Project acronym: MODELPLEX

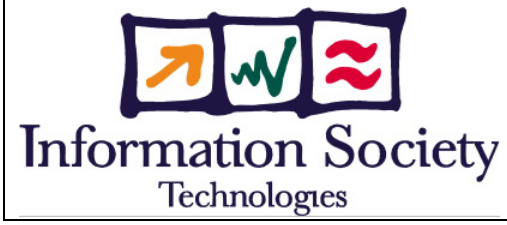
Project full title: MODELling solution for comPLEX software systems

Contract n° 034081

Workpackage 3: Model Engineering

Deliverable D3.2.d:

"Global Model Management Traceability Extension"



Project co-funded by the European Community under the "Information Society Technology" Programme

Contract Number: 034081

Project Acronym: MODELPLEX

Title: MODELling solution for comPLEX software systems

Deliverable N°: D3.2.d

Due Date: 12/2008

Delivery Date: 12/2008

Short Description:

This report describes an extension of the Global Model Management prototype, developed within Task 2.1, which provides support for traceability. The extension enables our partners to build and store cross-DSL and model-to-model transformation traces as well as corresponding models within the GMM environment.

Lead Partner: INRIA

Made available to: PU (Public)

Rev	Date	Author	Checked by	Internal Approval	Description
0.1	07/11/08	Draft – Hugo Bruneliere, INRIA			Initial version
0.2	21/11/08	Hugo Bruneliere, INRIA			First complete version
0.3	28/11/08	Hugo Bruneliere, INRIA			Various upgrades and screenshots adding
0.4	01/12/08	Contributions integrated from Hugo Bruneliere, INRIA		**1st Internal Reviewer** : Brian Pickering, IBM **2nd Internal Reviewer**: Mathias Fritzsche, SAP	Reviewers comments integration
1.0	15/12/08	Final deliverable	QB	EB	

Table of Content

1. Executive Summary 4
2. Acronyms and Terminology 4
3. Introduction 5
4. Motivation 5
5. Automated Traceability Support in M2M Transformation with ATL 6
   5.1. Automated Production of an Augmented Transformation 6
   5.2. Generation of Traceability Information 7
6. Traceability Navigation Support 9
   6.1. Generic Inter-Model Navigation Support 9
   6.2. Application of Inter-Model Navigation to Traceability 10
7. Demonstration on the SAP Use Case 14
8. Conclusion 16
9. References 17

List of Figures

Figure 1 AM3 UI for transformation launching 7
Figure 2 Generated traceability model (example) 8
Figure 3 Overview of our generic inter-model navigability solution 9
Figure 4 Application of our generic inter-model navigability solution to traceability (example taken from a part of the SAP use case) 10
Figure 5 AM3 UI for traceability navigation: model-level view 12
Figure 6 AM3 UI for traceability navigation: model element-level view 13
Figure 7 Overview of the SAP MDPE transformation chain (simplified) 14
Figure 8 Use of the AM3 tooling for traceability navigation (SAP use case) 15

1. Executive Summary

As described within [1], Global Model Management (GMM) aims to provide support for *modelling in the large*, i.e. managing global modelling resources in the field of MDE-oriented software development. These global resources are usually heterogeneous and distributed as they often come from different modelling tools. Thus, in order to be able to use them without unintentionally increasing the complexity of MDE, we need to invent new ways of creating, storing, viewing, accessing, modifying and using the metadata associated with all these global modelling entities. This is the reason why, within the context of Task 2.1, the concept of a *megamodel* was introduced and a corresponding prototype named *AM3* was developed (cf. [2]). The goal of our current work in Task 3.2 is to propose an extended version of this GMM prototype that adds some generic support for dealing with the different traceability links which may exist between various modelling artefacts.

This deliverable mainly focuses on describing the mechanisms and associated tooling which have been implemented in order to provide generic and integrated traceability support within the *AM3 prototype V1* [2]. This extended version of the GMM prototype is currently available from the MODELPLEX SVN [5]. It will also be published under the *Eclipse-GMT AM3* project [7].

2. Acronyms and Terminology

AM3	ATLAS MegaModel Management
AMW	ATLAS Model Weaver
ATL	ATLAS Transformation Language
DSL	Domain-Specific Language
GMM	Global Model Management
GMT	Generative Modeling Technologies
HOT	High-Order Transformation
M2M	Model-to-Model (Transformation)
MDE	Model Driven Engineering
MDPE	Model-Driven Performance Engineering
SVN	Subversion
TIPM	Tool-Independent Performance Model
WP	Work Package
UML	Unified Modeling Language

3. Introduction

From the beginning of the *MODELPLEX* project, extensive research work has been carried out centred on Global Model Management problems and on how to provide concrete solutions to them. The results of this investigation work concerns theoretical and conceptual aspects [1] but also more practical ones with the implementation of a concrete GMM prototype [2]. This work is being carried out within the context of Task 2.1 and continues to be worked on. The results obtained have also been improved, within the context of this present work on traceability in Task 3.2, thanks to the implementation of an extended version of the *AM3* GMM prototype.

To begin with, the mechanism allowing the automated generation of traceability models during the execution of ATL M2M transformations will be set out. Its implementation within the prototype will also be described. Then, the principles of generic navigation support will be introduced and explanations on how they are integrated within the prototype in the context of traceability will be provided. Finally, and before concluding, experiments on a part of the SAP use case with the two features previously introduced will be presented.

4. Motivation

Within the context of Task 2.1, an important piece of work has already been successfully completed in collaboration with SAP on their Model-Driven Performance Engineering use case. It mainly consists of building a complex chain of M2M transformations (with ATL), specifying the corresponding required models and metamodels, and using a megamodel (with AM3) in order to be able to store and retrieve explicitly the metadata which is associated with all these artefacts.

While developing such an MDE-based solution, the need to deal with traceability, not only within single M2M transformation but also within chains of such transformations, has been clearly identified. Indeed, it is required to be able to trace the different model elements within the whole chain of transformations (i.e. to navigate the various traceability links between them both forwards and backwards) from the original model to the finally produced simulation model.

In their requirement #303 "*Tool-support to deal with numerous modelling artefacts, transformations and relationships*" from the 24/04/08, SAP emphasizes the importance of supporting traceability within this context: "In order to support the specific needs of Model-Driven Performance Engineering and its application for the SAP Case Study (Composite Application) we need model management tool for the numerous kinds of artefacts. This needs to include facilities for dealing with traceability models."
This deliverable proposes a prototype which addresses this explicit requirement, and demonstrates the usability of the solution by its successful application to the SAP use case as an appropriate test.

5. Automated Traceability Support in M2M Transformation with ATL

When dealing with traceability, the first step is to build the traceability links effectively. This can be done manually but in the ideal case this should be performed automatically and systematically. This section describes how this step has been implemented in the context of ATL M2M transformation [9] and integrated within the AM3 prototype. First, the automated production of an augmented transformation which deals with traceability is presented. Then, the generation of the traceability information thanks to the execution of this augmented transformation is explained.

5.1. Automated Production of an Augmented Transformation

In order to provide some traceability support within a M2M transformation language like ATL [4], several solutions can be considered. The one which has been developed in Task 3.2 has already been described in detail within [3]. It uses a high-order transformation (HOT) that takes as input the initial transformation and then generates as output a new augmented transformation having exactly the same behaviour but also additionally producing the required traceability information. Thus, neither the initial transformations nor the ATL compiler and ATL virtual machine have to be modified in order to provide such traceability support.

The production of the augmented transformation is integrated in a transparent way (from the user point of view) within the extended version of the AM3 prototype. Indeed, when a user launches a registered transformation (composite or not) from the AM3 tooling and also chooses to generate the corresponding traceability information, the selected transformation is considered as the input to the HOT that produces (as output) the augmented transformation also dealing with traceability. Figure 1 shows the AM3 interface which allows the selected transformation to be executed and the automated trace generation feature to be activated.

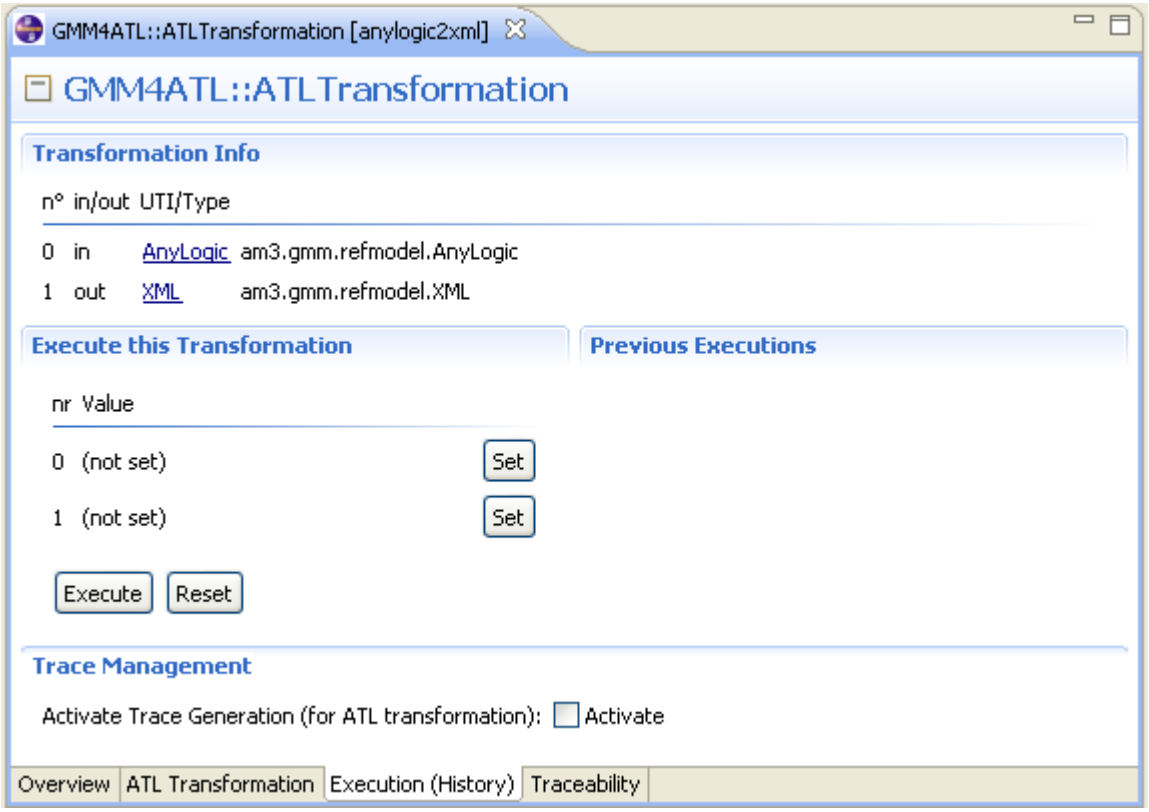


Figure 1 AM3 UI for transformation launching

Thus, the process of automatically building the augmented transformation as well as its effective launching is run in the background and so masked from the user.

5.2. Generation of Traceability Information

The augmented transformation adding traceability support having been produced, it can now be effectively executed in order to generate not only the expected resultant model(s) but also the corresponding traceability model. This execution is performed in such a way that the user does not see that the transformation which has been really launched is the augmented one rather than the initial one. Indeed, the outcome of the execution of this augmented transformation is the same as those expected from the execution of the initial one. The additionally generated traceability links are also available as they are stored in the automatically produced traceability model (see Figure 2).

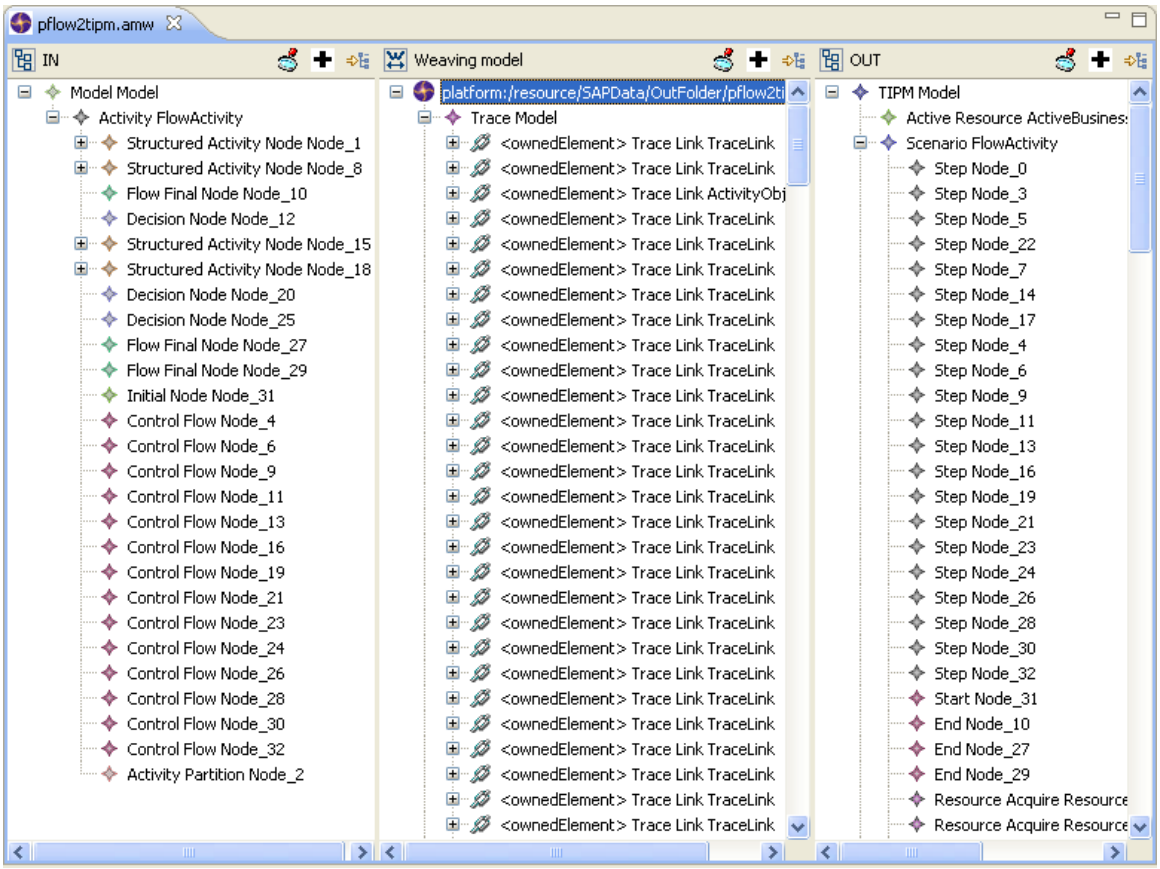


Figure 2 Generated traceability model (example)

As described in [3], the generated traceability models are weaving models. The previously shown traceability model, which provides a set of traceability links between various input model elements and output model elements, is edited using the AMW generic editor.

6. Traceability Navigation Support

When dealing with traceability, the second step is to be able to navigate the traceability links which have been previously produced (manually or automatically as explained in previous section 5) easily. This section presents how this step has been implemented within the extended version of the AM3 prototype. First, the generic support provided for inter-model navigation is described. Then, the application of this generic navigability support within the specific context of traceability is detailed.

6.1. Generic Inter-Model Navigation Support

Within Global Model Management and MDE in general, it is an important objective to be able to navigate between different interrelated models in a generic way. In order to provide such inter-model navigability support, two different but complementary navigation levels need to be considered:

- ***Model-level navigation***
- ***Model element-level navigation***

To be able to deal with these two levels in a generic way, different MDE techniques (and corresponding generic tools) must be combined:

- ***Global Model Management*** (AM3 [7]) for links at the ***model-level***
- ***Model Weaving*** (AMW [8]) for links at the ***model element-level***

The following figure provides an overview of our generic inter-model navigability solution which implies both previously mentioned technologies.

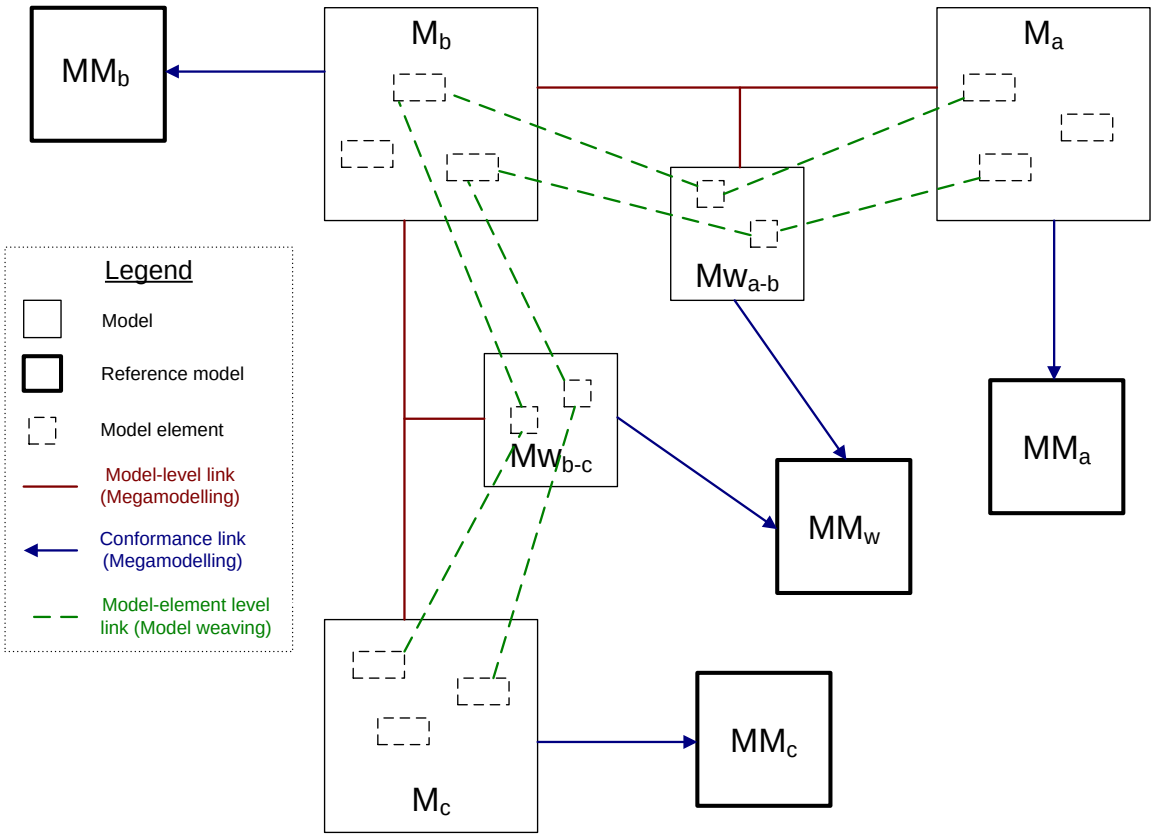


Figure 3 Overview of our generic inter-model navigability solution

Considering only two (or more) models containing a set of model elements, several links may be established between their elements. These links are explicitly stored within a separate weaving model which conforms to a specific weaving metamodel. Thus, model weaving facilities are used in order to represent model element-level links without having to modify the initial models (i.e. the woven models) artificially.

Next, considering a set of models (each of them conforming to a given metamodel), several different relationships or links may exist between them. All the implied models, metamodels, etc as well as the corresponding links are registered within the megamodel in such as way that they can be retrieved and used at any time. Moreover, each of these links at the model-level can be associated with a given weaving model providing the related links at the model element-level. This approach enables that all the required information is explicitly registered within the megamodel. Thus, Global Model Management facilities are used in order to represent model-level links and more generally to navigate the links at both model and model-element levels.

6.2. Application of Inter-Model Navigation to Traceability

There are several concrete applications of inter-model navigability from various MDE fields. An important one is the traceability domain. Thus, applying the previously presented principles on traceability in M2M transformation as set out in section 5, the big picture shown in Figure 4 is obtained.

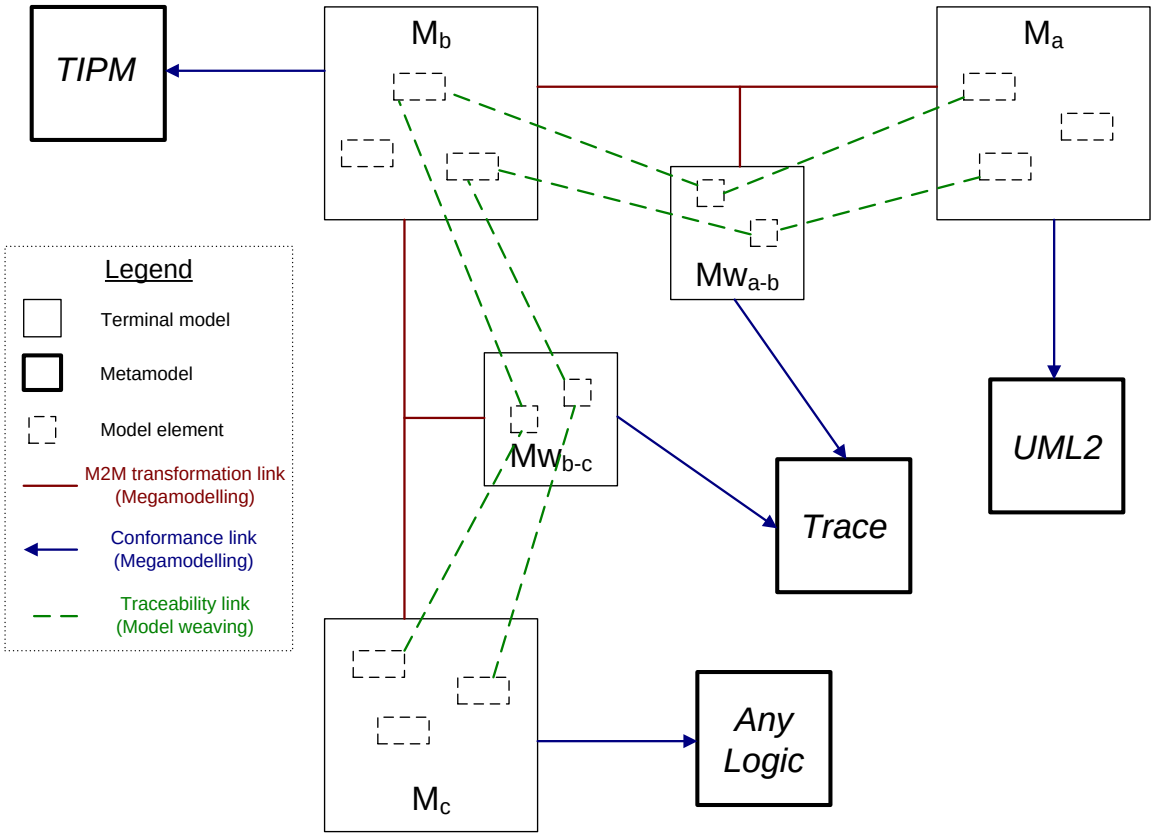


Figure 4 Application of our generic inter-model navigability solution to traceability (example taken from a part of the SAP use case)

In a similar way as shown in Figure 3, the traceability links between two models

are represented and stored within the megamodel and are detailed, at a model element-level, by traceability models. These traceability models are weaving models which conform to a traceability-specific metamodel. They are explicitly associated with the corresponding M2M transformation links within the megamodel so that they can be retrieved and directly used from the Global Model Management tooling.

To summarize, having the required metadata for all models (including traceability models) and relationships between them registered within a megamodel, the AM3 tooling can provide generic and extensible traceability navigation support.

Apart from the already available generic *Megamodel Navigator* and corresponding editors (cf. [2]), two navigability-specific views have been developed within the context of our work in Task 3.2:

- The ***Model-level Navigation*** view is about displaying the models which can be reached, by navigating a model-level link (a M2M transformation link in the present case), from the model currently selected in the *Megamodel Navigator;*
- The ***Model element-level Navigation*** view is about providing the model elements which can be reached, by navigating a model element-level link (a traceability link in the present case), from the model element currently selected in the model's standard editor.

Please note that these two views, even if they have been used within the context of traceability, are generic and can support any kind of inter-model navigability (as defined in section 6.1). Figure 5 and Figure 6 show examples of the use of the two AM3 navigation views in the traceability context.

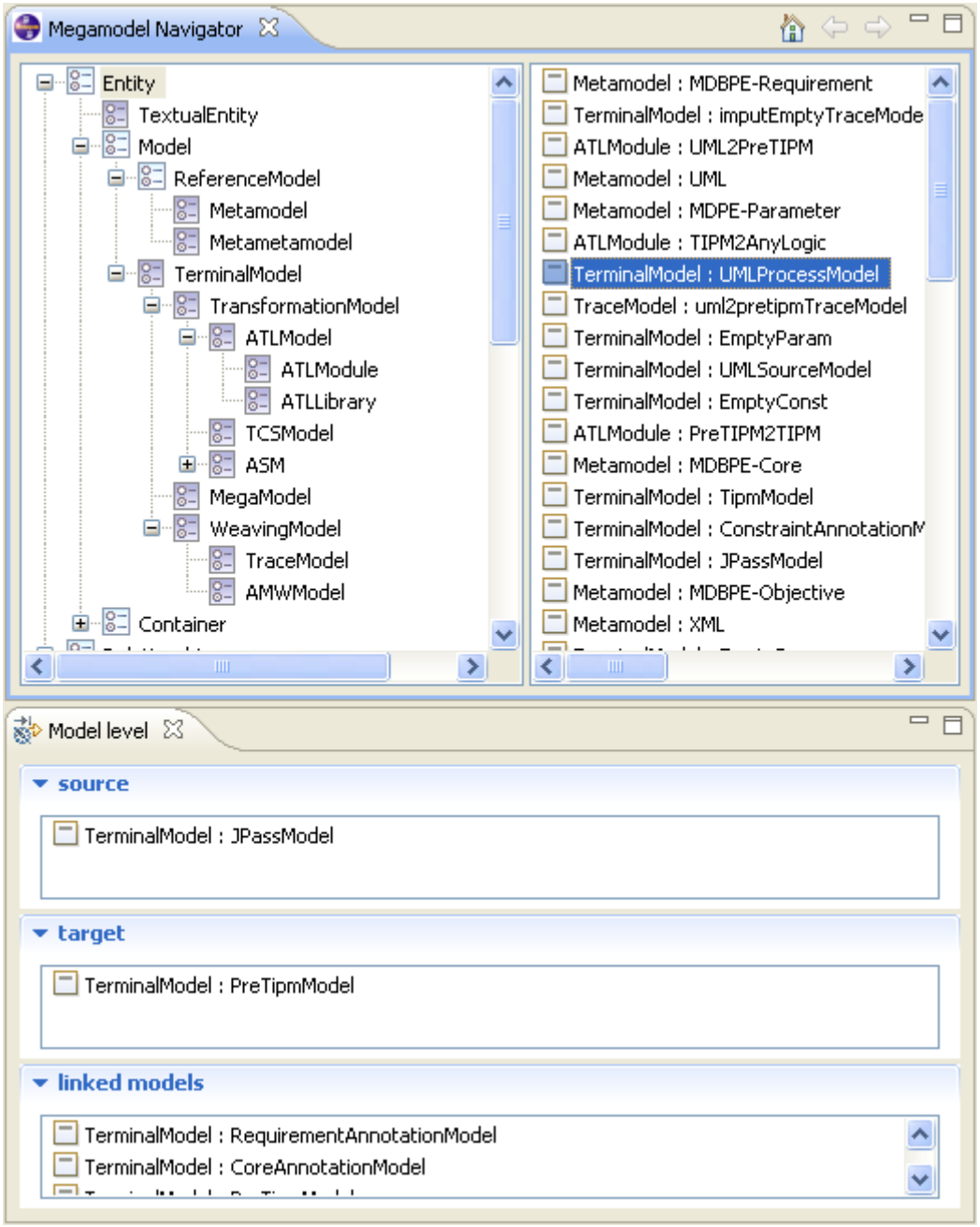


Figure 5 AM3 UI for traceability navigation: model-level view

Figure 5 shows how the model-level traceability links, corresponding to concrete M2M transformation executions, can be displayed and navigated using the provided AM3 interface. In this case, JPass-to-UML and UML-to-TIPM traceability links are considered (cf. section 7).

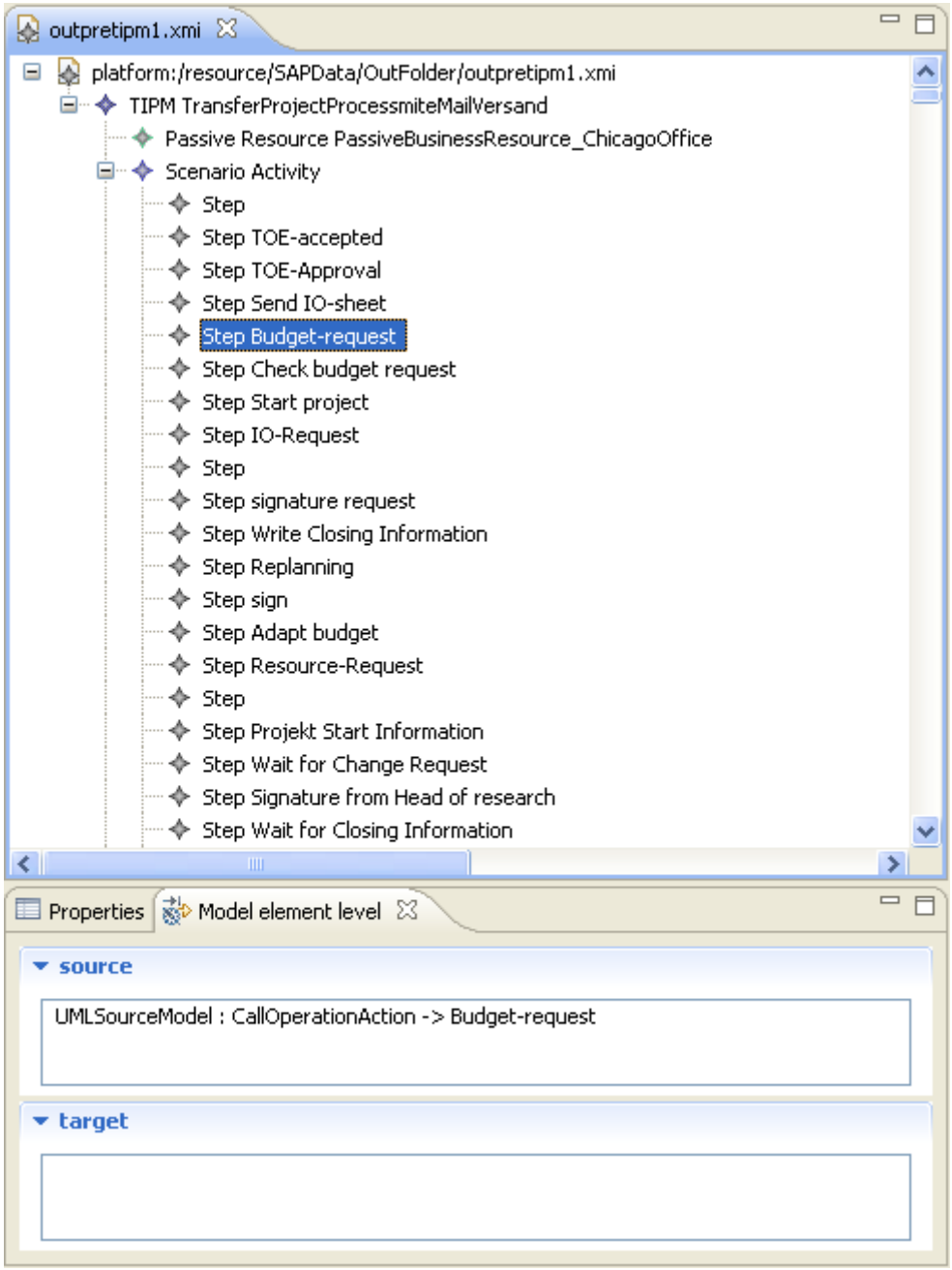


Figure 6 AM3 UI for traceability navigation: model element-level view

Figure 6 offers an example of how the model element-level traceability links, stored as independent traceability models (which are weaving models), can be browsed transparently thanks to the AM3 interface. In this case, a UML-to-TIPM traceability link is made explicit at the model element-level (cf. section 7).

7. Demonstration on the SAP Use Case

This section explains how the previously presented approach and corresponding implemented features within *AM3* have been applied to a concrete use case taken from the SAP Model-Driven Performance Engineering (MDPE) process.

The example shown here is the transformation chain from UML to the AnyLogic™ simulation tool. This chain is first and foremost about generating a tool-independent performance model (TIPM) from a UML2 activity model. Then, this TIPM model is transformed into a simulation-tool-specific model conforming to an AnyLogic metamodel. This AnyLogic model is used in order to produce an XML model that can be finally extracted to obtain a file which can be directly processed by the AnyLogic tool. Figure 7 provides an overview of this transformation chain and related artefacts.

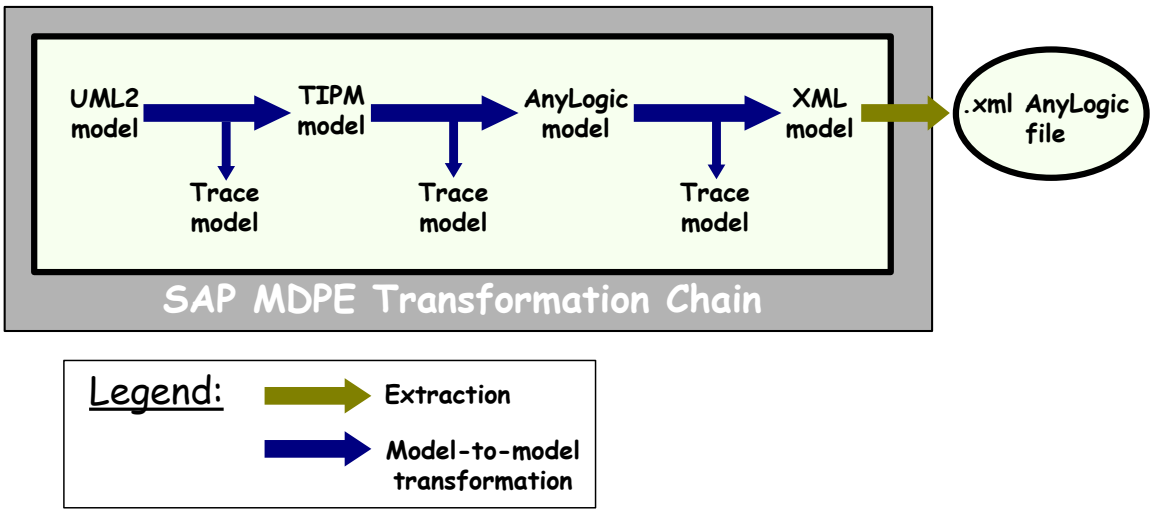


Figure 7 Overview of the SAP MDPE transformation chain (simplified)

The objective is to use the extended version of the *AM3* prototype in order to:

- Generate the traceability model at each step of the transformation chain automatically;
- Process these generated traces to navigate back from the final AnyLogic model to the initial UML model (and *vice-versa*).

Figure 8 shows how the *AM3* tooling can be used, for traceability navigation purposes, within the considered SAP transformation chain. In this case, the traceability links between the UML model and the TIPM model are displayed at both levels (i.e. model-level and model element-level) thanks to the *AM3 Megamodelling* perspective: thus, they can be browsed directly.

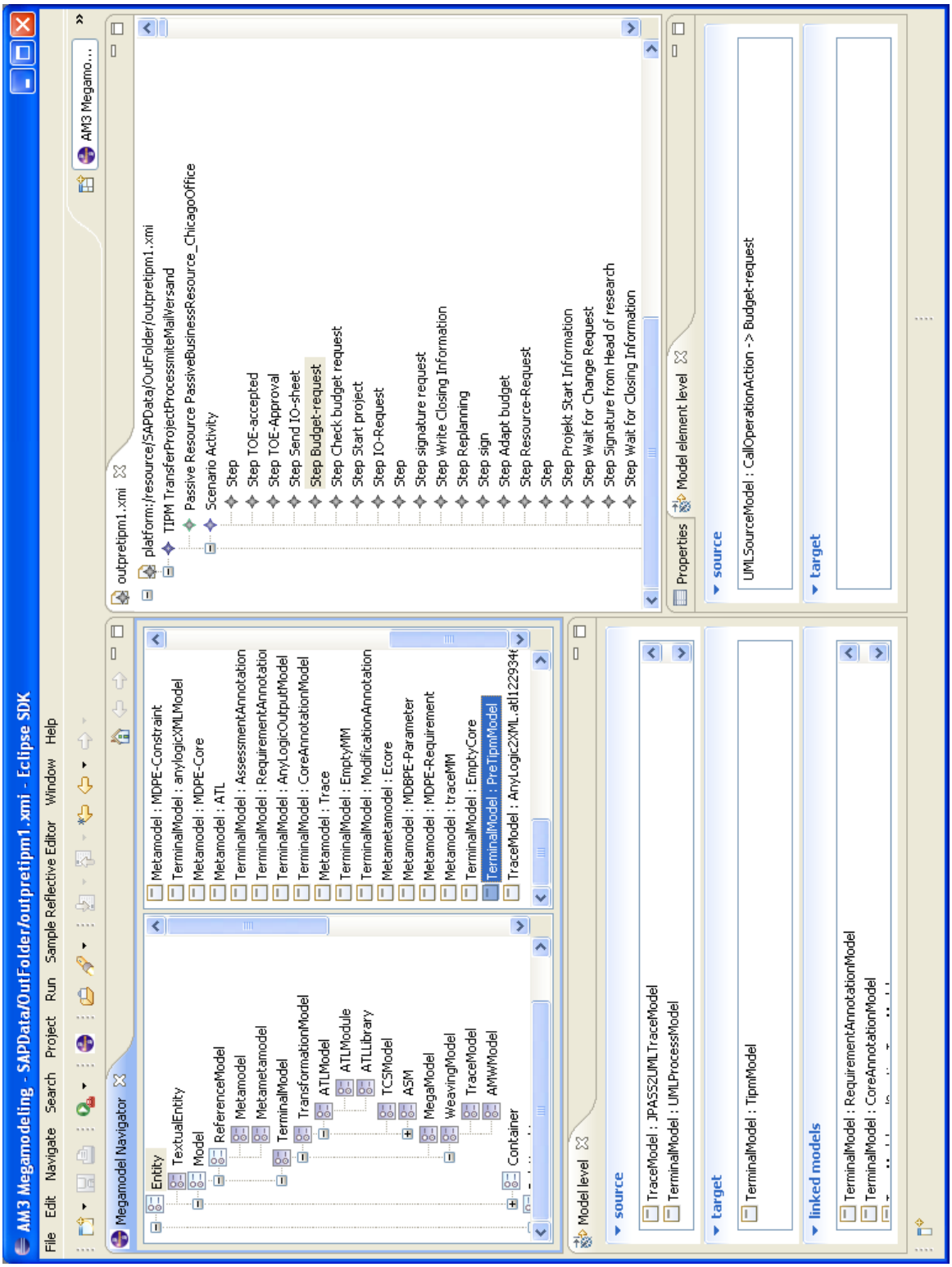


Figure 8 Use of the AM3 tooling for traceability navigation (SAP use case)

All the required *AM3* plugins providing this extended traceability support are available from the MODELPLEX SVN [5].

The demonstration of using the extended version of the *AM3* prototype on this SAP use case is also available in Flash format from the *MODELPLEX* SVN [6].

8. Conclusion

Within the different sections of this document, the extended version of the *AM3* Global Model Management prototype including traceability support has been described in detail. Automated traceability support for M2M transformation with ATL and the integration of that support within the *AM3* tool were presented first. Further, the generic navigation support offered by the extended *AM3* version and directly applicable in the context of traceability was explained. Finally, a demonstration of the use of these two traceability-dedicated *AM3* features on a concrete SAP Model-Driven Performance Engineering (MDPE) use case was provided.

Such experiments with traceability, which have already been performed with the prototype, show interesting results in terms of the explicit representation of the required information, use of this metadata for navigation purposes and for the (semi-) automation of the overall process. Indeed, the fact of having all this traceability information explicitly represented as models (i.e. weaving models and a megamodel) makes much easier to retrieve, analyse or process it. Moreover, new parts of the extended prototype, such as the generic navigation support, are more widely applicable than solely within the specific context of M2M transformation traceability.

Thus, the follow-up of this Task 3.2 work, within the context of the *MODELPLEX* project in the coming months, concerns our current and future work on Global Model Management in Task 2.1:

- Continue the general improvement of the core of the *AM3* prototype and previously implemented extensions;
- Fix bugs progressively on the newly added traceability features in order to upgrade them;
- Develop our experimentation and applications of the prototype on other WP1 partner use cases.

9. References

[1] *MODELPLEX* Deliverable: D2.1.a “Global Model Management Principles” – Resubmitted to EC in month 19 (March 2008)
[2] *MODELPLEX* Deliverable: D2.1.b “Model Management Supporting Tool” v2.0 – Submitted to EC in month 26 (October 2008)
[3] *MODELPLEX* Deliverable: D3.2.b “Trace Establishment Support” – Submitted to EC in month 18 (February 2008)
[4] *MODELPLEX* Deliverable: D3.1.d “Model Transformation Technologies” – Submitted to EC in month 18 (February 2008)
[5] AM3 prototype (including traceability support) – MODELPLEX SVN: https://www.modelplex.org/svn/work_area/WP2/T2.1 Global Model Management/deliverables/plugins
[6] SAP Use Case traceability demo – MODELPLEX SVN: https://www.modelplex.org/svn/work_area/WP2/T2.1 Global Model Management/demos/Traceability Demo (D3.2.b) - December 2008
[7] Eclipse-GMT AM3 project: http://www.eclipse.org/gmt/am3/
[8] Eclipse-GMT AMW project: http://www.eclipse.org/gmt/amw/
[9] Eclipse-M2M ATL project: http://www.eclipse.org/m2m/atl/