
Modélisation déclarative, ontologies, génération de code, modèles indépendants des plates-formes

Declarative modelling, ontologies, code generation, platform independent models

[Introduction](#)

[Source](#)

[Relevé de quelques éléments](#)

Introduction [\(^\)](#)

Ce dossier traite de **modélisation déclarative**, des **ontologies** qui permettent de considérer les modèles indépendamment des environnements et plates-formes informatiques sur lesquels ils « tournent » (sont simulés, exécutés).

Source [\(^\)](#)

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

« Declarative modelling for architecture independence and data/model integration : a case study »

Ferdinando Villa, University of Vermont, USA,
Marcello Donatelli, ISCI, Bologna, Italy,
Andrea Rizzoli, IDSIA, Lugano, Switzerland,
Peter Krause & Sven Kralisch, University of Jena, Jena, Germany,
Frits K. van Evert, PRI, Wageningen, The Netherlands

<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 ici – sans aucune complétude - quelques éléments de [cet article](#), dont il est fait des citations (texte entre guillemets).

Some vocabulary ^(^)

Du vocabulaire

Conceptualization : « A conceptualization is the objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold them. »

Modelling paradigm : « The set of abstractions that allows conceptualizing and expressing [the] cause-effect relationships [of the model] and their results is the adopted modelling paradigm. »

Declarative modelling : « The wording declarative modelling [...] has been used to identify a specification of models that is based on the semantics of the natural systems being modelled rather than the algorithms that calculate their changing states. »

Knowledge database : « A database that contains both the ontology and the instances that populate it is usually called a knowledge database. »

« The need for integrating dynamic models with independently developed datasets and other models » ^(^)

Le besoin d'intégrer des modèles dynamiques avec des données et autres modèles développés indépendamment par ailleurs

The declarative modelling « decouple[s] the representation of a model from its executable implementation ». « Declaratively expressed models are independent on architectural and software details, and need to be implemented into working algorithms before they can be simulated ». So the declarative modelling can lead to « the creation of repositories of models where the models' lifetime is not tied to that of specific modelling paradigms, execution architectures, or storage technology ».

« Declarative models only depend on the conceptualization (paradigm), and are thus easier to be exchanged and communicated as long as the basic conceptualization is agreed upon ». Declarative specifications « provide portability and facilitate integration between independent, uncoordinated data and models ».

Declarative modelisation content : ontologies and instances ^(^)

Contenu de la modélisation déclarative : ontologies et instances

Ontologies : « Declarative modelling can be supported by ontologies ». « In recent years, the use of ontologies has become commonplace to express formal conceptualizations in mutually understandable ways. Ontologies are expressed using common representational languages such as XML and they are optimized for web-based access and sharing. »

Instances : « Given an ontology describing model concepts, instances are defined to declaratively express specific models by referring to concepts laid out in the ontologies. »

Domain concepts and model concepts : The ontology « specifies the domain concepts and the model concepts. The domain concepts represent the entities (the variables and parameters), which are used by the model concepts (the causal relationship, expressed as equations) to describe the model itself ».

Illustration : « *Temperature* is a concept, *airTemperature* and *greenHouseTemperature* are instance of such concept which are also described via specific attributes, and $greenHouseTemperature = airTemperature * k$ is a model concept ».

Reliability and consistency : Ontologies and instances declaration contains the information that « enable[s] a software execution environment to simulate the behaviour of the model over a temporal and spatial extent. Moreover, [...] the rich meaning made possible by current ontology frameworks » allows to « properly connect models and data », « without risking the error resulting from matching inputs and outputs that specify different natural-world entities ».

Causal relationships and declarative approach / non declarative (“conventional”) approach ^(^)

Relations causales et approche déclarative / approche non déclarative (« conventionnelle »)

« A conceptualization capable of describing scientific models requires, at minimum, to express the notion of linkage between scientific observations and their change in time and space by causal relationships. »

« Causality in conventional models is usually expressed through equations, defined to calculate the value of variables. Equations, by naming the values of other variables, implicitly define causal relationships that are viewed as dependencies from a processing point of view. An ontology-based framework can make these dependency relationships explicit, and add meaning to them by means of specializing the kind of relation. »

Automation ^(^)

Automatisation

« The implementation of a declarative model is [...] delegated to an algorithm, rather than a human programmer. » Consequently, « the modeller can focus on the modelling details rather than the implementation ones : the modeller is forced into adopting tight, well-reasoned, and usually better description paradigms ».

The knowledge database ^(^)

La base de données des connaissances

The knowledge database is « a database that contains both the ontology and the instances that populate it ». « The knowledge base can contain information on data, models, scientific workflows and, in general, on all the concepts that are relevant to our modelling domain. »

« Storing a model in a knowledge base involves :

- expressing its logical structure by subdividing the model into components and
- mapping each component to a concept defined in the same storage. »

From the knowledge database to code ; from the declarative specification to a targeting modelling framework ^(^)

De la base de données des connaissances au code ; de la spécification déclarative à une plate-forme de modélisation cible

Les paragraphes « 2.3. From knowledge base to code » et « 3. CASE STUDY » de [l'article](#) présentent comment, à partir de la forme déclarative du modèle, il peut être généré automatiquement la forme logicielle dédiée à une plate-forme particulière de modélisation/simulation (c'est-à-dire qui permet de « faire tourner », d'exécuter le modèle dans l'environnement cible considéré). Le processus est ici illustré notamment avec les plates-formes [JAMS](#) et [ModCom](#). Une telle conversion est effectuée au moyen d'outils/composants génériques dédiés au processus de génération, qui sont spécifiques de chaque plate-forme/environnement de simulation.

« The declarative structure » can be « defined by means of a formally specified XML schema ». Indeed « an XML schema can be seen as a relatively informal ontology, where the meaning is suggested, if not formally identified, by the names of the node identifiers, and basic relationships are captured by the structure of containment of nodes within other nodes ».

A model can be « declaratively represented and translated to [...] different target architectures » : « [The Domain Concepts and the Model Concepts] can be expressed using the XML Schema Definition Language (XSD), and the instances of the concepts are XML files which can be processed and translated into software components targeting various modelling framework platforms. »

« The conversion of a model from its declarative format into a procedural form, ready to be compiled and executed, can be automated, following a principled approach » : « The conversion algorithm, implemented in the tools » will « parse all the model concepts, retrieving the data type information from the domain concepts in order to create the interface of the model component class in the target language or modelling framework. The model classes are implemented as components, exposing an interface for

their use in various contexts and environments. The interface varies according to the target framework, but in general it will include accessor methods, to set input values and get outputs, execution methods, to fire the model equations, and test methods, including pre and post-conditions. The code of the body of the methods is then generated according to the content of the state transition and output transformation equations. »

« In order to perform a translation from a declarative specification to a working software component, the software infrastructure must share and understand the conceptualization. The most sophisticated systems can read explicit ontologies and infer the simulation algorithms by reasoning on the contents of the ontologies. Other, simpler systems can be envisioned that are tied to one or a few specific ontologies [...] and produce algorithms that calculate them in a high- or low-level programming language, refusing to handle any knowledge that can't be interpreted in those terms. »

JAMS (Jena Advanced Modelling System) ([^](#))

« The Jena Advanced Modelling System (JAMS) provides a framework for the development, composition and application of distributed and process oriented environmental models » ...

ModCom ([^](#))

« ModCom is a modelling framework that was first described by Hillyer et al. (2003). Recent developments, including a C# implementation, are available online (Anonymous, 2006) » ...

- *mise en ligne le 27/09/06* – [La page au format pdf](#) (27/09/06)

Plate-forme INRA-ACTA-ICTA, Modelia <http://www.modelia.org>
