

PROGRAMMATION BASÉE MODÈLE

XMI, un format d'échange pas très standard

Le XMI est un format permettant d'échanger des modèles UML d'un outil à un autre. Bien que standardisé, la réalité de sa mise en œuvre n'est pas au niveau des attentes. Il reste en effet beaucoup de chemin à parcourir pour pouvoir échanger correctement des modèles UML.

A l'heure où l'électronique est de plus en plus présente dans le monde qui nous entoure, le domaine de la modélisation des systèmes temps réel ne cesse d'évoluer. Ainsi, aujourd'hui, avant de concevoir un système, il est intéressant de travailler sur un modèle qui est une représentation abstraite du système. Ce modèle permettra de documenter précisément ledit système, afin entre autres de faciliter la communication entre les équipes, et autorisera un certain nombre de vérifications très tôt dans le processus de développement. UML (Unified modeling language), issu de l'unification de différents langages de modélisation objet, est un des langages qui permet de réaliser cette phase de modélisation. Il a la particularité de s'appuyer sur une approche dite « centrée modèle » (model centric en anglais), c'est-à-dire que les différents diagrammes sont des vues du modèle qui peuvent être partielles, et qu'un même élément du modèle peut être représenté dans plusieurs diagrammes. Dans cette approche, les diagrammes sont donc dérivés du modèle qui est la référence. Cette approche est à opposer à des langages dits « centré diagramme » (diagram

A.- Exemple d'un diagramme de classes comprenant deux classes avec une relation d'association entre elles, sur lequel a été testé le format d'échange XMI.

centric en anglais) comme par exemple le SDL (Specification and description language) où le modèle est dérivé de différents diagrammes.

La représentation des modèles UML est standardisée et s'appuie sur un méta-modèle dénommé le MOF (Meta object facility). Afin d'échanger des modèles d'un outil à un autre, les modèles UML sont exportés et importés au format XMI (XML Metadata interchange). Le XMI est en fait une représentation textuelle du modèle selon le MOF. Or, comme nous l'avons vu, le modèle est la référence en UML et c'est pourquoi, dans les premières versions du XML, il n'y avait aucune information graphique de représen-

```
<uml:Model xmi:type="uml:Model" name=" Model" visibility="public">
  <packagedElement xmi:type="uml:Package" xmi:id="1" name="MyPackage2" visibility="public">
    <packagedElement xmi:type="uml:Class" xmi:id="2" name="Tata" visibility="public"/>
    <packagedElement xmi:type="uml:Class" xmi:id="3" name="Toto" visibility="public"/>
    <packagedElement xmi:type="uml:Association" xmi:id="4" name="memberEnd="2 3"/>
  </packagedElement>
</uml:Model>
```

```
<uml:Model xmi:type="uml:Model" name=" Model" visibility="public">
  <packagedElement xmi:type="uml:Package" xmi:id="1" name="MyPackage2" visibility="public">
    <packagedElement xmi:type="uml:Class" xmi:id="2" name="Tata" visibility="public"/>
    <packagedElement xmi:type="uml:Class" xmi:id="3" name="Toto" visibility="public"/>
    <packagedElement xmi:type="uml:Association" xmi:id="4" visibility="public">
      <memberEnd xmi:idref="2"/>
      <memberEnd xmi:idref="3"/>
    </packagedElement>
  </packagedElement>
</uml:Model>
```

B.- On voit ici deux exemples de fichiers d'export XMI issus de deux outils du commerce pour le diagramme de la figure 1.

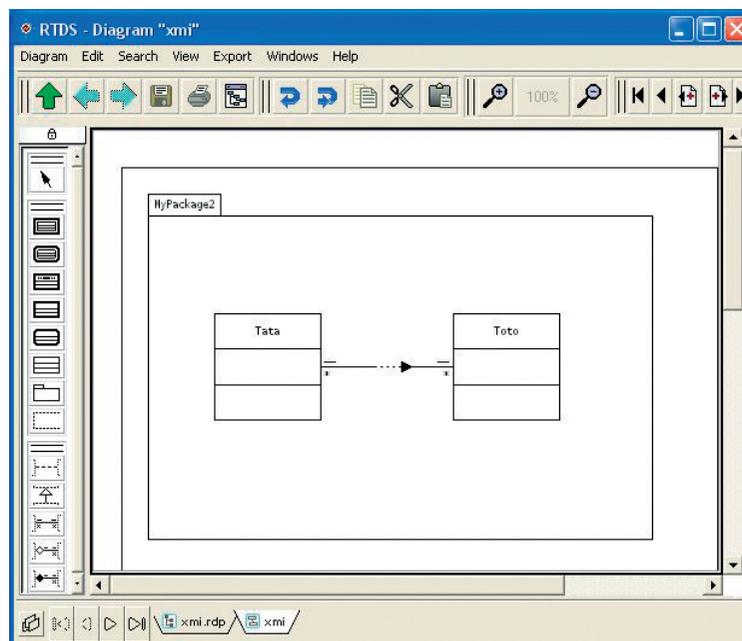
tation des diagrammes.

La version 1 d'UML avait une approche trop générique du système pour permettre une modélisation fine des systèmes quel que soit le domaine d'application. La version 2 permet depuis de définir des profils spécialisant le langage afin de décrire des modèles plus précis et d'offrir des fonctionnalités de vérification, de simulation, ou de génération de code. A ce jour, deux profils temps réel sont standardisés : le profil UML Z.109 basé

sur le SDL pour les systèmes de télécommunications, et le profil Marte pour la modélisation système (*Electronique* n°149, p.41, pour une description complète de ce profil). Associée à cette évolution d'UML, la version 2 du format XMI permet l'import et l'export des profils associés au modèle. Mais à ce niveau, il est important de noter que la possibilité d'échanger un modèle avec son profil associé reste aujourd'hui un vœu pieux, car les outils du marché ne supportent en général qu'un seul profil qui leur est souvent propriétaire.

Un format qui prête à interprétation

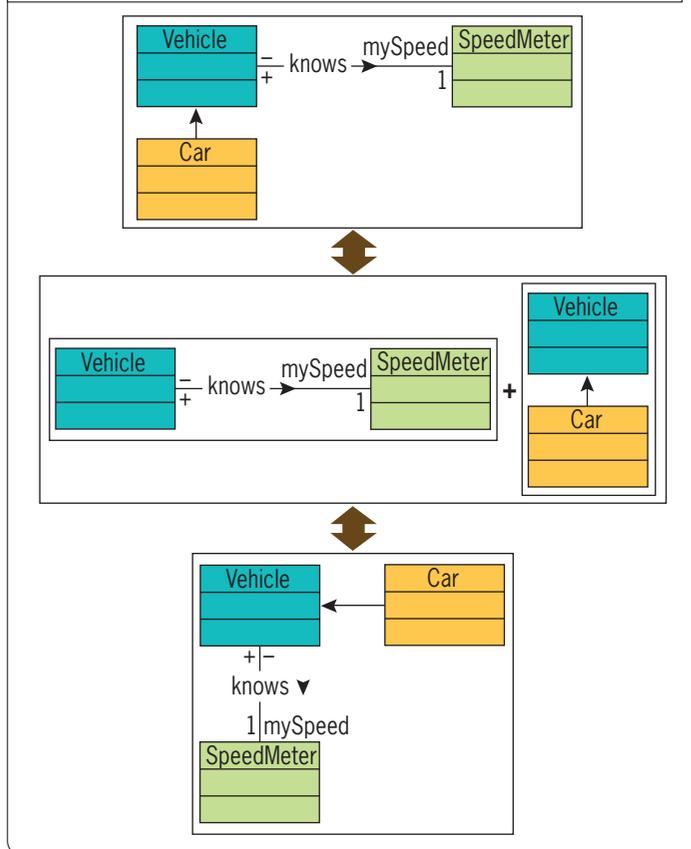
A l'occasion de l'écriture d'un module d'import de modèles UML au format XMI, il nous est apparu que le standard de format d'échange prêtait à interprétation et que les fichiers produits par divers outils, bien que conformes à la norme, étaient incompatibles entre eux. Prenons l'exemple le plus simple possible : un diagramme de classes comprenant deux classes avec une relation d'association entre ces classes (photo A). Pour ce même diagramme venant d'outils commerciaux, nous avons analysé deux exemples de fichier d'export XMI (photo B).



Trois représentations d'un même modèle

FIGURE

Les modèles suivants, bien que représentés différemment, sont identiques et produiront le même fichier XMI.



Le premier tag `uml:Model` indique qu'il s'agit d'un modèle UML. Le deuxième tag, `packageElement` avec l'attribut `Package`, définit le package qui contient les deux classes. Les troisièmes et quatrièmes tags avec l'attribut `Class` décrivent les deux classes à l'intérieur du package. Enfin, le cinquième tag avec l'attribut `Association` décrit l'association entre ces deux classes. Mais pour désigner les deux extrémités de l'association, un outil va utiliser l'attribut `memberEnd` et lister les deux identifiants, alors qu'un autre outil va utiliser le tag `memberEnd` pour chaque extrémité de l'association avec l'attribut `idref` pour identifier les classes.

On voit déjà sur cet exemple extrêmement basique apparaître des différences fondamentales d'organisation du fichier d'export. Inutile de préciser que, dans ce cas précis, il n'est pas possible de porter le modèle d'un outil à l'autre. Comme expliqué précédemment les informations graphiques concernant les diagrammes ne sont pas exportées, seul le modèle l'est. Or, un même

modèle peut être décrit par différents diagrammes et l'organisation en diagramme facilite la compréhension du modèle (figure). L'absence d'information graphique est donc très pénalisante pour l'échange de modèles simples. Inutile en plus d'imaginer échanger des modèles basés sur des profils.

Ainsi, alors que la version 1 d'UML semblait représenter un point de convergence des langages de modélisation, on assiste avec la version 2 à une multiplicité de profils propriétaires souvent non documentés. Très récemment sont même apparus de nouveaux langages de modélisation comme AADL dans l'avionique et le spatial, ou la méthodologie Autosar dans le domaine automobile, qui ne prennent pas leur origine dans l'UML. In fine, pour ce qui est de la portabilité des modèles, les ambiguïtés du format XMI rendent aujourd'hui l'échange en UML basique difficile et quasiment irréaliste avec le moindre profil.

CHARLES CASTELLI (LABORATOIRE COSI, ESIEE-PARIS) ET EMMANUEL GAUDIN (PRAGMADEV)