Modélisation déclarative et sémantique, ontologies, assemblage et intégration de modèles, génération de code

Declarative and semantic modelling, ontologies, model linking and integration, code generation

Introduction (^)

Ce dossier traite de **la représentation des interfaces des modèles dans une ontologie**. Il est question des bénéfices retirés de cette approche lorsqu'il s'agit d'assembler et d'intégrer des modèles (partage, réutilisation, qualité, fiabilité, robustesse). Il est présenté l'ontologie MIO (Model Interface Ontology) et une mise en pratique de l'approche sur le projet Seamless-IP par la communauté APES (Agricultural Production Externalities Simulator).

Source (^)

Ce dossier repose sur un article qui est paru lors de la conférence de l'**iEMSs 2006** (International Environmental Modelling and Software Society) :

« Enriching software model interfaces using ontology-based tools »

I.N. Athanasiadis [a], A.E. Rizzoli [a], M. Donatelli [b], and L. Carlini [b]
[a] Dalle Molle Institute for Artificial Intelligence, Lugano, Switzerland
[b] CRA – Research Institute for Industrial Crops, Bologna, Italy

http://www.iemss.org/iemss2006

Chemin d'accès complet

- URL http://www.iemss.org/iemss2006 ,
- sous la rubrique « Sessions » ,
- puis « S5. Integrated software solutions for environmental problems - architecture, frameworks and data structures (ISESS) (Dave Swayne and Rob Argent) » ,
- puis « Theme 4: Knowledge Engineered Modelling ».

Relevé de quelques éléments [(^)]

Il est repris ci-dessous – sans aucune complétude - quelques éléments de cet article, dont il est fait des citations (texte entre guillemets).

Context, situation [(^)]
Le contexte, la situation

From the modelled system to the software form of the model
Du système modélisé à la forme informatique du modèle

Simulating a system includes :

→ Firstly, the modelled system : the very complete and complex real system.

→ Secondly, the model : « an abstraction of the real world processes », « using a given formalism » ; « particular assumptions and hypotheses about the phenomena involved are made ».

→ Thirdly, the software implementation of the model : « a poor realization of the original formalisation » ; « more assumptions, more limitations » are introduced (discretization…).

Models are seldom reused
Les modèles sont rarement réutilisés

« Common practice has proven that software implementations of environmental models are seldom reused by broader communities or in different modelling frameworks. One of the reasons for this situation is the poor semantics of model interfaces. Model interfaces describe a critical amount of modellers' knowledge, but their software implementations fail to represent the complexity of model assumptions in software terms. »

What « integrating models across scales and disciplines » implies
Ce qu'implique le fait d'intégrer des modèles de différentes échelles et disciplines

« Software integration is not the sole necessary condition for a proper assemblage of […] models. »

« If a set of (good) software model implementations are working together, this is not at all a sign that the compound model makes any sense from a modelling point of view and generates credible results. »

« Sound integration of […] models also requires automated coupling of the knowledge hidden behind each software implementation. »

« A declarative approach for describing model interface to facilitate model linking and integration using ontologies » (^)

Adopter une approche déclarative pour décrire l'interface des modèles, de sorte à faciliter l'assemblage et l'intégration des modèles grâce aux ontologies

« Sound model linking and integration »
Assembler et intégrer correctement des modèles (pertinence, cohérence, exactitude)

« A software component implementing a model will consist of two parts, the interface and the implementation. The interface defines the inputs, outputs and parameters of a model, while the implementation defines the model equations. »

« Simple integration in software terms is not enough for sound model integration ». In usual approaches « a software implementation of a […] model does not take into account the semantics of the software interface ». « The information associated with the inputs, states, outputs and parameters is limited to their data type », which is « not enough for encapsulating the full knowledge of the model interface ». As a result, « someone has to read the documentation in order to understand how to reuse [a] model properly ».

The problem (the lack) is that « the model's knowledge related to its interface is not encapsulated in the actual interface of the model implementation in a self-explained fashion ».

A solution would be « to express all the knowledge related to the model interface in a declarative way, using an ontology » (expressed concept, characteristic times, units, pre- and post- conditions, temporal or spatial dimensions, sampling rates…). « Ontologies provide a formal support to express conceptualisations […]. Furthermore, model knowledge stored in the ontology can be used both for formal documentation and provide functionalities which go beyond the computation of model variables ».

The Model Interface Ontology, « an ontology for specifying model interfaces »
Model Interface Ontology, une ontologie pour spécifier les interfaces des modèles

The Model Interface Ontology is an ontology « that aims to encapsulate our knowledge on the model interface in a declarative fashion ».

In order to represent « biophysical agricultural models », two model types are identified in the Model Interface Ontology :

- **Static** models (« not required to be integrated over time ») which have inputs and outputs.
- **Dynamic** models (integrated over time) which have inputs and outputs, states and rates (for stocks and flows).

Measurement : is a class of the Model Interface Ontology. « The Measurement class is the key instrument for conceptualising the model interface elements ». « All inputs, outputs, states and rates of models are types of an abstract Measurement concept (ontology class), which is used for defining their semantics in different contexts (space, time units, and so on) ».

Measurement **properties** : « The Measurement class specifies the following properties of a model interface element :

- The observed quantity.
- The spatial observation context.
- The temporal observation context.
- The sampling frequency.
- Value conditions (minimum, maximum and default value and default unit). »

For more details, see the article in Section 3 « Towards an ontology for specifying model interfaces », Figure 2 « The relations between the model type concepts of the model interface ontology » and Figure 3 « The relations of the Measurement concept ».

The Model Interface Ontology development
Le développement de l'ontologie Model Interface Ontology

The Model Interface Ontology has been developed using the Web Ontology Language **OWL** , through the **Protégé** ontology editor (http://protege.stanford.edu/plugins/owl).

« The specifications of units and dimensions were based on the **SWEET** ontologies » (http://sweet.jpl.nasa.gov).

« The **Model Interface Ontology** is available online » (http://seamless.idsia.ch/ontologies , MIO ontology).

Tools (^)
Outils

On the Seamless-IP project (http://www.seamless-ip.org)
Sur le projet Seamless-IP

The approach, consisting in « publishing model interfaces in a declarative format » using « an ontology for capturing the semantics of model interface elements », « was undertaken by the Seamless-IP project and the community of Agricultural Production Externalities Simulator (APES) modellers ».

A tool has been developed « to enable modellers to share their knowledge related to environmental model components and their interface variables » : AgrOntologies.

A tool has been developed « to enable modellers to exploit the knowledge stored in the ontology by generating source code in an automated fashion » : DCC.

« AgrOntologies : a Web-based tool for communal ontology authoring »
AgrOntologies : un outil Web pour l'écriture collaborative d'une ontologie

The AgrOntologies tool is « an easy to use portal » for modellers to « register their models ». Modellers are not « exposed to all the complexity of the internal ontology structure ».

« Through the AgrOntologies portal, a modeller can » :
- « Specify model variables in detail, or even reuse existing variables defined by others ».
- « Define model interfaces ».
- « Put models together in components ».

« DCC : a tool for generating model source code »
DCC : un outil pour générer un code source du modèle

« The application Domain Class Coder (DCC) is a Windows application » which, from an input file extracted from the ontology generates C# code.

See URL : http://www.isci.it/tools (page « Windows XP utility applications »).

Different ontologies uses (^)
Différentes utilisations des ontologies

Ontologies have been exploited in Environmental Management Information Systems with different aims :

- For « seamless integration of environmental data repositories ».
- « More generic approaches for environmental data fusion ».
- « For efficient model integration » [(a)].
- « Focusing on extending the current framework by specifying model equations using semantic modelling primitives » [(b)].

« Ontology representations of both model interfaces [(a)] and equations [(b)] may lead [us] to a fully declarative modelling and simulation environment »

(a) **Ontology representation of model interfaces :**

« For model linking and model component integration ». *C'est le sujet de l'article.*

(b) **Ontology representation of model equations :**

« Specifying model equations using semantic modelling primitives ». *Voir le dossier modelia de l'article « Declarative modelling for architecture independence an data/model integration : a case study » de Ferdinando Villa.*