Validation sémantique d'objets à l'aide d'un modèle de référence et de contraintes

Cyril Faucher* — Samnang Chea* — Frédéric Bertrand* — Jean-Yves Lafaye*

** L3i, Université de La Rochelle,*

avenue Michel Crépeau, 17042 La Rochelle cedex1

http://l3i.univ-larochelle.fr/

{cyril.faucher, samnang.chea, frederic.bertrand, jean-yves.lafaye}@univ-lr.fr

RÉSUMÉ. *Nous montrons dans cet article l'usage d'un modèle de référence capturant la connaissance d'un domaine afin de permettre une validation de la conformité sémantique d'un corpus d'instances. L'Ingénierie Dirigée par les Modèles est au cœur du processus de validation afin de naviguer et de mettre en relation le métamodèle du domaine et les modèles sous-jacents. Nous l'appliquons au domaine temporel pour la saisie d'expressions temporelles périodiques dans une application Web.*

ABSTRACT. *In this paper, we account for the use of a set of reference domain (meta)models in order to check the semantic consistency of a set of instances. MDE techniques constitute the core of the validating process and allow a synchronized browsing of the domain metamodel and its related models. We come with a real case study within the field of temporal, periodical information management in web applications.*

MOTS-CLÉS : *Validation, sémantique, IDM.*

KEYWORDS: *Validation, semantic, MDE.*

1. Introduction

Les modèles objets fondés sur le concept de classe nous assurent une validité structurelle des objets instanciés. Cependant, selon la complexité du domaine métier, beaucoup d'efforts sont nécessaires pour mettre en œuvre une validation sémantique de l'ensemble des instances. Dans cet article, nous montrons de quelle manière cette validation peut être facilitée par l'utilisation d'un modèle capturant la sémantique du domaine métier.

Un métamodèle permet de définir des concepts et leurs relations. Il est possible de créer un modèle conforme au métamodèle et de vérifier son intégrité en termes de typage et de cardinalités. Ces contraintes de construction sont souvent insuffisantes pour garantir la correction sémantique du modèle. OCL peut être utilisé pour compléter le système de contraintes assurant l'intégrité, mais s'avère insuffisant lorsque la sémantique des modèles devient trop complexe. En effet, nous souhaitons

offrir à l'utilisateur la possibilité, soit de compléter, soit de relâcher des contraintes sémantiques. Dans ce cas, l'utilisation d'un langage comme OCL n'est pas adaptée. Nous prenons comme exemple un domaine modélisant des expressions temporelles périodiques. Le métamodèle du domaine permet de gérer des expressions telles que *« un événement a lieu tous les 1er jours de chaque semaine »*. La correction syntaxique des modèles n'est pas suffisante pour valider sémantiquement une expression. Par exemple, l'expression *« tous les 8ème jours de la semaine »* représente un modèle conforme à son métamodèle mais est sémantiquement incorrect. En effet, le rang d'un jour au sein d'une semaine est compris entre 1 et 7. Ici la valeur comprise entre 1 et 7 n'est spécifiée à l'aide d'une multiplicité mais par une valeur d'un attribut du modèle de classes. Ainsi les contraintes de cardinalité ne sont pas ici exploitables. D'autres exemples plus complexes motivent notre approche comme s'assurer que l'expression *« chaque lundi de chaque semaine »* précède *« chaque mardi de chaque semaine »*. Un exemple de complément pour une règle peut être que les années bissextiles ont lieu dans un premier temps *« tous les 4 ans »* et dans un second temps *« sauf tous les 100 ans mais tous les 400 ans »*. Le langage pour décrire ces règles doit être proche de celui utilisé par l'expert du domaine.

La suite de cet article est organisée en quatre sections. Pour illustrer notre démarche, la section suivante présente un métamodèle d'expressions temporelles périodiques. La section 3 s'attache à montrer l'apport d'un modèle de référence contenant la sémantique du calendrier Grégorien. La section 4 présente la validation de modèles à l'aide d'un modèle de référence et de contraintes. La section 5 présente nos conclusions et perspectives pour ces travaux.

2. Le métamodèle temporel

Le métamodèle temporel (cf Figure 1) décrit des concepts pour structurer des expressions temporelles décrites en intension (Carnap 1947) rattachées à des événements périodiques. Notre métamodèle n'a pas pour objectif de modéliser des propriétés d'ordonnancement de systèmes comme le permettent les CCSL[1], MARTE[2], etc. Il permet de modéliser les périodes d'occurrences de phénomènes tels que des événements touristiques. Ce métamodèle étend la norme ISO 19108[3] en réutilisant les concepts d'instant, de période et les relations d'Allen (Allen 1983). Evenement est la classe principale du métamodèle, elle possède des règles périodiques qui s'expriment, soit en terme de fréquence comme *« 1 fois par jour »*, soit à l'aide de descripteurs comme *« tous les 1er jours de chaque mois »*. Dans ce dernier exemple, *« tous les 1er jours »* est un premier descripteur et *« chaque mois »* en est un second. UniteC et UniteCNommee sont des énumérations qui décrivent les unités calendaires utilisées par les descripteurs i.e., *« tous les jours »*. Dans

[1] CCSL : Clock Constraint Specification Language
[2] http://www.omg.org/omgmarte/
[3] ISO, 2002. Text of 19108 Geographic information - Temporal schema

l'expression *« tous les 1er jours »*, *« 1er »* est un rang. La Figure 1 est une version simplifiée de notre métamodèle (Faucher *et al.*, 2010) présentant les éléments nécessaires à la compréhension de notre approche.

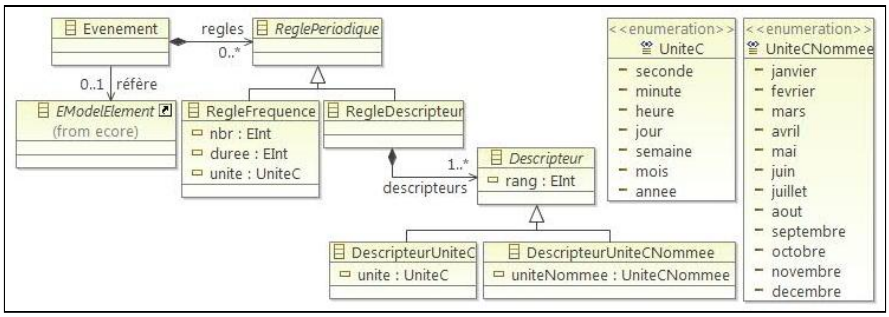

Figure 1. *Extrait du métamodèle temporel*

3. Un modèle de référence pour capturer la sémantique

Afin d'ajouter de la connaissance aux éléments calendaires inclus dans le métamodèle (i.e., concepts d'année, mois, heure, etc.), nous définissons un modèle de calendrier nommé « modèle de référence ». Chaque élément calendaire est décrit comme un événement périodique. Par exemple, l'expression *« une année commence le 1er jour de janvier et se termine le dernier jour de décembre »* (cf Figure 2) utilise des concepts de périodicité. Notre métamodèle permet de définir des événements avec des propriétés temporelles périodiques, et chaque élément calendaire peut être défini comme instances de notre métamodèle. Nous traitons ici le calendrier Grégorien, mais nous pourrions également utiliser la même approche pour définir le calendrier Républicain définit durant la Révolution Française.

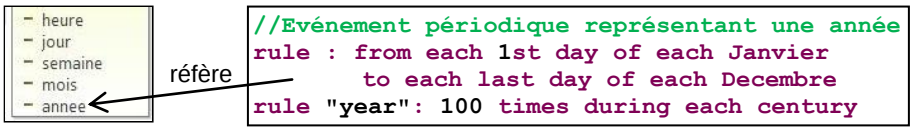

Figure 2. *Relation entre un événement et un élément du métamodèle temporel*

La Figure 2 montre la mise en correspondance de l'événement year représentant un élément calendaire avec le littéral annee de l'énumération UniteC issue du métamodèle. Cette mise en correspondance est réalisée par la référence référe fournie par la classe Evenement. Cette référence désigne la classe EModelElement du métamodèle d'Ecore. Cette architecture permet d'accéder à la définition temporelle

d'un élément du métamodèle et ensuite de vérifier des contraintes. Le modèle de référence est extensible, l'utilisateur peut ajouter de nouvelles règles, ce qui peut renforcer l'évaluation d'une contrainte. Une première version du calendrier pourrait définir la notion d'année bissextile comme ayant lieu « tous les 4 ans », l'expert du domaine pourrait ajouter dans un deuxième temps « sauf tous les 100 ans ».

4. Validation sémantique à l'aide de contraintes et du modèle de référence

Nous considérons la validation sémantique d'un modèle comme le résultat de l'évaluation d'une contrainte. Une contrainte est définie sous forme d'un invariant de classe comme cela pourrait être exprimé en OCL. Les aspects structurels d'un modèle ne sont pas vérifiés par ces contraintes. Nous avons choisi dans un premier temps d'implémenter ces contraintes en Java, notamment du fait qu'il est nécessaire de manipuler simultanément plusieurs modèles à différents niveaux de modélisation (cf Figure 3), ce qui n'est pas aisé en OCL.

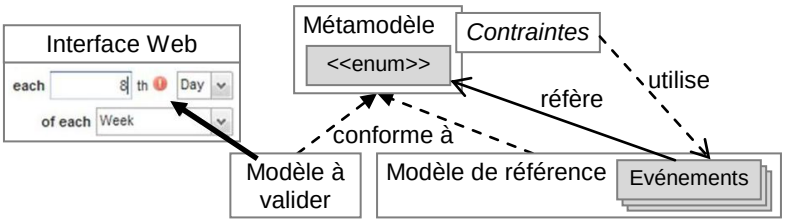

Figure 3. *Ressources mises en œuvre dans le processus de validation*

En effet, les contraintes possèdent des variables et paramètres dont les valeurs sont issues du modèle à vérifier et, également, du modèle de référence. L'Ingénierie Dirigée par les Modèles est au cœur du processus de validation afin de naviguer et de mettre en relation le métamodèle du domaine et les modèles sous-jacents. Prenons comme exemple deux expressions que l'on souhaite valider :

1. les 8èmes jours de chaque semaine (each 8th day of each week)
2. les 8èmes jours de chaque mois (each 8th day of each month)

La 1ère expression est incorrecte car le modèle de référence stipule qu'une semaine possède au plus 7 jours. En revanche, la 2ème expression est correcte car un mois a, au plus, 31 jours. La Figure 4 montre la mise en œuvre de la validation de la 1ère expression. La contrainte possède une fonction `freq` prenant en paramètres les unités de deux descripteurs successifs. Cette fonction retourne la fréquence définie dans le modèle de référence et associée au couple d'unités de descripteurs. Pour le couple (jour, semaine), le résultat est 7. Ainsi le rang de l'expression à valider doit être inférieur ou égal à 7. Dans l'exemple, le rang a pour valeur 8, donc supérieur à 7, et une icône d'erreur sera affichée (cf Figure 3). L'ajout des contraintes pourrait être géré par des annotations comme cela est décrit dans (Noguera et Duchien 2008). La partie « implémentation de la validation » consisterait à l'implémentation de la

fonction freq et à l'évaluation de la contrainte. Dans notre cas, la référence réfère subsisterait car elle fournit une information utile pour d'autres applications.

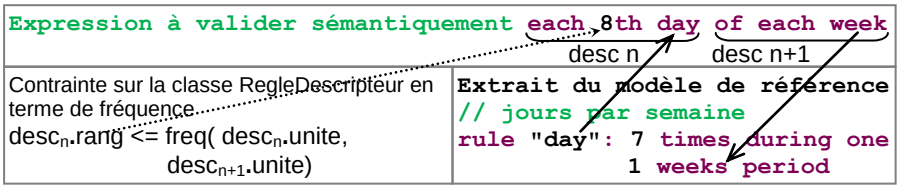

Figure 4. *Exemple de contrainte utilisant le modèle de référence*

5. Conclusions et perspectives

Dans cet article nous montrons de quelle manière la validation sémantique de modèles peut exploiter un modèle de référence, contenant une connaissance métier, en utilisant les techniques de l'Ingénierie Dirigée par les Modèles. Nous prenons comme exemple un métamodèle décrivant des expressions temporelles périodiques. À partir des contraintes définies sur ce métamodèle, la validation est réalisée en évaluant des contraintes utilisant le modèle de référence comme source de connaissance. La validation d'expressions temporelles a été mise en œuvre pour assister et contrôler la saisie d'informations temporelles périodiques décrivant des événements issus de dépêches d'information. Notre approche pourrait également être appliquée au domaine spatial pour valider, par exemple, des expressions décrivant le positionnement d'objets telles que *« chaque 10^{ième} mètre de chaque kilomètre »*.

Remerciements

Ce travail est financé par le projet ANR RelaxMultiMedias 2 (ContInt).

6. Bibliographie

Allen J. F., « Maintaining knowledge about temporal intervals », *Communications of the ACM*, vol. 26, n° 11, p. 832-843, 1983.

Carnap R., *Meaning and Necessity*, University of Chicago Press, 1947.

Faucher C., Teissèdre C., Lafaye J.-Y., Bertrand F., « Temporal Knowledge Acquisition and Modeling », *EKAW 2010*, vol. 6317 of LNCS (LNAI), Springer-Verlag, Lisbon, Portugal, p. 371-380, 2010.

Noguera C., Duchien L., « Annotation Framework Validation using Domain Models », *ECMDA 2008*, vol. 3844 of LNCS, Springer-Verlag, Berlin, Germany, p. 584-600, 2008.