

Adaptation d'exécution de modèles

Olivier Le Goer, Eric Cariou, Franck Barbier, and Samson Pierre

Université de Pau / LIUPPA
BP 1155, F-64013 PAU CEDEX, France
{prenom.nom}@univ-pau.fr

1 Introduction

Un des buts fondateurs de l'ingénierie des modèles (IDM) est de considérer les modèles comme les éléments principaux et productifs pour le développement d'applications. On distingue généralement deux approches productives des modèles : la génération de code à partir d'un modèle ou l'interprétation directe du contenu d'un modèle. Selon cette dernière approche, plus récente, on dispose d'un moteur d'exécution implémentant une sémantique opérationnelle capable d'interpréter un modèle. De tels modèles sont écrits dans des langages de modélisation (DSML pour *Domain-Specific Modeling Language*) intégrant la possibilité d'interpréter leur contenu. On parle alors de i-DSML pour *interpreted-DSML*. Dans cet article, nous étudions les i-DSML adaptables, c'est-à-dire des modèles qui sont exécutables et également adaptables pendant leur exécution. Cette adaptation a pour objectif de modifier le contenu du modèle en fonction par exemple d'un changement dans le contexte d'exécution.

Dans la section 2 nous caractérisons les éléments constitutifs d'un i-DSML adaptable. Puis dans la section 3, nous présentons l'intérêt de spécialiser les i-DSML adaptables pour définir des sémantiques d'adaptation qui font sens.

2 Caractérisation des i-DSML adaptables

Dans [2], nous avons caractérisé les éléments constitutifs d'un i-DSML adaptable. À la base, un i-DSML est un méta-modèle qui définit des éléments jouant un rôle statique et d'autres jouant un rôle dynamique. Par exemple, pour une machine à états, les états et les transitions forment le contenu statique et pendant l'exécution, il faudra connaître les états actifs courants qui eux forment la partie dynamique. Une sémantique d'exécution est associée à ce méta-modèle et est implémentée par un moteur interprétant le modèle. Pour une machine à états, la sémantique d'exécution précisera comment suivre les transitions quand un événement est généré.

Un i-DSML adaptable est une extension d'un i-DSML. Dans le méta-modèle est rajoutée une partie d'adaptation qui définit des éléments voués à l'adaptation. Par exemple, pour un processus modélisé qui doit se réaliser en un temps donné, une adaptation possible consiste à supprimer des activités non obligatoires pour gagner du temps. C'est au concepteur du processus de préciser ces activités optionnelles et, à cet effet, le méta-modèle doit permettre de les marquer comme telles. Ensuite, une sémantique d'adaptation est également ajoutée ; elle précisera comment adapter le modèle pendant son exécution. Nous considérons une sémantique d'adaptation comme étant une sémantique d'exécution spécialisée : elle traite des situations anormales ou extra-ordinaires tandis que l'autre s'occupe de l'exécution nominale. En pratique, la sémantique d'exécution ne modifie que les éléments dynamiques pour gérer un pas normal d'exécution tandis que la sémantique d'adaptation peut agir sur tout le contenu du modèle (voire sur les sémantiques d'exécution et d'adaptation elles-mêmes). Par exemple, pour notre processus, des activités, donc des éléments statiques, sont supprimés.

Par conséquent, le moteur d'exécution est étendu pour implémenter en plus la sémantique d'adaptation. Concrètement, en plus des opérations d'exécution, on trouvera des opérations dédiées à l'adaptation et de deux natures : (i) des opérations de vérification pour déterminer si une adaptation est requise et (ii) des opérations réalisant l'adaptation.

3 Spécialisation des i-DSML pour favoriser l'adaptation

Dans [1], nous avons décrit un ensemble de critères qui permettent de caractériser une adaptation :

Fonctionnelle vs non-fonctionnelle (qualité de service) : l'adaptation peut modifier ou ajouter des fonctionnalités ou bien concerne la qualité de service de celles existantes.

Prévue à l'avance vs non-anticipée : les adaptations sont « pré-câblées » dans le système dès le départ ou bien on peut gérer des cas non prévus à l'avance.

Prédictible vs non-déterministe : l'adaptation amène le système dans un état bien défini et stable ou bien laisse place à une certaine incertitude.

Auto-adaptation vs par un tiers : l'adaptation est réalisée par le système lui-même ou bien par une entité extérieure qui agit sur le système.

Générique vs métier : l'adaptation est agnostique ou bien est dédiée à un contenu métier particulier.

Le challenge scientifique majeur est clairement de pouvoir faire de l'auto-adaptation fonctionnelle, non anticipée, prédictible et générique. Dans [2], nous montrons un exemple de cela pour une adaptation d'exécution de machines à états : en cas d'un événement inconnu (non anticipée), la machine à états se modifie (auto-adaptation) pour rajouter des états et transitions pour gérer ce nouvel événement (fonctionnel) et cela sans avoir besoin de savoir quel système est modélisé par la machine à états (générique). Bien sur, cela n'est pas magique et ne fonctionne qu'avec des hypothèses fortes sur la forme des machines à états considérées. Notamment, nous imposons qu'un événement donné mène par principe toujours au même état. Cela permet de rajouter les transitions associées à cet événement. Le corollaire est que cette adaptation générique ne marche pas pour toute machine à états. Néanmoins, par rapport à une adaptation purement métier, elle sera davantage réutilisable.

Partant de ces constats, afin de favoriser les adaptations génériques, nous avons défini le concept de « famille » [3]. Pour un i-DSML adaptable, on pourra avoir besoin de spécialiser et contraindre le méta-modèle pour faire émerger des adaptations qui n'ont de sens que sous certaines hypothèses. Concrètement, on pourra définir des opérations de vérification et de réalisation d'adaptation dans un contexte précis. Une famille organise tout cela en associant ces opérations avec un méta-modèle spécialisé donné. On pourra ensuite également spécialiser des familles entre-elles, créant ainsi des hiérarchies de classes d'adaptation pour un i-DSML donné.

4 Perspectives

Un moteur exécute des opérations d'exécution et d'adaptation. Pour l'heure, c'est au développeur du moteur d'orchestrer « en dur » dans son code ces différentes opérations. Nous travaillons à externaliser de telles orchestrations dans un modèle propre. Ce modèle sera interprété à son tour par un moteur, c'est donc également un i-DSML. Ce moteur d'orchestration pourra être couplé à n'importe quel moteur d'exécution pour prendre en charge l'adaptation d'un i-DSML. En considérant ce i-DSML d'orchestration comme lui-même adaptable, on pourra modifier ses modèles et donc les sémantiques d'adaptation qui s'y trouvent réifiées. On fera alors de l'adaptation d'adaptation d'exécution de modèles, dit autrement, de la méta-adaptation, en réutilisant nos principes des i-DSML adaptables sur des i-DSML adaptables.

Références

1. Franck Barbier, Eric Cariou, Olivier Le Goer, and Samson Pierre. Software adaptation : classification and case study with State Chart XML. *IEEE Software*, (à paraître), 2015.
2. Eric Cariou, Olivier Le Goer, Franck Barbier, and Samson Pierre. Characterization of Adaptable Interpreted-DSML. In *ECMFA 2013*, volume 7949 of *LNCS*, pages 37–53. Springer, 2013.
3. Samson Pierre, Eric Cariou, Olivier Le Goer, and Franck Barbier. A Family-based Framework for i-DSML Adaptation. In *ECMFA 2014*, volume 8569 of *LNCS*, pages 164–179. Springer, 2014.