



Pas d'objets à sens unique !

Jérôme Euzenat
INRIA Rhône-Alpes

LMO a été créé pour rapprocher la représentation de connaissance par objets et la programmation par objets de manière un peu plus articulée que les montres à moutarde.

Nous nous sommes attelés, au début des années 90, à développer différences et rapprochements. L'un des traits distinctifs de la représentation par objets devait être, au moins dans notre esprit, l'existence d'une sémantique précise et "dénotationnelle" à l'instar des logiques. Un des traits caractéristiques, à l'époque, de la programmation par objets était la présence de méta-modèles dans la lignée de SmallTalk-76 et ObjVLisp. Ceci nous faisait entrevoir la difficulté de la tâche : il n'est pas possible de donner une sémantique dénotationnelle simple aux objets de la programmation. Les tentatives les plus convaincantes utilisaient la théorie des ensembles non-bien fondés.

Depuis, le monde à évolué. Les langages C++ et Java sont devenu des standards de fait de la programmation par objets. L'aspect méta-modélisation des langages de programmation est quasi-inexistant (le package reflex de Java, ne permet que de manipuler dynamiquement des classes compilées). Les représentations de connaissance se sont largement "logicisée" et tentent d'envahir le web. Elles adoptent au passage les traits les plus ouverts et la capillarité abandonnant les tentatives d'encapsulation (non, ne cherchez pas la contrepétrie). On observe en parallèle la montée en puissance des langages de modélisation au travers d'UML et maintenant de l'approche MDA. Cette approche qui place en son centre une représentation pour la manipuler de l'extérieur et la transformer est en fait très proche de la représentation de connaissance telle qu'on la concevait à la création de LMO. Par ailleurs, certaines initiatives telles que "strict UML" cherchent à exprimer précisément la sémantique d'UML. Est-ce à dire que le programme de LMO est en train de s'accomplir subrepticement?

En fait, les aspects de modélisation sont remontés dans les couches de spécifications qui sont exécutées dans des langages relativement simples n'ayant pas besoins d'être réflexifs. Mais les langages de modélisation, pour les mêmes arguments pratiques que ceux qui conduisaient à la présence de méta-modèles dans les langages de programmation (utiliser les mêmes outils sur le langage de modélisation que sur le modèle), se sont dotés de méta-modèles réflexifs. Une fois de plus, la sémantique a du mal à faire son nid (ne se trouve-t-elle pas repoussée dans les transformations?) et j'ai bien peur que nous soyons repartis pour un tour...

Patron, la même chose !