

Modane : un outil de conception pour les codes de simulation numérique

■ B. LELANDAIS - M.-P. OUDOT / CEA – DAM Île-de-France

L'augmentation continue de la puissance des supercalculateurs permet la création de codes de simulation numérique de plus en plus aboutis. Par conséquent, les physiciens et mathématiciens en charge du développement sont confrontés à des algorithmes complexes qu'ils doivent intégrer dans de gros simulateurs en utilisant des technologies informatiques pointues. Le CEA – DAM étudie depuis plusieurs années l'apport des technologies d'ingénierie des modèles au cycle de développement de ses simulateurs. Ce travail a notamment donné naissance à l'application Modane [1], présentée dans cet article.

Le processus classique de développement d'un code de simulation numérique consiste à écrire des documents de spécification, généralement au format Latex, et à développer, à partir de ces documents, du code source en C++. Les ingénieurs en charge de ce développement ont, pour la plupart, une formation en mathématiques appliquées. Ils appréhendent de manière naturelle la phase de spécification mais, lorsque la phase de codage débute, ils sont confrontés à une quantité importante de concepts et de technologies informatiques: le paradigme orienté objet, le langage C++, les différents modes de parallélisme, etc. Cette complexité peut entraîner beaucoup d'erreurs de conception: manque d'interface, taille trop importante des classes, couplages trop nombreux, utilisation abusive de l'héritage. De plus, les choix d'architecture sont très dépendants de l'expérience de l'ingénieur et sont rarement documentés. Avec le temps et l'évolution du logiciel, le code source devient la seule véritable base de connaissances dans laquelle il est complexe de retrouver les concepts métier importants tels que les modules physico-numériques, les variables ou encore les principales opérations.

Afin de remédier à ces difficultés, le logiciel Modane a été développé. Il permet aux ingénieurs de développement de décrire graphiquement la partie statique des codes de simulation, c'est-à-dire l'ensemble des composants comme les variables, structures ou modules, et des relations entre ces composants, comme les héritages ou les associations. Modane est composé d'une interface graphique pour capturer la connaissance métier, d'un ensemble de règles pour valider les éléments saisis et de générateurs de code C++. La génération de code garantit que l'architecture du logiciel est en accord avec la connaissance saisie dans l'interface graphique. Le code généré est dédié à la plateforme de simulation

numérique Arcane, conjointement développée par le CEA – DAM et IFP Énergies nouvelles [2].

L'utilisation de Modane se déroule en trois phases. La première s'effectue dans l'interface graphique (figure 1). Une palette permet de créer des diagrammes et des composants représentant les concepts métier clefs identifiés lors des spécifications. Cette palette offre également des outils pour représenter les relations d'héritage et d'association entre ces composants. Il reste alors à renseigner les propriétés de l'ensemble des éléments au sein de fenêtres graphiques dédiées. Toute cette connaissance est stockée dans un modèle arborescent visible dans un explorateur. D'un point de vue technique, cette interface graphique est réalisée par personnalisation du modèleur Magicdraw avec un profil UML et du code Java.

La deuxième étape consiste à déclencher sur le modèle un ensemble de règles de vérification: contraintes de nommage, compatibilité de types, choix de conception, etc. Des alertes classées par niveau (information, avertissement et erreur) sont levées. Les messages d'erreur sont affichés dans l'interface et les éléments en erreur comportent une pastille rouge, dans les diagrammes et dans l'explorateur de modèle, ce qui facilite les corrections.

Enfin, le modèle validé est exporté pour pouvoir être exploité dans Eclipse, l'environnement de développement C++ [3]. La génération de code est déclenchée par un menu contextuel accessible sur le fichier de modèle et elle produit toute la structure du code source C++ pour Arcane de l'application: l'ensemble des descripteurs de composants (fichiers XML), la totalité du code source des classes de base de l'application ainsi que le squelette des classes réservées à l'ingénieur de développement. L'ensemble des classes est compilable par Eclipse et le développement du corps des opérations peut immédiatement

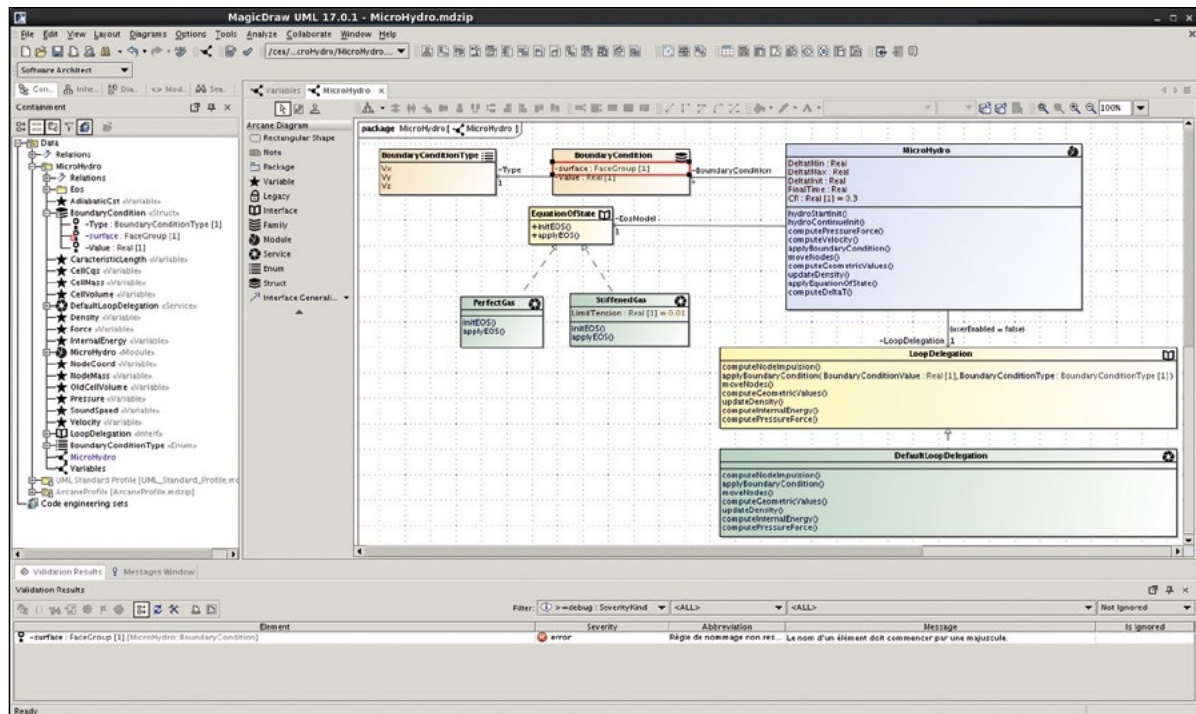


Figure 1. Fenêtre principale de l'interface graphique du logiciel Modane.

commencer. Précisons que le parallélisme n'est pas géré par Modane, qui ne s'intéresse qu'à la partie statique de l'application, mais par Arcane, qui prend en charge le parallélisme MPI et offre des outils et services pour le parallélisme de tâches et la vectorisation.

Développée depuis septembre 2011, l'application Modane est opérationnelle depuis avril 2012. Elle est maintenant utilisée par une dizaine d'ingénieurs de développement sur deux projets du CEA - DAM. Ces ingénieurs reconnaissent un réel gain d'efficacité; ils se concentrent sur les concepts métier de l'application durant la phase de conception et sur la partie fonctionnelle des algorithmes durant la phase de développement. Les changements d'architecture (*code refactoring*) sont également grandement facilités, atout important lorsqu'une méthode de développement itérative est utilisée. On remarque également que l'équipe Arcane peut faire évoluer facilement les bibliothèques dont les appels se trouvent dans le code généré. Il suffit alors de modifier les générateurs pour prendre en compte les modifications. Après quatre années d'utilisation, une première analyse du code produit par les équipes de développement relève moins d'erreurs de conception que sur les projets précédents. Du fait des générateurs, le code source est parfaitement homogène. Le niveau de qualité s'est donc amélioré. Néanmoins, le principal apport de Modane reste son mode de développement collaboratif: les ingénieurs façonnent ensemble l'architecture de l'application autour des concepts métier importants, ils échangent autour

des diagrammes, les utilisent comme support pour communiquer et les nouveaux collaborateurs appréhendent plus aisément les codes de simulation grâce à leur représentation graphique. À ce titre, Modane capitalise une partie de la connaissance.

Nous continuons actuellement à peaufiner le modèle et faisons évoluer les générateurs pour utiliser les fonctionnalités des nouvelles normes C++. Afin d'éviter la phase d'export du modèle, nous travaillons également à une version totalement intégrée à Eclipse et étudions l'apport que nous offrirait le langage Nabla [4] sur certaines parties.

Références

- [1] B. LELANDIS, M.-P. OUDOT, "Modane: A Desing Support Tool for Numerical Simulation Codes", *Oil Gas Sci. Technol. – Rev. IFP Énergies nouvelles*, 71, p. 57 (2016).
- [2] G. GROPELLIER, B. LELANDIS, "The Arcane development framework", *Proceedings of the 8th workshop on Parallel/High-Performance Object-Oriented Scientific Computing (POOSC'09)*, Genova, Italy, July 6-10, p. 1-11 (2009).
- [3] D. STEINBERG, F. BUDINSKY, M. PATERNOSTRO, E. MERKS, *EMF Eclipse Modeling Framework*, Addison Wesley, 2008.
- [4] J.-S. CAMIER, "Improving Performance Portability and Exascale Software Productivity with the ∇ Numerical Programming Language", *Proc. of the 3rd International Conference on Exascale Applications and Software (EASC '15)*, Edinburgh, United Kingdom, April 21-23, 2015, p. 126-131 (2015).