d'extension.

Pour permettre une vision globale et complète du processus d'ingénierie des systèmes d'information, il est donc nécessaire de disposer de méta-modèles de processus unificateurs, multi-vues et adaptables. L'objectif de cet article est de proposer un noyau de base pour la création de tels méta-modèles.

Cet article est organisé comme suit : la section 2 définit les différents niveaux de modélisation. La section 3 discute les modèles et méta-modèles de processus existants et expose la problématique. La section 4 présente le patron « Item-Description » de (Coad, 1992) et l'instanciation profonde (Atkinson *et al.*, 2001) qui servent de base à notre proposition de patron pour la méta-modélisation des processus. La section 5 montre comment le patron proposé facilite la définition de méta-modèles de processus. Enfin, la section 6 conclut l'article et présente les perspectives de notre travail.

2. Les niveaux de modélisation

Il existe de nombreuses définitions d'une méthode d'ingénierie de systèmes d'information. Toutes mettent en évidence la nécessité d'offrir des langages permettant de modéliser des produits et une démarche correspondante. La double facette produit/processus replacée dans l'architecture de l'OMG (OMG, 2002) permet de définir une méthode comme étant composée :

– d'un méta-modèle de produit : par exemple, UML.

– d'un modèle de processus : par exemple, le modèle de processus de la méthode RUP.

Dans cet article, nous nous concentrons sur les processus. L'architecture à quatre niveaux définie par l'OMG pour les processus est détaillée ci-après (cf. Figure 1). Le méta-méta-modèle permet de définir un méta-modèle de

processus. «Activité», « Acteur » et « Ressource » sont des méta-classes[1], instances de la méta-méta-classe[2] « Classifier ». Le modèle instancie le méta-modèle pour présenter un processus : l'analyse des processus métier, réalisée par un analyste, produit un modèle de processus métier. « Analyse des processus métier » et «Description informelle du processus métier» sont des instances de la méta-classe «Activité». Enfin, le niveau instance représente le processus réel dans le cadre du développement d'un système d'information en particulier.

Le méta-modèle de processus de l'OMG (niveau M2), SPEM, se focalise sur un point de vue des processus particulier nommé activité. Les modèles de processus instanciés à partir de SPEM sont donc aussi orientés activité. Il existe d'autres catégories de modèles de processus que nous présentons dans la section suivante.

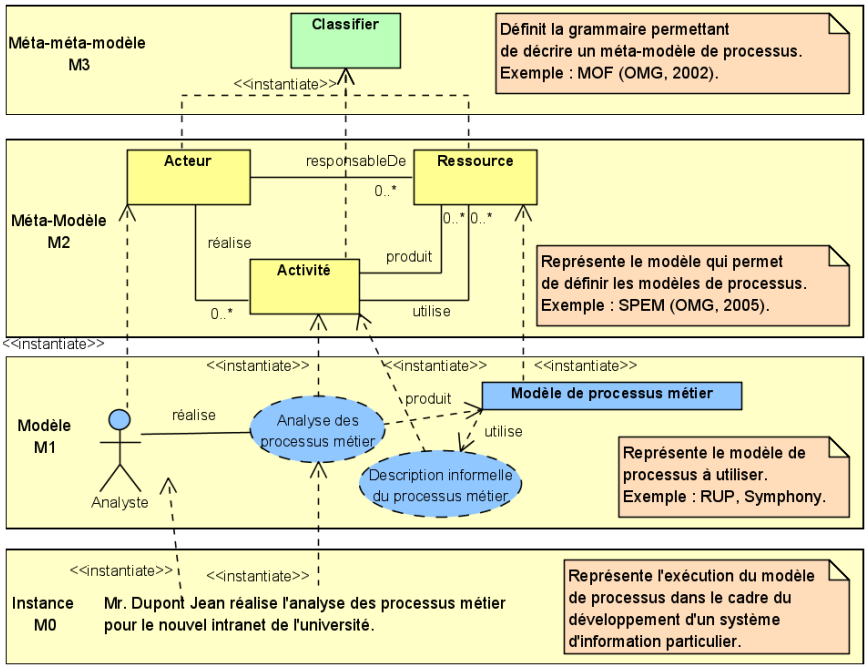


Figure 1. *Exemple d'architecture à quatre niveaux pour les processus.*

3. Les modèles et méta-modèles de processus existants

D'après (Dowson, 1987), un modèle de processus peut se focaliser sur :

– les activités,

1. M2-level class (OMG, 2002).
2. M3-level class (OMG, 2002).

– les objets qui résultent de ces activités (les produits),

– les décisions qui engendrent le traitement des activités et des produits.

D'autres catégories de modèles de processus ont été définies :

– les modèles de processus orientés contexte (Rolland *et al.*, 1995),

– les modèles de processus orientés stratégie (Rolland *et al.*, 1999).

Dans les différentes approches (sections 3.1 à 3.5) :

– les modèles de processus (niveau M1) sont représentés par des classes,

– les méta-modèles de processus (niveau M2) sont représentés par des stéréotypes.

3.1. *Modèles de processus orientés activité*

Les modèles de processus orientés activité représentent les activités et leur ordonnancement pour la réalisation d'un produit. Ce sont des modèles comme : RUP (IBM-Rational, 1998), Symphony (Umanis-LSR-Sigma, 2002) et 2TUP (Valtech, 2003) par exemple.

De nombreux méta-modèles permettent de définir des modèles de processus orientés activité : SPEM (OMG, 2005), l'OPEN Process Framework[3], OOSPICE[4],...

Le modèle de processus Symphony est une instance du méta-modèle de processus SPEM. Par exemple (cf. Figure 2), « Analyse », « Architecture technique » ou « Conception » sont des phases de Symphony (Hassine, 2005). Elles sont instances de la classe « Phase » telle que SPEM la définit. De même, «Cycle de vie en Y», qui est le cycle de vie de Symphony est instance de «LifeCycle» défini dans SPEM.

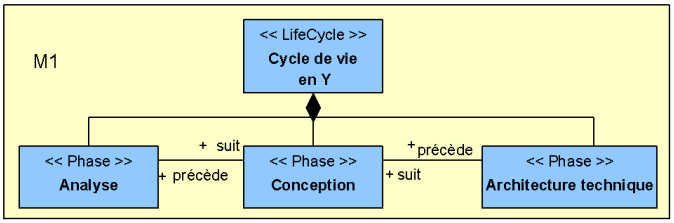


Figure 2. *Extrait du modèle de processus orienté activité Symphony.*

3. http://www.donald-firesmith.com/
4. http://www.oospice.com

3.2. *Modèles de processus orientés produit*

Les modèles de processus orientés produit couplent l'état du produit à l'activité qui génère cet état. Ils sont assimilables à des diagrammes état-transition (Rolland, 2005).

Il existe différents méta-modèles de processus comme les statecharts de Harel (Harel, 1987), le méta-modèle de processus Entity (Humphrey *et al.*, 1989), et le diagramme état-transition d'UML (ou state machines) (OMG, 2006).

Un méta-modèle de processus définit comment un produit passe d'un état à un autre, c'est-à-dire par quelle transition.

La Figure 3 montre, par exemple, la transition permettant de passer d'un produit en cours de développement à un produit en cours de validation.

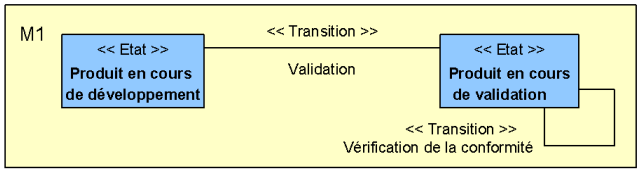


Figure 3. *Extrait d'un modèle de processus orienté produit.*

3.3. *Modèles de processus orientés décision*

Les modèles de processus orientés décision présentent les transformations ou les élicitations successives du produit dues à des décisions (Rolland, 2005).

CAD° (Conversations among Agents on Decisions over Objects) du projet DAIDA (Jarke *et al.*, 1992) inspiré de (Potts *et al.*, 1988), et IBIS (Kunz *et al.*, 1970) sont des méta-modèles de processus orientés décision.

Par exemple, le méta-modèle de processus orienté décision de Potts permet de représenter des alternatives répondant à un problème, les arguments sur lesquels ces alternatives sont basées et les produits concernés par ce problème. La Figure 4 montre un modèle de processus orienté décision concernant la création d'une entité dans un schéma entités/associations.

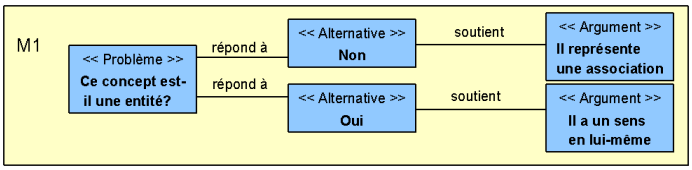


Figure 4. *Extrait d'un modèle de processus orienté décision.*

3.4. *Modèles de processus orientés contexte*

Les modèles de processus orientés contexte prennent en compte l'intention et la situation d'un acteur (analyste, ingénieur des méthodes...) à un instant donné du projet (Rolland, 2005).

Cette catégorie de modèles de processus a été introduite lors du projet européen NATURE, un méta-modèle du même nom a été défini (Rolland *et al.*, 1995).

Dans le méta-modèle NATURE, le contexte est composé d'une situation (par rapport à un produit) et d'une intention (une cible par rapport à un produit). Un contexte peut être exécutable (il engendre une action sur un produit), choix (il permet de choisir entre plusieurs contextes) ou plan (il impose une succession de contextes). Dans l'exemple de la Figure 5 la racine de l'arbre des contextes est composée d' une situation vide et d'une intention qui est de définir le modèle de processus.

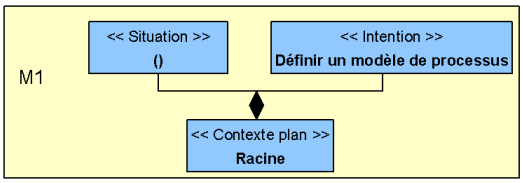


Figure 5. *Extrait d'un modèle de processus orienté contexte.*

3.5. *Modèles de processus orientés stratégie*

Les modèles de processus orientés stratégie permettent de représenter les processus multi-démarches et prévoient plusieurs chemins possibles pour élaborer le produit en se basant sur les notions d'intention et de stratégie (Rolland, 2005).

MAP est, à notre connaissance, le seul méta-modèle de processus orienté stratégie, il a été défini par (Rolland *et al.*, 1999). Ce méta-modèle représente des intentions pouvant être atteintes selon différentes stratégies. Il a été utilisé pour modéliser la méthode d'ingénierie des besoins CREWS-L'Écritoire (Rolland *et al.*, 1999). Par exemple, « Éliciter un but » et « Écrire un scénario » correspondent à des intentions de CREWS (cf. Figure 6). « Start » et « Stop » sont les intentions de départ et d'arrivée. La stratégie « Prose libre » permet de passer de l'intention « Éliciter un but » à « Écrire un scénario ».

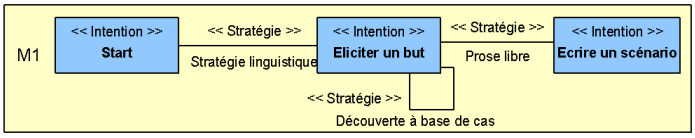


Figure 6. *Extrait du modèle de processus orienté stratégie, CREWS- l'Écritoire.*

3.6. *Synthèse*

Les points précédents montrent qu'il existe une multitude de modèles et de méta-modèles de processus représentant différents points de vue (orientés activité, produit, etc.). De plus, certains points de vue se situent à un niveau d'abstraction opérationnel (produit et activité) alors que d'autres se situent à un niveau plus intentionnel (stratégie, contexte et décision). Ces points de vue ont des relations entre eux ; par exemple, un modèle de processus orienté activité peut être vu comme un raffinement d'un modèle de processus orienté stratégie. De plus, aucun méta-modèle ne couvre tous les points de vue : chacun est positionné sur un point de vue particulier. SPEM, s'il permet de définir correctement des modèles de processus orientés activité, n'est, par exemple, pas suffisant pour définir des modèles de processus orientés produit.

D'autre part, certains méta-modèles de processus sont trop spécifiques, les modèles de processus instanciés ne sont pas adaptables au vocabulaire d'une organisation. Par exemple, SPEM implique l'utilisation d'un langage trop spécifique en terme de concepts (phase, cycle de vie, itération et activité).

Enfin, la plupart des méta-modèles n'offre pas de mécanismes d'adaptation, sauf SPEM qui est extensible grâce au mécanisme de profils de UML.

Nous pensons que dans le cadre d'une méthode d'ingénierie des systèmes d'information, voire de son adaptation dans une organisation, les différents points de vue ne sont pas antagonistes mais complémentaires. Par exemple, l'utilisation d'un modèle de processus orienté activité devrait être complémentaire avec celle d'un modèle de processus orienté produit. Il faut pouvoir suivre quelle activité influe sur quel produit, comment et à quel moment du processus. Il est donc nécessaire d'unifier les différents méta-modèles pour faciliter l'utilisation des points de vue multiples.

De plus, pour répondre au problème des langages trop spécifiques, une solution consiste à proposer un méta-modèle de processus pouvant être facilement adaptable à des projets ou des organisations, par ajout et suppression de concepts (activité, produit,...). Traditionnellement, dans un méta-modèle, un concept est modélisé par une méta-classe principale liée à d'autres méta-classes par des associations (par exemple, une activité produit et consomme des ressources, cf Figure 1). Certains concepts ne peuvent cependant pas être modélisés par une seule méta-classe.

Dans cet article, nous proposons un patron permettant de catégoriser des concepts dotés de propriétés spécifiques et partageant des propriétés communes et ceci à deux niveaux :

– au niveau des modèles de processus (M1) pour valuer des propriétés applicables à chaque concept du méta-modèle de processus (par exemple, le fait

que chaque phase puisse être optionnelle ou non) et à l'ensemble des concepts d'une catégorie (par exemple, le fait que toute phase soit composée d'activités).

– au niveau du processus (M0) pour valuer des propriétés applicables à chaque occurrence de concept du modèle de processus (par exemple, la durée d'une phase d'analyse pour un projet donné) et à l'ensemble des phases d'un projet (par exemple, la durée maximale d'une phase).

Le patron proposé est donc une première étape vers la modélisation de méta-modèles de processus unifiés et multi-vues. Nous ne traitons dans cet article que l'aspect « unificateur » et détaillerons l'aspect « multi-vues » dans un article ultérieur.

4. Un patron pour la méta-modélisation des processus

Dans un premier temps nous exposons les concepts et les catégories de concepts retenus pour la construction d'un méta-modèle de processus, puis nous présentons le patron « Item-Description » sur lequel nous nous basons pour construire notre patron appelé « Concept-Catégorie de concepts ». Nous expliquons ensuite l'instanciation profonde qui est utilisée dans le patron « Concept-Catégorie de concepts » et enfin, nous présentons ce dernier dans le formalisme P-SIGMA (Conte *et al., 2001).*

4.1. *Des concepts et des catégories de concepts*

Comme nous l'avons vu dans la section précédente, il faut distinguer les concepts de leur catégorie. Un certain nombre de concepts ont été identifiés. Notre solution prend en compte chacun de ces concepts, ainsi que leur catégorie : unité de travail et catégorie d'unité de travail (le terme activité utilisé précédemment nous paraît trop limitatif, une activité est une sorte d'unité de travail, au même titre qu'une phase), produit et catégorie de produit, agent et catégorie d'agent, décision et catégorie de décision, contexte et catégorie de contexte, stratégie et catégorie de stratégie, contrainte et catégorie de contrainte, guidage et catégorie de guidage...

Cette association d'un concept à sa catégorie de concepts évoque le problème résolu par le patron « Item-Description ».

4.2. *Le patron « Item-Description »*

Ce patron a été introduit par Peter Coad dans (Coad, 1992). Il est utilisé lorsque des valeurs d'attributs sont attribuées à plusieurs objets instances d'une même classe. La classe « ItemDescription » (telle que représentée par (Gzara, 2000) dans la Figure 7) possède des valeurs d'attributs qui s'appliquent à plusieurs Items. Un « Item » a ses propres valeurs d'attributs et méthodes pour accéder aux valeurs d'attributs de la classe « ItemDescription ». La Figure 8 montre une imitation de ce patron pour des voitures et leur modèle.

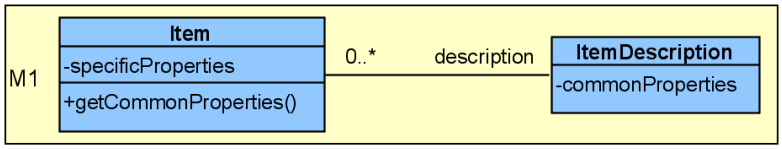


Figure 7. *Le patron « Item-Description ».*

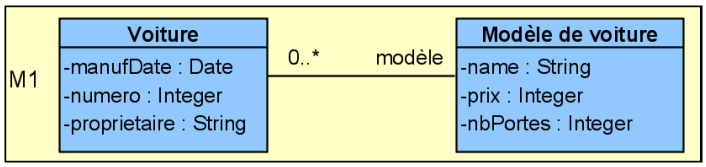


Figure 8. *Imitation du patron.*

4.3. *L'instanciation profonde*

Le patron « Item-Description » a été initialement proposé pour construire des modèles de produit (niveau M1) d'une application (cf. Figure 8). Nous utilisons le patron « Item-Description » pour la modélisation des processus sur trois niveaux (cf. Figure 9) :

– niveau M2 : pour représenter le méta-modèle de processus,

– niveau M1 : pour représenter le modèle de processus,

– niveau M0 : pour représenter le projet, c'est à dire l'exécution du processus.

La classe Phase (une catégorie d'unité de travail parmi d'autres) instancie les propriétés de la méta-classe « Catégorie d'unité de travail -M2 » et hérite des propriétés de la classe «Catégorie d'unité de travail-M1 ». Les propriétés héritées (par exemple duréeMax,...) sont instanciées au niveau M0 par P1 (modélisant l'ensemble des phases d'un projet, dans l'exemple toutes les phases auront une duréeMax égale à 150 jours).

La classe « Analyse » (une unité de travail) instancie les propriétés de la méta-classe « Unité de Travail -M2 » et hérite des propriétés de la classe « Unité de travail -M1 ». Les propriétés héritées (par exemple durée) sont instanciées au niveau M0 par A1, modélisant une phase d'analyse dans un projet donné (dans l'exemple, A1 a une durée de 50 jours).

Afin de permettre l'instanciation d'attributs sur des niveaux différents, la Figure 9 propose de combiner instanciation et héritage. Cette notion (héritage et instanciation d'attributs de méta-classes) est très proche de la notion de Powertype (Odell, 1994). L'instanciation profonde est cependant retenue pour alléger les schémas de modélisation.

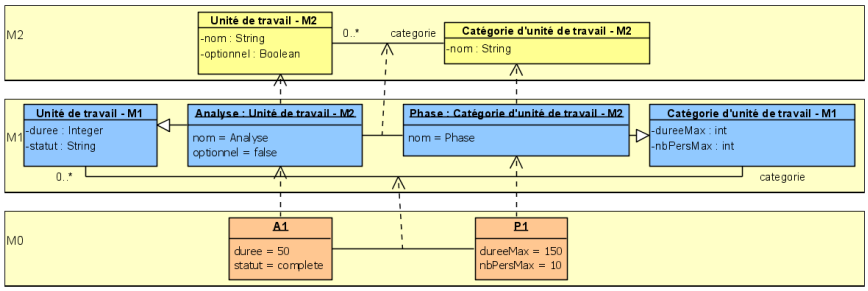


Figure 9. *Instanciation et héritage d'attributs de méta-classes.*

L'instanciation profonde (« Deep Instantiation ») a été introduite par (Atkinson *et al.*, 2001). Elle permet d'instancier un attribut ou une association au niveau de modélisation que l'on souhaite et non plus au niveau strictement inférieur. L'instanciation profonde consiste à affecter un exposant à un attribut ou à une association. Chaque fois que la classe à laquelle l'attribut appartient est instanciée, l'exposant est retranché de 1. Tant que l'exposant est supérieur à 0, l'attribut n'est pas instancié. Lorsque l'exposant de l'attribut est égal à 0, l'attribut est instancié. La notation adoptée pour les attributs est la suivante :

– « attribut : type », lorsque l'attribut a un exposant égal à 1 (P=1). Il est instancié normalement au niveau inférieur de modélisation.

– « attribut^{P} : type », lorsque l'attribut a un exposant supérieur à 1 (P>1). Il est instancié lorsque P=0, au P^{ième} niveau inférieur de modélisation.

– « attribut=valeur », lorsque l'attribut a un exposant égal à 0 (P=0), il est alors instancié.

Les associations ont un exposant qui est décrémenté lorsque les classes sont instanciées. Lorsque l'exposant vaut zéro, un lien est créé entre les objets.

La Figure 10 présente l'adaptation du patron « Item-description » pour la méta-modélisation de processus, pour unité de travail et catégorie d'unité de travail et leur instanciation, en utilisant le mécanisme d'instanciation profonde. Au niveau du méta-modèle (M2), l'attribut «durée» de la classe « Unité de

travail » a un exposant 2. Cela signifie que cet attribut est instancié au deuxième niveau inférieur, c'est à dire au niveau du processus (M0).

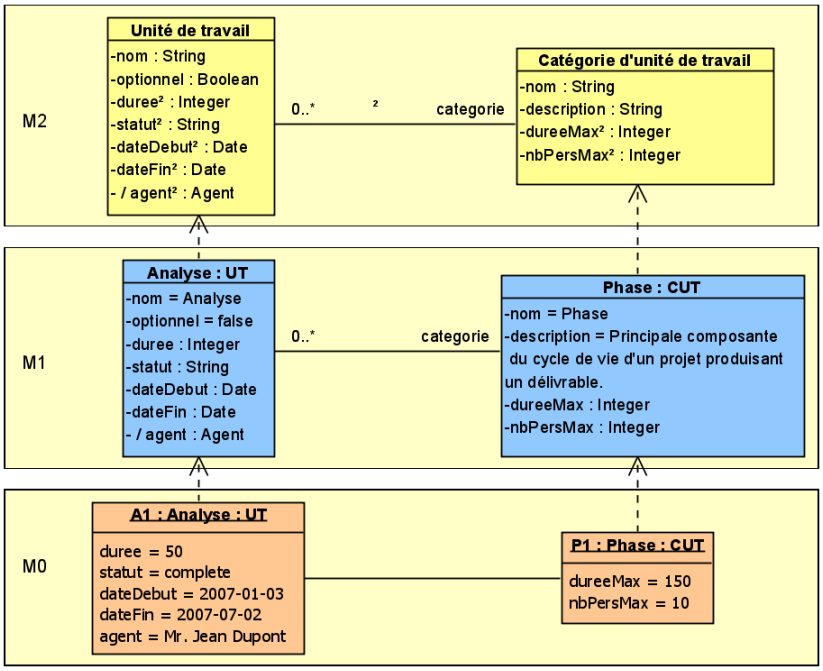
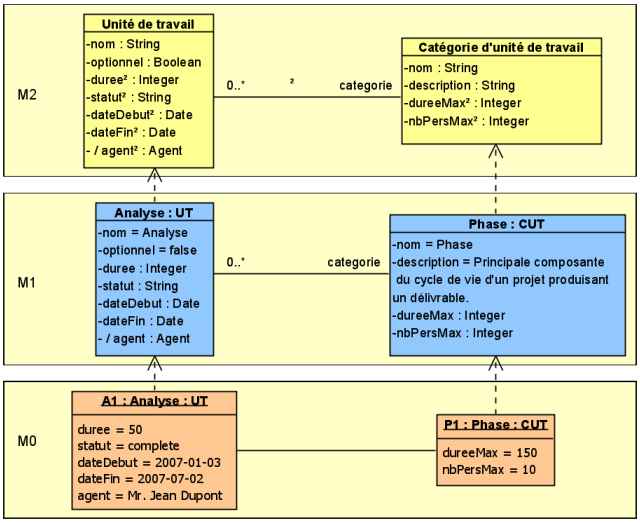
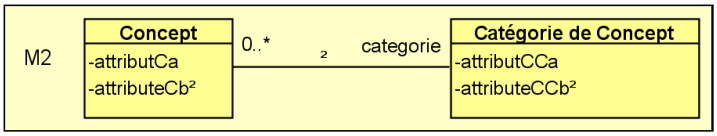


Figure 10. *Utilisation de l'instanciation profonde pour la méta-modélisation des processus.*

4.5. *Le patron « Concept-Catégorie de Concepts »*

Le patron « Concept-Catégorie de concepts » présenté dans le tableau ci-dessous est une adaptation du patron « Item-Description » pour la méta-modélisation de processus et utilise notamment l'instanciation profonde.

Nom	Concept – Catégorie de concepts
Problème	Permet de catégoriser des concepts dotés de propriétés spécifiques et partageant des propriétés communes à deux niveaux de modélisation.

Motivation	Nous souhaitons définir le concept d'unité de travail. Ce concept doit être décliné sous la forme de deux méta-classes de niveau M2: – Catégorie d'unité de travail définit les propriétés instanciables par les catégories d'unités de travail (par exemple phase) et les instances des catégories d'unité de travail (P1 dans la figure). – Unité de travail définit les propriétés instanciables par les unités de travail (par exemple analyse) et les instances des unités de travail (A1 dans la figure). 
Solution-modèle	 Les attributs « attributCa » et « attributCCa » sont classiquement instanciés au niveau M1. Les attributs « attributCb² » et « attributCCb² » et l'association entre « Concept » et « Catégorie de Concept » bénéficient de l'instanciation profonde et sont instanciés au niveau M0.
Force	Ce patron permet de partitionner les concepts d'un méta-modèle de processus en deux classes : les concepts et les catégories de concepts. Les imitations de concept et catégorie de concepts font parties du méta-modèle de processus. Leurs instances définissent les éléments faisant partie des modèles de processus et des catégories de modèles de processus auxquels ils se rattachent.

5. Ingénierie des processus par imitation du patron « Concept-Catégorie de Concepts »

Nous présentons ici un exemple d'un méta-modèle de processus unificateur et adaptable, puis un exemple partiel d'instanciation de ce méta-modèle pour SPEM (OMG, 2005) et Symphony (Hassine, 2005), qui est conforme à SPEM.

5.1. *Exemple de méta-modèle de processus*

Nous présentons ici un exemple de méta-modèle de processus qui contient trois imitations du patron « Concept- Catégorie de Concepts » pour les concepts unité de travail et catégorie d'unité de travail, produit et catégorie de produit, agent et catégorie d'agent.

Une catégorie d'agent peut participer à une catégorie d'unité de travail, de même qu'un agent peut participer à une unité de travail. Une unité de travail peut avoir des produits en entrée ou en sortie. Chaque produit a un ou plusieurs agents responsables. Des unités de travail peuvent s'effectuer en séquence ou en parallèle.

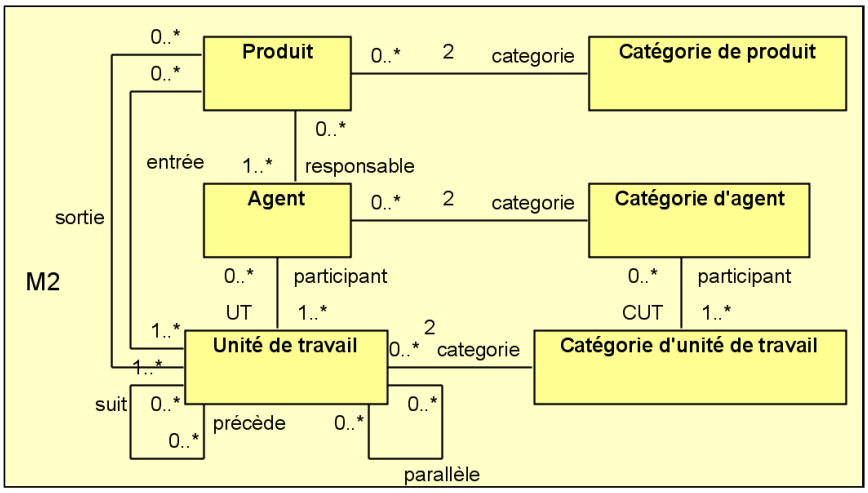


Figure 11. *Extrait d'un méta-modèle de processus : trois imitations du patron.*

5.2. *Instanciation pour SPEM et Symphony*

La Figure 12 illustre les concepts de Symphony (par exemple étude préalable, analyste des processus métiers, modèle de processus métiers) et leur catégorie de concepts définie dans SPEM (par exemple Phase, ProcessRole et WorkProduct).

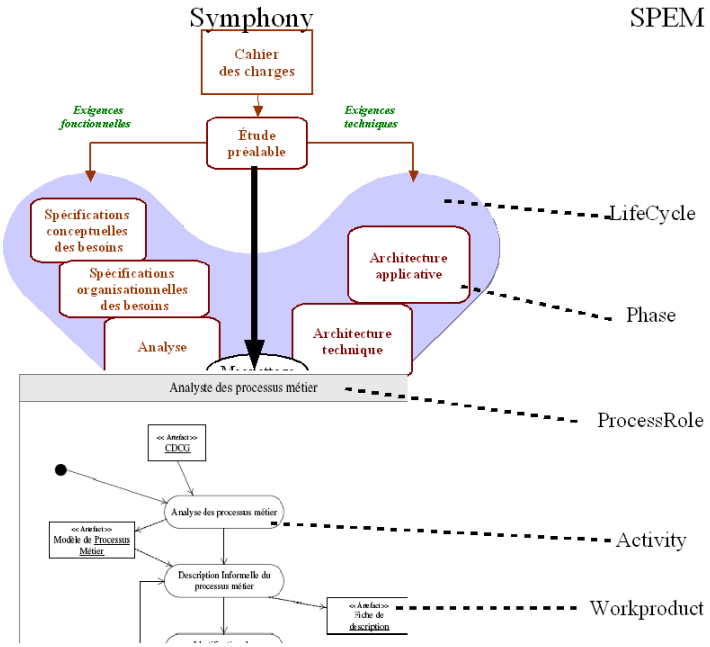


Figure 12. *Illustration des catégories de concepts de SPEM et des concepts de Symphony.*

La Figure 13 illustre la construction de SPEM et de Symphony à partir du méta-modèle de processus de la Figure 11. Pour plus de clarté, les classes de cet exemple ne contiennent ni attribut ni méthode.

Dans un premier temps, il est nécessaire d'instancier les catégories de concepts qui constituent la catégorie de modèle de processus, ici SPEM. Lorsque SPEM est instancié, nous pouvons instancier les concepts qui sont liés à une catégorie de concepts. L'ensemble de ces concepts forme le modèle de processus. Ici, nous avons instancié certains concepts de Symphony (plus clairs sur la figure). Les concepts de Symphony sont rattachés aux catégories de concepts de SPEM. Par exemple, le concept « Étude préalable » est rattaché à la catégorie de concepts « Phase ».

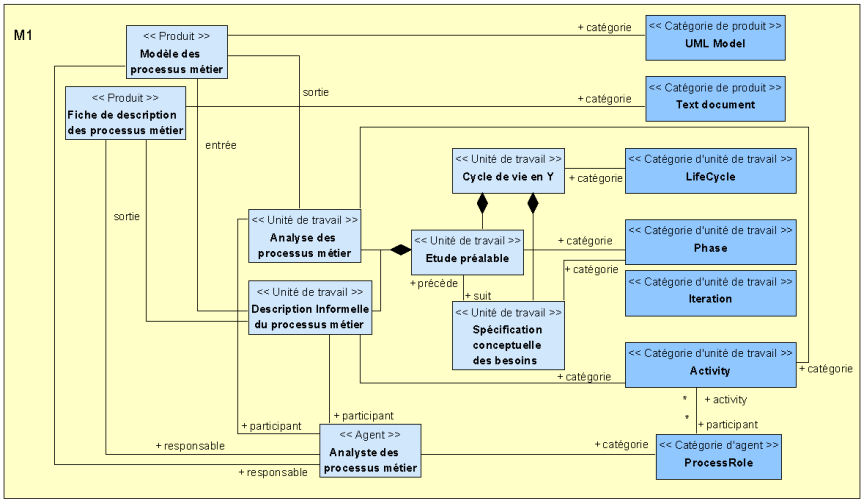


Figure 13. *Construction d'une partie de SPEM et de Symphony avec le méta-modèle de processus.*

6. Conclusion

Dans cet article nous avons montré que les méta-modèles de processus étaient nombreux, monolithiques et peu adaptables. Ils ne suffisent pas à présenter un processus d'ingénierie des systèmes d'information sous tous les points de vue possibles. Nous avons proposé le patron « Concept-Catégorie de Concepts » qui permet, par imitation, de créer des méta-modèles de processus unifiés.

Par la suite, il est nécessaire de détailler les associations entre les différentes catégories de concepts et les différents concepts: elles seront spécifiées sous forme de patrons d'associations. D'autres patrons seront proposés pour modéliser d'autres formes de concepts. Ensuite, il faudra introduire les notions de points de vue et de niveaux d'abstraction que nous avons évoquées rapidement dans cet article. Enfin, il s'agira de proposer un outil basé sur un catalogue de concepts (imitations du patron « Concept-Catégorie de Concepts ») permettant de modéliser facilement des méta-modèles de processus unificateurs et multi-vues.

7. Bibliographie

Atkinson C., Kühne T., « The Essence of Multilevel Metamodeling », *UML 2001 4th International Conference*, Toronto, Canada, October 1-5, 2001, Lecture Notes in Computer Science, vol. 2185 , p. 19-33.

Coad P., « Object-Oriented Patterns », *Communication of the ACM*, vol. 35, n° 9, 1992, p. 152-159.

Dowson M., « Iteration in the software process », *Review of the 3rd International Software Process Workshop*, Proceedings of the 9th International Conference on Software Engineering, Monterey, CA, USA 1987, ACM Press, p. 36-39.

Front-Conte A., Fredj M., Giraudin J.-P., Rieu D., « P-Sigma : un formalisme pour une représentation unifiée de patrons », *Actes du XIXème Congrès INFORSID*, Martigny, Suisse, 24-27 mai, 2001 , p. 67-86.

Gzara L., Les patterns pour l'ingénierie des systèmes d'information, Thèse de l'INPG, décembre 2000.

Harel D., « Statecharts: A Visual Formulation for Complex Systems », *Science of Computer Programming*, vol. 8, n° 3, 1987, p. 231-274.

Hassine I., Spécification et formalisation des démarches de développement à base de composants métier : la démarche Symphony, Thèse de l'INPG, septembre 2005.

Humphrey W. S., Kellner M. I., « Software Process Modeling: Principles of Entity Process Models », *ICSE*, Pittsburg, PA, USA, May 15-18 1989, p. 331-342.

Jarke M., Mylopoulos J., Schmidt J. W. , Yannis Vassiliou, « DAIDA: An Environment for Evolving Information Systems », *ACM Transactions on Information Systems*, January 1992, vol. 10, n° 1, p. 1-50.

Kunz W., Rittel H.W.J., « Issues as elements of information systems », Working Paper n° 131, Heidelberg-Berkeley, 1970

Odell J., « Power Types », *Journal of Object-Oriented Programming,* vol. 7, n° 2, May 1994, p. 8-12.

OMG, MetaObjectFacility (MOF) Specification, Version 1.4, April 2002.

OMG, Software Process Engineering Metamodel Specification, Version 1.1, January 2005.

OMG, Unified Modeling Language: Superstructure, Version 2.1, April 2006.

Potts C., Bruns G., « Recording the Reasons for Design Decisions », *International Conference on Software Engineering,* Singapore, April 11-15, 1988, p. 418-427.

Rolland C., « L'ingénierie des méthodes : une visite guidée », *e-TI*, n° 1, (http://www.revue-eti.net/document.php?id=726), 25 octobre 2005.

Rolland C., Prakash N., Benjamen A., « A Multi-Model View of Process Modelling », *Requirements Engineering,* Springer-Verlag London Limited, vol. 4, n° 4, 1999, p. 169-187.

Rolland C., Souveyet C., Moreno M., « An Approach for defining ways-of-working », *Information System Journal*, vol. 20, n° 4, 1995, p. 337-359.