

MuSE, un outil de spécification de modèles métier basé sur une approche MDA

Pierre Imbert¹*, Marc Petit²†, and David Blin³‡

¹ EIFER – European Institut For Energy Research, Karlsruhe, Germany

`pierre.imbert@eifer.uni-karlsruhe.de`

² EDF R&D, Palaiseau, France

`marc-f.petit@edf.fr`

³ EIFER – European Institut for energy research, Karlsruhe, Germany

`david.blin@eifer.uni-karlsruhe.de`

Résumé

MuSE est un outil de spécification de modèles métier pour la simulation. Il repose sur un langage dédié et permet la génération d'éléments pour l'intégration dans la plateforme CURTIS des codes décrits ainsi que leur implémentation partielle. La genèse de cet outil démontre l'intérêt d'une approche Dirigée par les Modèles en termes de rapidité de démonstration et d'appropriation par un public non spécialiste.

Abstract

MuSE is a specification tool for domain specific simulation models. It relies on a Domain Specific Language and generates code for both the (partial) development of simulation codes and their integration in the CURTIS platform. The genesis of this tool demonstrates the benefits of a Model Driven approach for fast prototyping and understanding by non specialists.

1 Contexte : la plateforme CURTIS

Les acteurs qui construisent la ville sont multiples et généralement dédiés à des domaines spécifiques (transport, énergie, urbanisme, développement économique, déchet, etc.). La ville évolue à la convergence des décisions dans chacun de ces domaines. La prise de décision est délicate car la réalité et les mécanismes qui sous-tendent le système urbain sont difficiles à appréhender. La planification urbaine et *pluri-domaine* bien que de mise, reste un exercice difficile et souvent cloisonné par domaine.

De par sa proximité avec les territoires et la richesse de ses métiers, EDF dispose d'une expertise utile à l'appréhension de certaines parties du système (de systèmes) urbain. En ce sens, EDF R&D a développé un service pour aider les grandes villes (ex. Singapour) à définir leur planification *pluri-domaine*.

Dans ce cadre, un outil d'aide à la décision a été développé – appelé EDF City Platform ou CURTIS, combinant plusieurs fonctionnalités : navigation 3D dans la ville, simulation de scénarios d'évolution de la ville et calculs d'impact, visualisation des résultats et analyse multicritères.

La valeur ajoutée de CURTIS, est l'approche intégrée qui prend en compte le caractère *pluri-domaine* du système urbain dans ses analyses. A cette fin, elle intègre et couple plusieurs modèles de calculs, chacun dédié à un domaine et appelé *domain model*.

*Responsable de lot, product owner

†Initiateur, architecte et implémenteur du premier PoC

‡Responsable de projet

Les *domain models* sont spécifiés en amont de la plateforme CURTIS par des experts du domaine, puis développés par des informaticiens, le tout suivant un processus de modélisation rigoureusement défini. Celui-ci a pour finalité la traduction du *domain model* en une forme exécutable par un ordinateur et valide sur cinq niveaux : (1) réponse aux besoins exprimés par le client (2) validité au regard de l'état de l'art dans le domaine (3) conformité de l'exécutable informatique au modèle (4) capacité de couplage à d'autres *domain models* (5) compatibilité avec la plateforme CURTIS.

L'automatisation (accélération, amélioration de la reproductibilité) du processus de modélisation via une approche dirigée par les modèles est le sujet de MuSE.

2 Problématique

Le processus de modélisation présente plusieurs limitations :

- Les spécifications fonctionnelles et techniques dans leur forme actuelle (document en langage naturel) sont potentiellement ambiguës, notamment pour un développeur informatique qui ne connaît pas le domaine. Le manque de formalisation interdit de plus toute vérification de cohérence ou génération (même partielle) de code.
- Les *domain model* sont conceptualisés indépendamment les uns des autres, et de fait la réalité d'une ville est souvent appréhendée de manière assez différente d'un domaine à un autre. Cela rend fastidieux - voire impossible - le couplage des codes résultants (4ème axe de validation).
- Les travaux de validation (suivant les 5 axes) des modèles sont réalisés en grande partie à la fin de l'étape de programmation. Il en résulte qu'en cas de modèle non valide, la correction nécessite de re-dérouler intégralement le processus.

Notre expérience révèle qu'une amélioration de l'étape de spécification pourrait permettre, en limitant les demandes de correction ou d'information supplémentaire, de limiter les itérations et par là de diviser par deux le temps nécessaire à l'ensemble du processus de modélisation et développement (conclusion conforme à [4]).

Les améliorations envisagées sont :

- Standardiser le contenu et le formalisme d'une spécification fonctionnelle et technique appliquée à un *domain model*. Ce standard doit être applicable à la diversité des domaines et des approches de modélisation (data flow, analyse spatiale, système d'équations, orientée objet, dynamique de système, continue/discrète, multi-agent, etc.). Dans cette perspective, des langages de modélisation unifiés tel que UML, SysML [2] ou Modelica [3] ont été considérés mais se sont montrés à la fois complexes (notamment pour UML) et insuffisants.
- Partager une ontologie de la ville, c'est-à-dire un modèle de l'ensemble des informations qui la concernent, du point de vue des domaines métier considérés.
- Permettre la validation (des niveaux 1 à 5) dès la conception des *domain models*
- Favoriser la collaboration des experts en cours de spécification des *domain models*, ainsi que l'enrichissement de l'ontologie.
- Générer automatiquement des informations issues de la spécification dans un formalisme adapté au cas d'usage (intégration, développement ...).

3 Cas d'utilisation de MuSE

MuSE est un outil informatique dont les utilisateurs principaux sont les experts ayant en charge la spécification fonctionnelle et technique d'un *domain model* en vue de son intégra-

tion dans CURTIS. Plusieurs expertises sont nécessaires pour écrire ces spécifications : celle du domaine concerné, celle de l'approche de modélisation utilisée, celle de développement informatique dans le langage exécutable défini, celle du couplage des *domain models* et celle de la connaissance des besoins du client. Pour l'ensemble de ces utilisateurs, on identifie pour MuSE les cas d'utilisation suivants :

- **UC1** Créer une spécification en utilisant des vues par type d'information (entrées, sorties, validation, traitements, ordonnancement, etc.), avec un formalisme et une visualisation adaptée;
- **UC2** Valider automatiquement la cohérence et la pertinence des informations contenues dans la spécification, en cours d'écriture;
- **UC3** Consulter et extraire des concepts depuis une ontologie du domaine considéré;
- **UC4** Tester la validité des spécifications à partir d'une version exécutable simplifiée;
- **UC5** Identifier les couplages possibles entre les *domain models*;
- **UC6** Générer automatiquement depuis les informations contenues dans les spécifications les documentations nécessaires aux étapes de programmation et intégration;
- **UC7** Capitaliser les spécifications des *domain models* pour permettre la recherche et la comparaison de *domain models*;
- **UC8** Vérifier en cours d'écriture des spécifications la disponibilité de données à utiliser dans le cadre d'une étude particulière.

4 Environnement de MuSE

MuSE a pour but de permettre la spécification et l'intégration de codes de calcul dans une plateforme informatique (CURTIS). Il est par conséquent nécessaire d'anticiper les interfaces, notamment vis-à-vis de deux composants particuliers : l'ontologie des villes (UC3) et le langage de description des codes de calcul (UC4) appelé Flexible Pivot Language ou FPL.

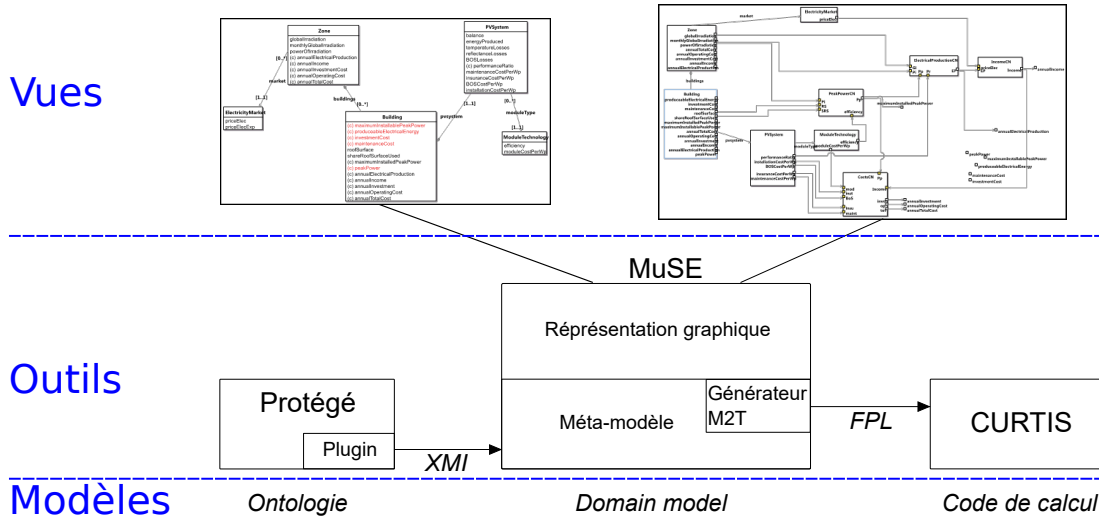


FIGURE 1 – Architecture de Muse

Pour des raisons d'optimisation ou de restriction de domaine, chaque code de calcul a une vision différente de la ville qu'il est illusoire de vouloir réconcilier intégralement. Il reste cepen-

dant possible de concevoir une description commune utile aux échanges de données au sein du système : c'est le rôle de l'ontologie [1] (une description de ce qu'est la ville, pas de la façon dont elle est vue dans un code ou un autre). Utilisée comme base de concepts pour la spécification des modèles de simulation elle facilite à la fois l'intégration et le couplage des codes. Elle ne doit cependant rester qu'un guide, chaque expert étant en mesure de pratiquer les adaptations nécessaires à un modèle tout en sachant que celles-ci présentent un coût vis-à-vis de la plateforme.

Afin de découpler la vision métier des modèles de simulation des contraintes techniques d'intégration (et notamment des moteurs d'exécution), un langage intermédiaire est utilisé : le Flexible Pivot Language ou FPL. C'est un ensemble de documentations et de conventions permettant une description simple et efficace des domain model, sous une forme exécutable de haut niveau. Le domain model édité dans MuSE est indépendant de la plateforme là où le FPL intègre des aspects spécifiques à CURTIS.

5 MuSE

Compte-tenu du besoin de créer un langage de modélisation spécifique au domaine de la simulation ainsi que de celui de transformer les modèles résultants (génération de FPL notamment), une approche de type Model-Driven Architecture (MDA) mais basée sur un Domain Specific Language (DSL) a été retenue. Cette solution offre à la fois la richesse désirée et la capacité à créer un outil taillé pour des utilisateurs non spécialistes de langages comme UML.

La disponibilité des futurs utilisateurs étant limitée et au vu de la complexité de l'approche retenue, un processus itératif a été retenu pour le développement de MuSE.

Une preuve de concept (PoC) a été élaborée en interne sur la base de l'environnement Eclipse. Une ontologie de la ville étant développée par ailleurs dans l'outil Protégé, un composant d'export a été ajouté à ce dernier pour créer le lien désiré entre ontologie et outil de modélisation. Cette PoC a permis de démontrer à peu de frais les principes proposés et de valider tant l'utilité que la faisabilité de l'approche.

Au-delà de son rôle initial, la preuve de concept a permis d'impliquer dans cette démarche des interlocuteurs au départ novices dans l'approche MDA. L'appropriation rapide des concepts et de la logique sous-jacente par de nouveaux intervenants a conduit à une proposition d'interface construite avec les utilisateurs qui, si elle ne reprend pas la terminologie MDA classique, tire parti des capacités offertes par l'Ingénierie Dirigée par les Modèles. Les questions posées au cours de ce processus ont d'ailleurs conduit à élargir considérablement le périmètre de MuSE qui, de simple outil d'illustration formalisée de spécification tend à devenir aujourd'hui un environnement de conception à part entière.

La chaîne de création d'un code de simulation, du modèle physique à l'implémentation, est un processus complexe. Dans le cadre de la plateforme CURTIS, il nécessite de passer par différents outils ainsi que les langages correspondants : MuSE, FPL, moteur de simulation, scripts d'intégration. L'utilité réelle d'un outil comme MuSE dépend donc de l'expressivité de son langage ainsi que de la capacité à transformer les informations qui y ont été saisies en éléments concrètement intégrables dans la plate-forme. Une expérimentation a ainsi été menée sur un cas concret (un modèle d'évaluation de potentiel photo-voltaïque), permettant à la fois de valider les travaux réalisés et d'anticiper les adaptations nécessaires à la création d'une chaîne de transformation complète.

Afin de prendre sa place vis-à-vis de la plateforme CURTIS (elle-même utilisée dans un contexte commercial), MuSE doit devenir un outil industriel. Au-delà de la démonstration initiale il est donc nécessaire d'assurer la transférabilité des développements et de la maintenance

hors du cadre R&D. Cet objectif a été rempli lors d’une première phase d’expérimentation. Celle-ci a consisté à sous-traiter la mise en œuvre de plusieurs fonctionnalités, validant ainsi la qualité de l’expression des besoins. L’étape suivante de développement d’une version initiale pleinement utilisable est en phase initiale (lancement d’une consultation) au moment de la rédaction de cet article.

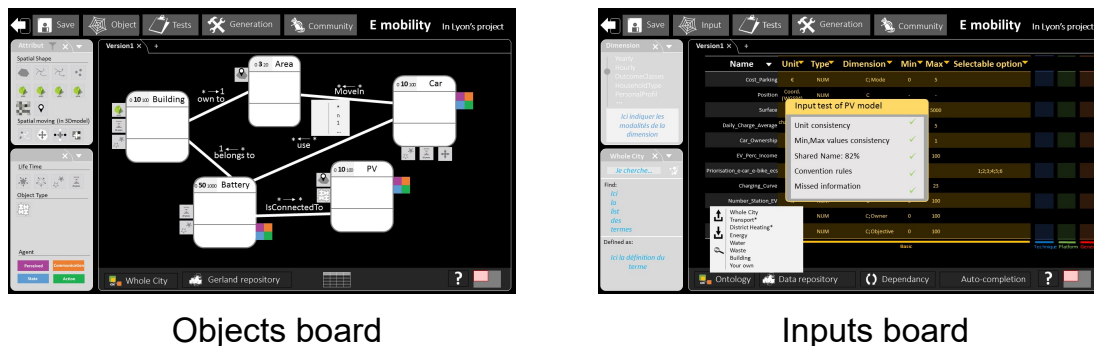


FIGURE 2 – Concepts de boards (vues) pour Muse v1.0

6 Conclusion et Perspectives

Malgré la complexité des approches dirigées par les modèles (niveaux d’abstraction, langages formels, transformations), les outils sont aujourd’hui disponibles pour suivre un processus itératif partant d’une preuve de concept développée rapidement jusqu’à une implémentation à part entière. Cette approche permet une implication rapide des futurs utilisateurs et au final l’adoption par toutes les parties prenantes.

Le système définitif n’étant pas encore réalisé, il reste la principale perspective d’évolution de ce travail. Les verrous identifiés (faisabilité technologique et appropriation par les experts métier) ayant été levés, nous sommes aujourd’hui en mesure de valider un premier résultat. Les voies d’amélioration restant à explorer concernent essentiellement le travail collaboratif et l’enrichissement de l’ontologie en rétro-action à l’élaboration d’un modèle.

Références

- [1] Henri Boccon-Gibod. Application de méthodes et outils du web sémantique pour la gouvernance d’un système d’information industriel. *Semaine de la Connaissance*, 2006. <http://democritique.org/CV/>.
- [2] Sanford Friedenthal, Alan Moore, and Rick Steiner. *A practical guide to SysML : the systems modeling language*. Morgan Kaufmann, 2014.
- [3] Peter Fritzson. *Principles of object-oriented modeling and simulation with Modelica 2.1*. John Wiley & Sons, 2010.
- [4] Jones. Software quality in 2013 : A survey of the state of the art, 2013. <http://namcookanalytics.com/wp-content/uploads/2013/10/SQA2013Long.pdf>.