

# Building a Bridge between Information Engineering and Model Driven Architecture

Xiaoxia Lin\*, Mustafa Arikan\*, Gerti Kappel‡

\*ARIKAN Productivity Group, Vienna, Austria  
{firstname.lastname}@arikan.at

‡Business Informatics Group (BIG)  
Institute of Software Technology and Interactive Systems  
Vienna University of Technology, Austria  
gerti@big.tuwien.ac.at

Dec. 2004

## Abstract

The Model Driven Architecture (MDA) approach of OMG aims to automatically derive software from models. Information Engineering (IE) is succeeding to do this with well established industry solutions for more than one decade. IE can be introduced into the MDA world using model transformation such that both of them can profit from each other in a cooperative way.

keywords: MDA, MOF, UML, Information Engineering, Eclipse

## 1 Problem Statement

### 1.1 MDA

A software system can be seen in terms of models and/or code. MDA is an approach to achieve the separation between software specification, i.e., models and software code in order to master the rising complexity of the software engineering (SE) world. This is expressed in the three layers of MDA, the so called *platform-independent model* (PIM), *platform-specific model* (PSM) and implementation. The PSM is a further refinement of a PIM in the direction of implementation. The process of converting one model to another model, i.e., from PIM to PSM is called model transformation [1]. MDA and related technologies, like for example QVT/MOF [2], aim for a series of specifications to enable automatic model transformation as far as possible. The separation of a software system into several models with different abstraction levels makes it possible for tools with the same or different technological background to cooperate with each other.

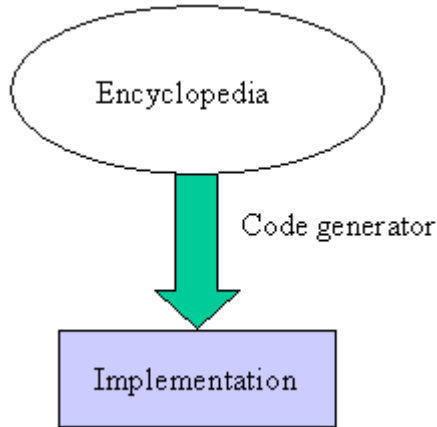


Figure 1: Information Engineering

## 1.2 Information Engineering

One of the aims of MDA is to raise the abstraction level of software development from coding to modeling. But this is not new at all. There are similar approaches which have been practiced since more than one decade ago. One of the examples is *Information Engineering* (IE). The tool AllFusion Gen<sup>1</sup> of Computer Associates is based on this concept.

IE was defined in [8] as the application of set of formal techniques for the planning, analysis, design, and construction of information systems on an enterprise-wide basis or even across a major set of the enterprises. Because an enterprise is usually very complex, automated techniques are required during planning, analysis, design, and construction. The specification of the system is stored in an encyclopedia, whereas system functionalities are expressed with various diagrams including pseudocode written in a fourth-generation language. This specification and the platform properties act as the input for a code generator to construct the actually deployable implementation. Figure 1 shows a simple view of this process.

Both IE and MDA declare models as the key player in the software development process where the implementation should be generated (transformed in terms of MDA) from one or many models. This separation of specification from implementation improves the quality and productivity of a software development process. IE and MDA serve the same goal, however with quite different approaches:

- IE is top down, whereas MDA is iterative.
- IE focus on data technology, i.e., ER-diagramming, whereas MDA is based on object technology, i.e., UML.
- IE concentrates on data-centric domains, whereas MDA tries to be a general framework for all kinds of domains.

---

<sup>1</sup>It is called as 'Gen' in the rest of the paper.

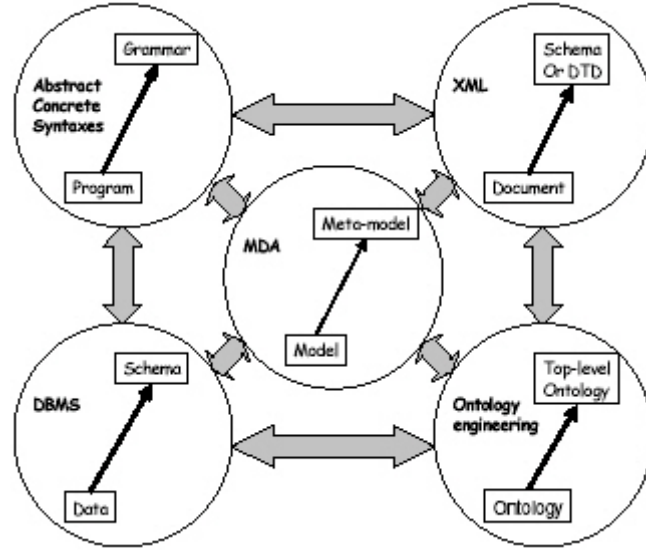


Figure 2: Technological spaces with some relations among them [7]

### 1.3 Missed Bridges

IE has proved to be a successful approach specially for data-centric application development for more than 10 years. However no other tool can reuse the design stored in an encyclopedia and the code generators of the IE tools can only translate the models in an encyclopedia of the particular tool. The connection between IE tools and other technologies is missing.

The concept of *Technological Space* (TS) was used in [7] to refer to a working context with a set of associated concepts, body of knowledge, tools, required skills and various other capabilities. It was mentioned in [7] that no TS is an island because the bridges can be built between the various TSs with model transformation. Figure 2 gives an overview of this idea with five example TSs. The crucial point of this figure is that MDA is providing a central switch station. Thus, for each of the TSs besides MDA only one bridge to MDA TS is needed instead of three. An open and cooperative SE world can be built in this way with MDA, where the TSs can communicate with each other and software engineers can combine tools or reuse models developed with other tools to construct their software in a new way.

Because of the wide acceptance of MDA, building a bridge to MDA TS is obviously the best choice for IE to connect to the other technologies. The *Meta-Object Facility* (MOF) [3] provides possibilities to define the IE-encyclopedia meta model in a standard language and the instances of the meta models can then be transformed into other TSs and vice versa. The challenge lies in the area of model transformation because there are still no well established foundations for it [10].

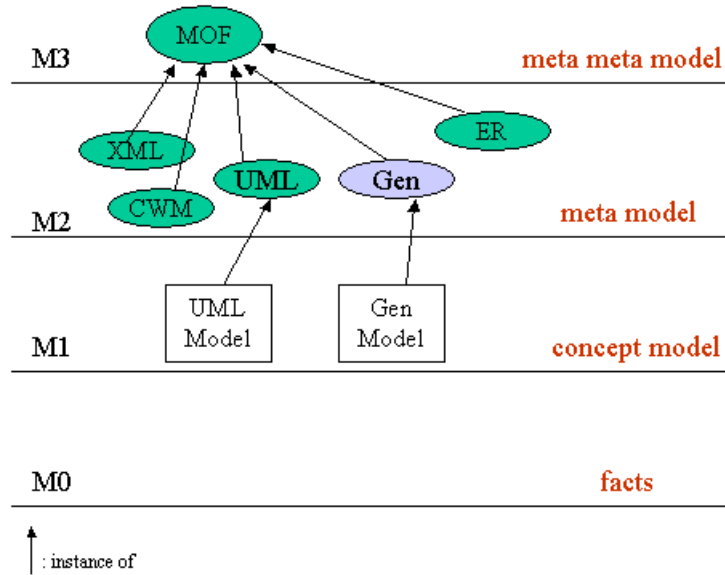


Figure 3: Gen model in the model hierarchy

## 2 Introducing IE into MDA World

### 2.1 Model Transformation as Kernel

Figure 3 illustrates the model hierarchy with examples in the layers. This infrastructure consists of four layers, each layer acts as the instance of the one above. A meta model of an IE Tool ('Gen' in this figure) is a description of the IE encyclopedia with a meta-meta model, i.e., MOF. Giving a meta-model written in MOF for Gen, all the instances of this meta model can be transformed to models defined with another modeling language being MOF-compliant. Figure 4 makes this clearer. A Gen model can be transformed into models of any MOF-compliant meta models. We will use UML models as case study. In order to introduce Gen into MDA world the Gen meta model in MOF must be defined at first. Then we can make Gen to communicate with other tools using model transformation approaches.

The model transformation approaches are categorized in [10] into model-to-model and model-to-code approaches at the top level. Transforming a Gen model into a UML model and vice versa needs model-to-model approaches. They are direct-manipulation approaches, relational approaches, graph-transformation-based approaches, structure-driven approaches, hybrid approaches and other approaches like using XSLT. The UMLX in category graph-transformation and *Atlas Transformation Language* (ATL) in hybrid are looked promising and both have Eclipse plug-ins. Framework technology can be used to combine or extend the existing approaches to achieve our goals.

Both meta model and model can be serialized as XML by using *XML Metadata Interchange* (XMI) [4]. Because XMI is the standard specification for model exchange, a tool has at first to be 'XMI-enabled', this means to have capability to import a model from an XMI document and export model as XMI-Document. Once the meta model and the model can be serialized as XMI, a model transformation tool can be fed with

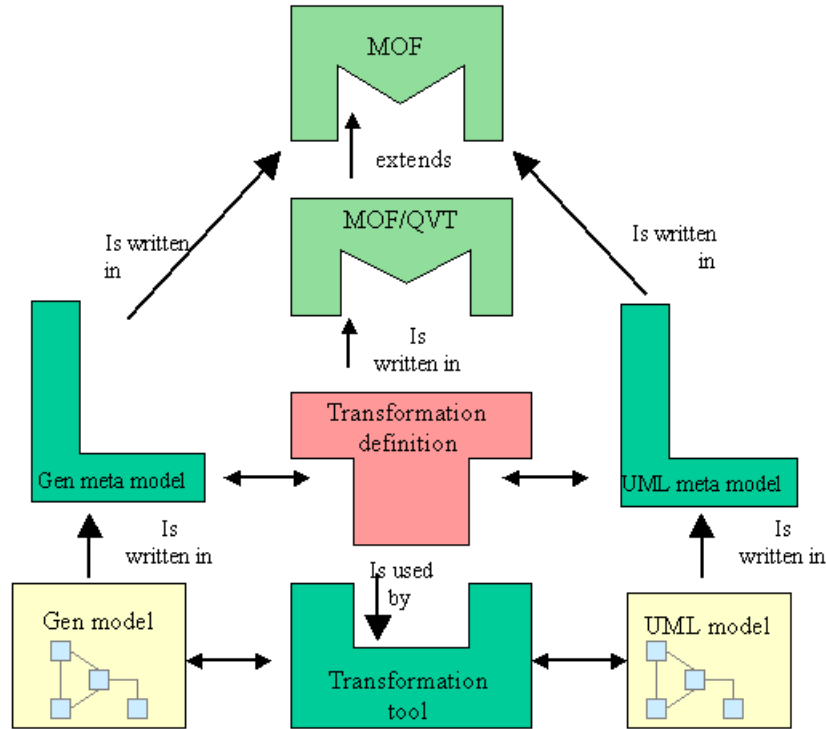


Figure 4: Transformation between an Gen Model and a UML Model (lightly adopted from [9])

them and perform the transformation.

We can call the Gen models as 'executable models' because they are concrete enough for a generator to produce the implementation. Also UML is in it's latest Version (UML 2.0 [5]) is called to be executable [6]. It is an open research question to compare the expressive power of the Gen models and UML models.

## 2.2 Eclipse as Platform

The Eclipse Platform is an Open source IDE built on discovering, integrating and running modules called plug-ins. A tool provider can write such plug-ins that operate in Eclipse with other plug-ins. The most significant enhancement in Eclipse version 3.0 must be *Rich Client Platform* (RCP). The RCP is the minimal set of plug-ins needed to build a rich client application. Such rich client applications are free to use any API deemed necessary for their feature set, and can require any plug-ins above the bare minimum [12]. This offers a possibility for easily developing a desk application with Eclipse technology and such applications can consume the suitable Eclipse-plug-ins. An IDE can also be built in such way if it is needed.

There are a series of open source projects running in Eclipse community about meta modeling, model transformation and code generation. Each of these projects consists of many plug-ins that can be used from other plug-ins. Eclipse is a prime technology for building a tool for model transformation, not only because of the interesting function-

alities of these plug-ins, but also the flexible nature of Eclipse's plug-ins architecture, which meets the 'design for change' principle. In addition, the RCP technology of Eclipse allows the developer to select suitable portion of the plug-ins to build a tool.

## 2.3 A Possible Tool Development Scenario

The maturity of a tool for introducing Gen into the MDA world can be divided into three levels: XMI enabling, model transformation and cooperation with other MDA tools. One possible tool development scenario based on Eclipse technology can look like this:

1. Defining the Gen encyclopedia in MOF and Developing an RCP-Client as Gen plug-in for importing and exporting XMI documents.
2. Integrating a model transformation tool in the Gen plug-in to enable the conversion between Gen models and models defined in other modeling languages.
3. Because of the UML dominance we can begin with the integration or development of an UML editor in the Gen plug-in. The Gen model should be kept synchronized with the UML model.
4. The integration of further model transformation tools to enhance the usage of converted Gen models, i.e., to generate rich client applications in java.

## 3 Conclusion

IE and MDA are very different approaches but overlap with each other by making effort to shift the developer's work from coding to modeling. The tool AllFusion Gen from Computer Associates Inc. has been succeeding in application development by using IE principles since the 90's. Introducing Gen into MDA world can be realized with model transformation with the tools built on an Eclipse platform. Model transformation is the heart and soul of this process and making Gen 'XMI-enabled' should be the very first step. This is a challenge including a lot of research work. Prototyping is a good choice to begin with.

## References

- [1] OMG: *MDA Guide*; <http://www.omg.org/docs/omg/03-06-01.pdf>
- [2] OMG: *OMG/RFP/QVT/MOF 2.0 Query/Views/Transformations RFP*; <http://www.omg.org/docs/ad/02-04-10.pdf>, 2002
- [3] OMG: *Meta Object Facility Specification 1.4*; <http://www.omg.org/docs/formal/02-04-03.pdf>, 2002

- [4] OMG: *XML Metadata Interchange(XMI) Specification 2.0*;  
<http://www.omg.org/docs/formal/03-05-02.pdf>, 2003
- [5] J. Rumbaugh, I. Jacobson, G. Booch: *The Unified Modeling Language Reference Manual, 2nd Edition* ; Addison-Wesley, 2004
- [6] Chris Raistrick, Paul Francis, John Wright, Colin Carter, Ian Wilkie: *Model Driven Architecture with Executable UML* ; Cambridge University Press, 2004
- [7] Jean Bezivin, Gregoire Dupe, Frederic Jouault, Gilles Pitette, Jamal Eddine Rougui: *First experiments with the ATL model transformation language: Transformation XSTL into XQuery* ; OOPSLA 2003 Workshop on Generative Techniques in the context of Model-Driven Architecture
- [8] James Martin: *Information Engineering, Book I: Introduction*; NJ: Prentice-Hall, Inc., 1989
- [9] Jos Warmer: *The role of OCL in the model driven architecture*; International Symposia on Formal Methods for Components and Objects(FMCO) 2002
- [10] Krzysztof Czarnecki, Simon Helsen: *Classification of Model Transformation Approaches*; OOPSLA 2003 Workshop on Generative Techniques in the context of Model-Driven Architecture
- [11] Jim des Rivieres: *Eclipse Project Slide Presentation*;  
<http://www.eclipse.org/eclipse/index.html>
- [12] eclipse.org: *Eclipse Rich Client Platform* ;  
<http://www.eclipse.org/articles/index.html>