

# AM3: ATLAS MegaModel Management

## 1 Context

Applying Model Driven Engineering (MDE) to the design, running, administration and maintenance of complex computer systems means that we need to cope with a very high number of well defined and precisely focused independent metamodels, each one corresponding to a given DSL (Domain Specific Language). Using model transformation and model composition to solve semantic bridging problems means that we need to work with large libraries - private or public, centralized or distributed - of transformations and compositions. Recent undertakings in the MDE community recognize that dramatic increases in complexity of systems need new solutions. Second generation MDE environments may help answering these problems if they are able to scale up in the presence of a rapidly growing number of modeling artifacts representing different concern facets, different stakeholder views, different development states of composite complex systems while offering a regular management interface to them. This project is about building tools that will simplify the handling of model engineering tasks when the number of involved artifacts will be in the order of the hundred or more. It will try to propose some help in selecting and adapting adequate MDE COTS (Component-of-the-Shelf) when these components have to be found in large distributed Web spaces.

If we wish to apply MDE to solve complex problems, it is likely that the number of resources involved will be quite high. In order to answer the corresponding global management question, we need not only to understand how to build the individual components themselves (models, metamodels, transformation composition, etc.), but we also need to invent new ways to cope with the large structured repositories of these components. The present situation in modeling languages and environments is similar to the past situation in programming languages and environments. Following de Remer and Kron [1] we can target "Modeling-in-the-Small versus Modeling-in-the-Large". Following Wiederhold, Wegner and Ceri [2] we can target "Megamodeling". In any case, we need to address global coordination between MDE resources (models, metamodels, transformations, semantic correspondences, etc.) located anywhere, for example on Web repositories. We want to apply integrated MDE techniques to help solving this problem. "Integrated" means that we believe the artificial complexity introduced by the use of MDE should not be dealt with classical programming level tools (like *make*, *ant*, etc.) but with model engineering tools. Our initial vision of global model management is described in [3]. Now the time has arrived to provide the tools to support this vision. The AM3 Eclipse/GMT subproject is intended to host these experimental tools.

## 2 Objectives

The goal of **AM3 (ATLAS MegaModel Management)** is to provide a practical support for modeling in the large. The objective is to deal with global resource management in a model-engineering environment. We base this activity on the concept of a "megamodel". A megamodel is a registry of model engineering resources available in a given scope (a zone). AM3 assumes that tools for model transformation and model composition already exist and are available. These facilities may come for example from ATL, VIATRA2, UMLX or AMW but other choices are also possible. The basic idea is that there is no unique metamodel for megamodels. Instead, the user may use metamodels from a library or invent their own ones. This means that the kind of artifacts stored in or retrieved from a given MDE repository may vary. We know some of these artifacts (models,

metamodels, transformations, semantic correspondences, etc.) but many others may be contextually defined. This is one of the difficulties of this global model management: the tools should be metamodel-agnostic in order to handle this contextual variability.

### 3 Approach

Global management tools have plenty of potential applications. They can provide a lot of leverage over common problems. The idea of modeling in the large is to establish and use global relationships and metadata on the basic macroscopic entities (mainly models and metamodels), ignoring the internal details of these global entities.

A megamodel is a model recording global information on tools, services and other models. A model may be a terminal model, a metamodel or a metametamodel. Models are considered as a whole together with tools, services and other global entities. Within a model management environment, the main elements produced or consumed are models, metamodels or transformations. With the AM3 supported megamodels, we want also to allow the manipulation of other resources such as XML documents, database tables or flat files. For this reason, we have introduced the notion of artifacts as basic entity. A megamodel gives the representation of the user-working context (a zone). A zone is a set of resources and each zone is represented by a given megamodel. It is the set of all the artifacts, which are accessible in a given context. One goal of the megamodel management tool is to manage artifacts with all necessary functionalities, and to allow executing the control actions associated with the artifacts.

AM3 is part of the AMMA platform (ATLAS Model Management Architecture). The AMMA platform has been implemented on top of the Eclipse/EMF framework. The AM3 plug-in is based on the availability of the megamodel's metamodel. The main features of AM3 are:

- Management of various artifacts,
- Management of various relations between artifacts,
- Sharing and exchange of megamodel elements,
- User interfaces for viewing (browsing, creating, changing, etc.) megamodel elements.

### 4 AM3 subproject content

AM3 subprojects can be separated into two parts:

- AM3 framework,
- AM3 resources.

The MDE resources available in the AM3 subproject will allow illustrating the use of the various facilities provided in the framework.

#### 4.1 AM3 framework

AM3 framework provides an environment for modeling in the large with the following basic initial features:

- Managing megamodels
  - Adding, editing metadata and deleting artifacts
  - Adding, editing and deleting relations
- Using megamodels
  - Sharing artifacts
  - Searching artifacts
- Managing tools and services
  - Using tools with artifacts
    - Running transformations
      - Model transformations
      - Projectors

- Injectors
- Extractors
  - Launching several transformations in batch mode
- Editing artifacts

In addition to these, it is very likely that additional functionality will be found useful and implemented in this environment.

## 4.2 AM3 resources

AM3 resources are all resources (e.g. metamodels) published under the AM3 subproject. These resources can be used in the AM3 framework. A set of metamodels and a set of projectors (injector and extractor) will initially be published.

However, the resources internal to the AM3 subproject will mainly be used in the initial period as illustrative of the possibilities. Progressively the tools will be applied to distributed and scattered resources. First the available artifacts will be found in the GMT project itself (ATL transformations, VIATRA2 transformations, UMLX transformations, AMW weaving metamodels, etc.). Then the idea is to reference other external distributed repositories like public libraries of metamodels or available sets of transformations.

## 5 Target audience / End users

As the project grows, we plan to publish some typical scenarios for using AM3. For example, one wishes to display a table of health care data. Discovering that a HL7-compatible metamodel exists maybe a first step. Discovering that a MS Excel metamodel can be used may be a second step. Discovering that, in a third repository, there is a transformation from similar input data to table representation may be a third step. Adapting the transformation and checking that a compatible engine is locally available may be the next step.

## 6 Initial contribution

The initial contribution of AM3 will be available before the end of 2005, for both resources and initial prototypes.

## 7 Work Plan and schedule

In the next year, new major releases are planned. It is hoped that new users and contributors will join the subproject as committers.

## 8 Current Development Team

The current core development team of the AM3 consists of

Project Leader: Professor Jean Bézivin

Principal Advisor: Frédéric Jouault

Chief Technological Developer: Freddy Allilaire

## 9 Bibliographical references

- [1] DeRemer, F., Kron, H.: *Programming-in-the-Large Versus Programming-in-the-Small*, IEEE Trans. on Soft. Eng. 2(2), (1976)
- [2] Wiederhold, G., Wegner, P., Ceri, S.: *Toward Megaprogramming CACM*, Volume 35, Issue 11, Pages: 89 - 99, (November 1992)
- [3] Bézivin, J, Jouault, F, Rosenthal, P, and Valduriez, P.: *Modeling in the Large and Modeling in the Small*. Lecture Notes in Computer Science, N°3599, edited by Uwe Aßmann, Mehmet Aksit, Arend Rensink. Springer-Verlag GmbH, pages 33-46.