

Opérationnalisation de connaissances pour le calage de modèles numériques – Application en hydraulique fluviale

Jean-Philippe Vidal¹, Sabine Moisan²

¹ Unité de Recherche Hydrologie-Hydraulique, CEMAGREF
3 bis, quai Chauveau – CP 220 – 69336 Lyon Cedex 09
`vidal@lyon.cemagref.fr`

² Projet ORION, INRIA Sophia-Antipolis
2004, route des lucioles – BP 93 – 06902 Sophia Antipolis
`sabine.moisan@inria.fr`

Résumé

Cet article présente une expérience d'ingénierie des connaissances réalisée sur une utilisation avancée des codes de calcul : le calage de modèles numériques. Les connaissances associées à cette tâche ont été modélisées grâce à un langage de représentation de connaissances développé pour une tâche voisine : le pilotage de programmes. Après une présentation de la démarche adoptée, les résultats de notre modélisation sont organisés selon trois niveaux de généricité différente. La tâche générique de calage a été spécialisée dans le domaine de l'hydraulique fluviale, puis une application avec un code de calcul spécifique a été implémentée de manière opérationnelle. Enfin, l'accent est mis sur les leçons tirées de cette expérience.

Mots-clés : Système à base de connaissances, Calage de modèles numériques, Applications, Hydraulique fluviale

1 Introduction

La modélisation numérique est devenue au fil du temps le quotidien de nombreux scientifiques, notamment dans le domaine des sciences environnementales. Les codes de calcul permettent ainsi de simuler le fonctionnement de systèmes physiques par résolution numérique des équations régissant leur comportement. A l'origine cantonnés aux laboratoires de recherche, ces programmes informatiques ont évolué¹ pour se transformer en outils opérationnels à la fois performants et conviviaux. La place grandissante de ces outils numériques au sein de la communauté scientifique a provoqué l'émergence d'un savoir lié aux codes

1. L'exemple du domaine de l'hydraulique est donné dans (Abbott *et al.*, 2001).

eux-mêmes (Picard *et al.*, 1999) mais aussi d'un savoir-faire concernant leur utilisation, souvent difficile à capitaliser (Moisan & Ermine, 2000).

Les connaissances mises en œuvre durant les utilisations avancées des codes de calcul constituent ainsi une expertise qui est actuellement difficilement transmissible. L'intégration de ces connaissances et des codes de calculs pourrait ainsi répondre aux besoins des utilisateurs en terme d'assistance durant des tâches complexes. Parmi ces tâches, le calage de modèle² constitue une étape essentielle du processus de modélisation numérique, représenté sur la Figure 1.

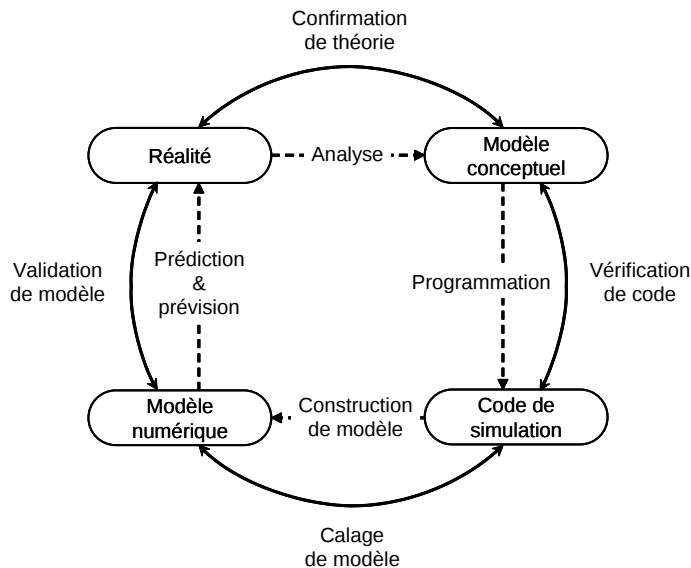


FIG. 1 – Référentiel de modélisation numérique, d'après (Refsgaard & Henriksen, 2002).

Ce mode de représentation, initié par un comité technique interdisciplinaire sur la crédibilité des modèles (Schlesinger *et al.*, 1979), a été repris de nombreuses fois par Sargent (Sargent, 2001) et étendu récemment par (Refsgaard & Henriksen, 2002). La Figure 1 met ainsi en évidence la distinction entre *code de calcul* – le programme informatique qui implémente les équations régissant le comportement générique du système – et *modèle numérique* – la représentation numérique d'un système physique donné. La tâche de calage est définie dans ce référentiel comme la procédure qui permet de vérifier que la mise au point du modèle numérique à partir du code de simulation est correctement réalisée. Le

2. L'ambiguïté du terme *modèle* et de son champ lexical associé (Nouvel, 2002) nous a conduit à employer – autant que faire se peut – le terme de *modélisation numérique* pour faire référence au sens donné dans le domaine de la simulation numérique, et le terme *modélisation des connaissances* pour faire référence à celui donné en ingénierie des connaissances.

calage s'inscrit ainsi comme une partie de la *validation opérationnelle* (Knepell & Arangno, 1993) visant à faire en sorte que les résultats issus du modèle correspondent aux données disponibles sur le système modélisé. Cette tâche apparaît donc essentielle dans le processus de modélisation numérique.

L'objectif final de cette étude est de mettre à disposition les connaissances expertes au travers d'un système d'assistance au calage de modèle numériques, et en particulier de modèles hydrauliques. Pour ce faire, nous nous sommes engagés dans un processus d'ingénierie des connaissances en vue de modéliser l'expertise nécessaire pour mener à bien cette tâche.

Cet article présente et commente la démarche adoptée ainsi que les résultats obtenus. La première partie introduit la méthode employée pour représenter les connaissances et les implémenter dans un système opérationnel. La modélisation des connaissances, qui fait l'objet de la partie suivante, a été effectuée selon trois niveaux : le premier niveau correspond aux connaissances génériques, c'est-à-dire indépendantes du domaine d'application ; au deuxième niveau sont considérées les connaissances propres au domaine d'application considéré, ici l'hydraulique fluviale ; le troisième niveau est quant à lui constitué des connaissances spécifiques à un code de calcul donné. Enfin, la dernière partie de l'article est consacrée aux leçons tirées de cette expérience.

2 Démarche adoptée

Notre modélisation des connaissances s'est organisée en deux temps. Elle nous a conduit à donner une représentation de trois types de connaissances, suivant la typologie de (Chau *et al.*, 2002) :

- *Les connaissances descriptives* font référence aux objets – concrets et abstraits – manipulés pendant la tâche de calage ;
- *Les connaissances procédurales* concernent la structure de la tâche elle-même en terme de définition et d'enchaînement des sous-tâches ;
- *Les connaissances de raisonnement* représentent quant à elles les heuristiques utilisées pour la réalisation effective de ces sous-tâches.

Une étude de faisabilité réalisée à l'aide du formalisme graphique UML (Muller & Gaertner, 2000) nous a permis de fournir une première représentation de la tâche de calage et de jeter les bases d'un prototype opérationnel. Dans un deuxième temps, ce prototype a été réalisé à l'aide d'outils spécialisés.

2.1 Première approche

Pour les besoins d'analyse du prototype, nous avons représenté les connaissances descriptives et procédurales impliquées dans la tâche de calage à l'aide du formalisme UML. Les diagrammes de classes ont été employés pour représenter les connaissances descriptives, tandis que les diagrammes d'activités se sont révélés tout à fait adaptés à la représentation des connaissances procédurales.

Cette option de représentation pour ce deuxième type de connaissances a été préférée à celle des diagrammes SADT préconisée par (Bruaux *et al.*, 2003), par souci d'homogénéité et de compatibilité avec la formalisation des connaissances descriptives. Ces diagrammes permettent en outre de représenter aisément une décomposition en sous-activités, ainsi que les flux d'objets – et leur état – manipulés par les activités. Les connaissances de raisonnement seront quant à elles représentées par des critères fournis par les experts de chaque domaine considéré.

2.2 Utilisation d'outils spécialisés

Sur les bases définies au cours de cette première approche, un prototype a été construit à partir d'un langage textuel de représentation de connaissances et d'un moteur d'inférence associé. Aucun langage ni aucun moteur dédiés au calage de modèle numériques n'étant disponible, nous avons décidé de réaliser ce prototype à l'aide d'outils existants développés pour une tâche proche.

Le langage nommé YAKL a été développé pour la représentation de connaissances sur le pilotage de programmes (Moisan, 2003). Le pilotage de programmes vise à l'utilisation optimale d'un ensemble de programmes préexistants en automatisant la chaîne de traitement: choix, paramétrage et enchaînement des programmes. La tâche de pilotage s'apparente donc à la tâche de calage, dans le sens où elles correspondent toutes deux à une utilisation avancée des codes de calcul.

Notre expérience sur le langage YAKL (Moisan, 2002) a prouvé que sa syntaxe permet de représenter à la fois des classes et des activités, mais aussi des règles définissant l'exécution de chaque activité. Ce langage permet ainsi de modéliser aisément les heuristiques constituant les connaissances de raisonnement impliquées dans chaque tâche ou sous-tâche considérée. De plus, ce langage est doté d'un analyseur qui fournit une traduction en structures de données informatiques utilisables par un moteur d'inférence. Ces outils vont ainsi permettre d'obtenir un prototype opérationnel de système à base de connaissances.

3 Proposition d'une modélisation de la tâche de calage

Notre modélisation des connaissances s'est appuyée d'une part sur de rares articles provenant de différents domaines liés à l'environnement et illustrant des tentatives de formalisation du processus de calage, et d'autre part sur de nombreux entretiens avec des concepteurs et des utilisateurs – expérimentés ou non – de codes d'hydraulique et d'hydrologie. Cette modélisation s'est organisée autour de trois niveaux de connaissances qui font l'objet des sections suivantes.

3.1 Modélisation du calage comme tâche générique

La modélisation des connaissances descriptives du niveau correspondant à la tâche générique sont représentées sur le diagramme de classes de la Figure 2.

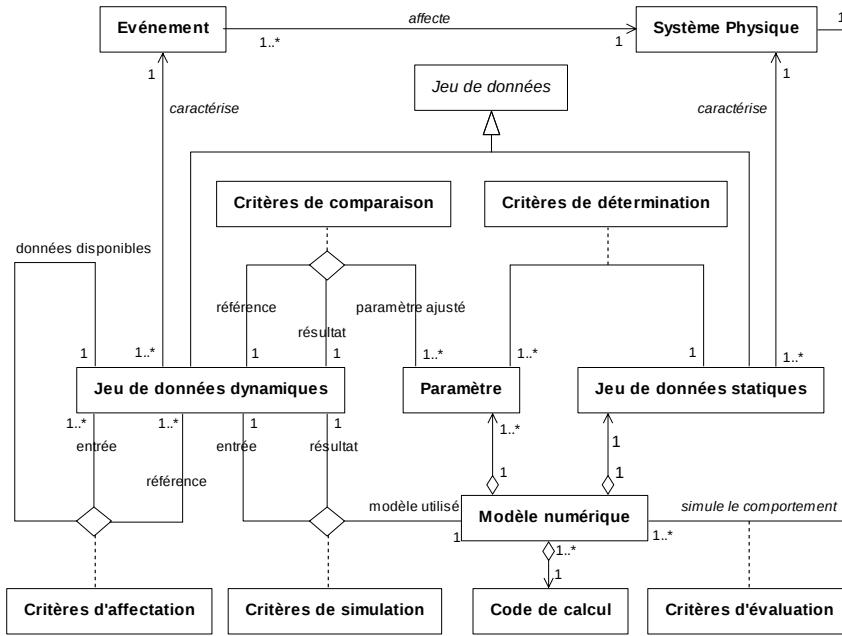


FIG. 2 – Modélisation des connaissances génériques pour le calage – Diagramme de classes UML.

Au sein de ce diagramme cohabitent des entités du monde réel et des entités du monde numérique. Le *système physique* modélisé et les *événements* qui l'affectent sont ainsi caractérisés dans le monde numérique respectivement par un *jeu de données statiques* et un *jeu de données dynamiques*. Un *modèle numérique* a été représenté comme une agrégation du jeu de données statiques caractérisant le système, d'un *code de simulation*, et de *paramètres*. À ces entités s'ajoutent des critères modélisant les connaissances de raisonnement s'appliquant sur chacune des sous-tâches du calage.

Les différentes sous-tâches ont été représentées sur le diagramme d'activités de la Figure 3, sur lequel on retrouve des instances anonymes des critères de raisonnement. La tâche de calage vise à obtenir un modèle numérique *calé*, à partir du modèle *non calé* considéré et d'un jeu de données dynamiques, et au travers des sous-tâches suivantes :

- L'*affectation des données* produit des *entrées* – à fournir au code de calcul pour lancer une simulation – et des *références* à partir des données disponibles.
- La *détermination des paramètres* s'intéresse tout d'abord à choisir les paramètres du modèle qui vont permettre de caler ce dernier, puis à leur attribuer une valeur.

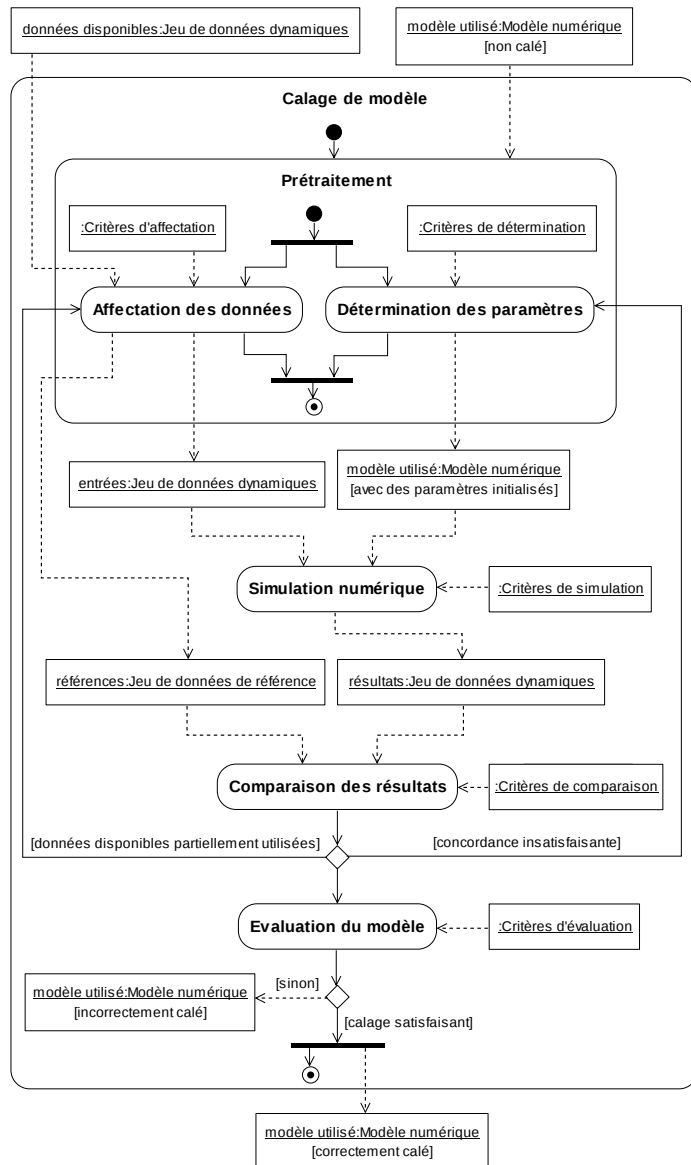


FIG. 3 – Modélisation de la tâche de calage – Diagramme d'activités UML. La sous-tâche de Prétraitement définit deux tâches exécutées en parallèle : Affectation des données et Détermination des paramètres.

- La *simulation numérique* utilise les entrées sélectionnées et le modèle dont on a déterminé les paramètres pour produire des *résultats* grâce au code de simulation.
- La *comparaison des résultats* aux références permet ensuite de lancer un processus de réparation si la concordance n'est pas satisfaisante. Cette réparation peut s'incarner soit dans une modification des valeurs des paramètres, soit dans une redéfinition de ceux-ci. L'étape de comparaison est effectuée pour tous les couples entrées-références que l'on peut obtenir à partir des données disponibles.
- L'*évaluation du modèle* vise enfin à attribuer au modèle numérique calé une estimation de sa capacité à reproduire le comportement du système modélisé.

3.2 Spécialisation pour l'hydraulique fluviale 1-D

Cette étude a été à l'origine motivée par le constat des difficultés rencontrées par les praticiens des codes d'hydraulique pour caler leurs modèles (Cunge, 2003). Nous avons donc transposé les connaissances génériques développées plus haut dans le domaine de l'hydraulique fluviale monodimensionnelle.

L'hydraulique fluviale 1-D s'attache à simuler les écoulements en rivière de plaine à l'aide des équations unidimensionnelles de Saint-Venant. Le système physique considéré est un tronçon de rivière affecté par des événements de crue, et son comportement est simulé par un modèle hydraulique basé sur un code de calcul 1-D. Les paramètres considérés dans le calage sont des coefficients à base physique – mais non mesurables directement – rendant compte notamment de la rugosité du lit de la rivière (cf. Figure 4).

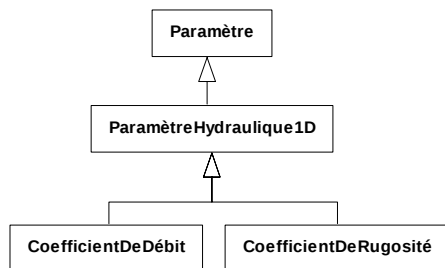


FIG. 4 – Représentation sous forme d'un diagramme de classes UML de la spécialisation de la classe générique *Paramètre* pour l'hydraulique 1-D.

De plus, en hydraulique fluviale 1-D, les données manipulées sont habituellement figurées par des objets graphiques en deux dimensions, c'est-à-dire des courbes ou des points. Ce domaine d'application a donc été décrit sur la base

d'une classification de ces objets graphiques (Vidal *et al.*, 2003). Ainsi, un *hydrogramme* est une courbe – formée de débits en fonction du temps – mesurée dans une section de la rivière et liée à une crue.

Les connaissances procédurales mises en œuvre en hydraulique 1-D ont été modélisées en spécialisant les sous-tâches génériques. Les connaissances de raisonnement, représentées par des critères associés à ces sous-tâches spécialisées, ont été extraites de notre expérience et d'entretiens avec des experts du domaine.

Ainsi, lors d'un calage type de modèle hydraulique 1-D, l'expert recense les données mesurées pendant un événement de crue pour construire les conditions aux limites du tronçon de rivière considéré et constituer un jeu de données de références. En parallèle, à partir des observations de terrain, il détermine la nature et le nombre de paramètres à prendre en compte et leur attribue des valeurs par défaut. La propagation de la crue dans la rivière est alors simulée sur ces bases. Les hauteurs d'eau calculées le long du tronçon – réunie sous le terme *ligne d'eau* – sont ensuite comparées visuellement aux niveaux d'eau réels observés. Si la concordance n'est pas suffisante, l'expert va ajuster les paramètres du modèle avant de relancer une nouvelle simulation. Enfin, l'évaluation du modèle obtenu se base sur sa capacité à reproduire d'autres événements de crue.

3.3 Mise en œuvre avec un code spécifique

Afin de mettre en œuvre de manière opérationnelle les connaissances modélisées dans les deux niveaux précédents, un prototype a été réalisé avec un code d'hydraulique 1-D développé au CEMAGREF et utilisé pour simuler le comportement hydraulique de réseaux fluviaux variés (Giraud *et al.*, 1997). Les connaissances spécifiques à l'utilisation de ce code appelé MAGE constituent ainsi le troisième niveau de notre modélisation, pour lequel nous avons pleinement bénéficié des apports du moteur de pilotage de programmes et du langage associé.

Nous avons ainsi aisément encapsulé les informations contenues dans les fichiers utilisés et produits par le code MAGE. La Figure 5 présente un exemple d'articulation des connaissances spécifiques au code utilisé avec les connaissances de l'hydraulique 1-D. Le code lui-même et les connaissances liées à son utilisation ont aussi été représentés (cf. Figure 6) : script d'exécution, fichiers d'entrée et de sortie, gestion des paramètres de simulation, tests d'erreur et règles de réparation.

Le prototype de système à base de connaissances a été testé pour le calage d'un modèle de la rivière Hogneau, située dans le nord de la France. Ce modèle a été calé de manière à reproduire la crue cinquantennale survenue mi-février 2002, et ainsi à dimensionner les ouvrages nécessaires pour prévenir les inondations relatives à ce type de crue.

4 Discussion

La démarche suivie, depuis la modélisation de la tâche générique de calage jusqu'au test du prototype sur un cas réel, a permis d'une part de juger de

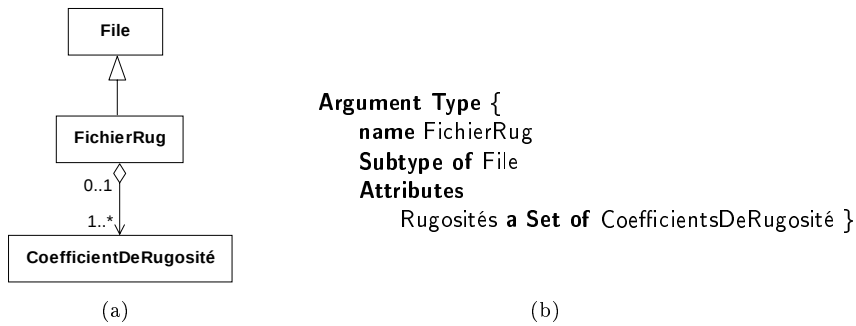


FIG. 5 – Représentation du fichier *.rug utilisé par MAGE et décrivant les coefficients de rugosité, dans le formalisme UML (a) et en langage YAKL (b). La classe File est prédéfinie dans le langage YAKL.

```

Code {
  name Mage
  Input Data
    FichierHyd name FHyd
      comment "Fichier décrivant l'hydrogramme d'entrée"
    FichierRug name FRug
      comment "Fichier décrivant les paramètres de rugosité"
    ...
  Output Data
    FichierBin name FBin
      comment "Fichier binaire des résultats"
    FichierErr name FErr
      comment "Fichier de listing d'erreurs"
    ...
  Assessment criteria
    Rule { name DétectionDUneErreurDePasDeTemps
      If assess_data FErr PasDeTempsTropPetit
      Then assess_operator AugmenterLePasDeTemps repair }
    ...
  Call
    syntax ./Mage5.exe < inputm.get_filename() endsyntax }
}
  
```

FIG. 6 – Extrait de la représentation du code MAGE et des connaissances associées en langage YAKL.

la pertinence du processus d'ingénierie des connaissances que nous avons engagé, et d'autre part d'ouvrir des perspectives vers une utilisation optimale des connaissances modélisées.

4.1 Retour d'expérience

En premier lieu, cette modélisation des connaissances sur le calage de modèles numériques a démontré que ces connaissances pouvaient être exploitées au sein d'un système d'aide au calage. Le prototype réalisé dans le domaine de l'hydraulique fluviale nous a permis de tirer plusieurs enseignements de cette expérience.

La modélisation et l'implémentation des connaissances s'avère satisfaisante, autant au niveau générique qu'au niveau du code de calcul utilisé. Seules les connaissances de raisonnement du domaine d'application apparaissent incomplètes, et ne peuvent prétendre recouvrir la diversité des cas de calage présentée par l'hydraulique fluviale 1-D. De nouveaux entretiens avec des experts hydrauliciens sont donc à programmer afin d'enrichir cette partie de la base de connaissances. Au cours de ces entretiens, le prototype réalisé servira de base pour déterminer précisément ces limitations.

La démarche de conception descendante adoptée a mis en valeur trois niveaux distincts de connaissances. Nos options de modélisation ne nous ont pas permis en premier lieu de préserver une indépendance totale entre les deux premiers niveaux – générique et hydraulique. Ces restrictions, imposées notamment par la syntaxe du langage de représentation de connaissances, ont conduit à de légères modifications dans l'implémentation de celui-ci.

Les connaissances de raisonnement issues du domaine d'expertise ont été modélisées sous la forme de règles de production, dont un exemple est donné par la Figure 7.

```

Rule {
  name ComparaisonEntreNiveauxDEauMesuresEtLigneDEauCalculee
  If assess_operator ComparaisonDesResultatsHydrauliques
    LigneDEauAuDessusDesNiveauxDEau ,
  Then send_operator DéterminationDesParametresHydrauliques
    RugositéTropImportante }

```

FIG. 7 – Exemple d'une règle initialisant un processus de réparation suite à une évaluation de la comparaison entre niveaux d'eau mesurés et ligne d'eau calculée. Les concepts mis en jeu dans cette règle sont propres au domaine de l'hydraulique (cf. Section 3.2).

La même structure de règle a été utilisée pour modéliser des connaissances du premier niveau, c'est-à-dire celles correspondant à la tâche générique de calage. Le principal problème réside justement dans le fait que nous avons dû expliciter dans la base de connaissances des éléments communs à tout calage de modèle numérique. Le constat de ce problème, provenant de l'utilisation d'un moteur

de pilotage de programmes, nous a donc suggéré de proposer des spécifications pour un moteur mieux adapté à la tâche de calage.

4.2 Vers un moteur d'inférence dédié au calage de modèles

La base de connaissances réalisée pour le prototype contient, au-delà des connaissances liées à la tâche générique de calage, toutes les sous-étapes de la structure d'inférence associée à cette tâche. Ces sous-étapes ont été introduites via des artefacts, en utilisant les notions à la base du pilotage de programmes, et notamment la planification. L'emploi du langage YAKL et du moteur d'inférence associé impose donc une réécriture systématique de ces parties communes de la base de connaissances.

Afin de pallier ces inconvénients, nous proposons d'intégrer la structure d'inférence associée à la tâche de calage dans un moteur dédié, dérivé du moteur de pilotage qui nous a servi pour réaliser le prototype. Dans cette optique, un langage approprié a été élaboré en reprenant les bases du langage YAKL. Ainsi, certaines classes présentes en pilotage de programmes, comme par exemple *données* ou *paramètres*, ont été simplement dérivées, et d'autres plus spécifiques ont été créées pour modéliser les notions de *système physique*, d'*événement* ou de *modèle numérique*. Le nouveau moteur devra ainsi être à même de manipuler ces différentes entités.

5 Conclusions et perspectives

Le processus d'ingénierie des connaissances engagé sur la tâche de calage de modèle nous a conduit à réaliser un prototype opérationnel dans le domaine de l'hydraulique fluviale monodimensionnelle.

La modélisation des connaissances s'est articulée autour d'une décomposition selon trois niveaux : les connaissances liées à la tâche générique de calage, les connaissances propres au domaine d'application, et les connaissances spécifiques au code de calcul utilisé. Les deux derniers niveaux de connaissances constituent une capitalisation conséquente, d'une part de l'expertise liée au calage de modèles hydrauliques, et d'autre part des connaissances nécessaires à une utilisation optimale du code de calcul MAGE.

Le premier niveau de connaissances a quant à lui permis d'explicitement la structure d'inférence de la tâche de calage générique. Cette structure va servir de base à la constitution d'un moteur d'inférence dédié à la tâche de calage de modèles numériques, qui permettra d'assister les utilisateurs de codes de calcul dans cette activité.

Références

ABBOTT M. B., BABOVIC V. M. & CUNGE J. A. (2001). Towards the hydraulics of the hydroinformatics era. *Journal of Hydraulic Research*, **39**(4), 339–349.

- BRUAUX S., KASSEL G. & MOREL G. (2003). Étude critique de la méthode ComonKADS – Application au calage de codes de calcul. In *Actes de la conférence Ingénierie des Connaissances IC'2003*, Laval, France: ESIEA Recherche.
- CHAU K. W., CHUNTIAN C. & LI C. W. (2002). Knowledge management system on flow and water quality modeling. *Expert Systems with Applications*, **22**(4), 321–330.
- CUNGE J. A. (2003). Of data and models. *Journal of Hydroinformatics*, **5**(2), 75–98.
- GIRAUD F., FAURE J.-B., ZIMMER D., LEFEUVRE J. C. & SKAGGS R. W. (1997). Hydrologic modeling of a complex wetland. *Journal of Irrigation and Drainage Engineering*, **123**(5), 344–353.
- KNEPELL P. L. & ARANGNO D. C. (1993). *Simulation Validation: a Confidence Assessment Methodology*, volume 3512 of *IEEE Computer Society Press Monograph*. Los Alamitos, California: IEEE Computer Society Press, first edition.
- MOISAN S. (2002). Knowledge representation for program reuse. In F. VAN HARMELLEN, Ed., *Proceedings of the 15th European Conference on Artificial Intelligence, ECAI-2002*, volume 77 of *Frontiers in Artificial Intelligence and Applications*, p. 240–244, Lyon, France: IOS Press.
- MOISAN S. (2003). *Program Supervision: YAKL and PEGASE+ Reference and User Manual*. Research Report 5066, INRIA Sophia-Antipolis, Projet Orion, Sophia-Antipolis, France.
- MOISAN S. & ERMINE J.-L. (2000). Gestion opérationnelle des connaissances sur les codes. In P. TCHOUNIKINE, Ed., *Actes de la conférence Ingénierie des Connaissances IC'2000*, Toulouse, France.
- MULLER P.-A. & GAERTNER N. (2000). *Modélisation objet avec UML*. Paris, France: Eyrolles, deuxième édition.
- NOUVEL P. (2002). *Enquête sur le concept de modèle*. Science, histoire et société. Paris, France: Presses Universitaires de France.
- PICARD S., ERMINE J.-L. & SCHEURER B. (1999). Gestion des connaissances pour des grands logiciels de calcul scientifique. In R. TEULIER, Ed., *Actes de la conférence Ingénierie des Connaissances IC'99*, p. 171–180.
- REFSGAARD J. C. & HENRIKSEN H. J. (2002). Modelling guidelines – A theoretical framework. In J. C. REFSGAARD, Ed., *HarmoniQUA – State-of-the-Art Report on Quality Assurance in Modelling Related to River Basin Management*, chapter 3. HarmoniQUA.
- SARGENT R. G. (2001). Some approaches and paradigms for verifying and validating simulation models. In B. A. PETERS, J. S. SMITH, D. J. MEDEIROS & M. W. ROHRER, Eds., *Proceedings of the 2001 Winter Simulation Conference*, Arlington, Virginia: Informs College on Simulation.
- SCHLESINGER S., CROSBIE R. E., GAGNÉ R. E., INNIS G. S., LALWANI C. S., LOCH J., SYLVESTER R. J., WRIGHT R. D., KHEIR N. & BARTOS D. (1979). Terminology for model credibility. *Simulation*, **32**(3), 103–104.
- VIDAL J.-P., MOISAN S. & FAURE J.-B. (2003). Knowledge-based hydraulic model calibration. In V. PALADE, R. J. HOWLETT & L. C. JAIN, Eds., *KES'2003 – Proceedings of the seventh International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, volume 2773 of *Lecture Notes in Artificial Intelligence*, p. 676–683, Oxford, U.K.: Springer.