From Code to Models: Past, Present and Future of MDE Adoption in Telefónica

Miguel A. Fernandez

Telefónica Research & Development
Valladolid, Spain
mafg@tid.es

Abstract. In this paper we present the experience of Telefónica in moving from a code-centric development process to a model-centric one. We first explain early failures and their main causes. Later we discuss some achievements in the field of Domain-Specific Languages and finally we convey our present and future plans for moving towards a purely model-centric approach and the challenges foreseen in this road, which can mainly be classified in three categories: organizational barriers, adequate reutilization of the newly created assets as well as the existing ones (legacy systems) and proper training of the technical staff. We also raise some concerns about the maturity of the state-of-the-art in this field and about the limitations of general purpose modelling languages for code generation, while expressing our hope that significant successes can be achieved in the field of Domain-Specific Modelling.

Keywords: DSM, DSL, MDE, UML, full code generation, telecom service modelling, network configuration.

1 Introduction

Traditionally, the telecommunications industry presented huge barriers to competition. There were large state-owned companies, well established in their respective countries, developing their business strategies under the protection of the monopoly situation. However, the situation is changing as of late, with those barriers gradually being removed, bringing us to a scenario where many new agents are beginning operations in a more open and competitive environment.

It is now clear that these well-established companies have to redefine their objectives and strategies to better face the future, in a way that lays the foundations of change to next-generation networks and value-added services. This must be perceived not only as positive, but also necessary for their survival. For this purpose it seems advisable, as a first step in defining these future strategies, to analyse the change itself, beginning with the factors that have unleashed that change, namely the regulatory framework, the emerging technologies, the Internet and social changes.

The change in monopoly legislation has brought a new reality, very different from the previous one, where business development is a key factor to the success or failure of telecom operators. The main effort has been to move on from the network itself and its operation towards new business opportunities and customer care. This trend leads us to identify some aspects relevant to the new scenario.

Firstly, business management must be much closer to the customer, whose satisfaction becomes a very important concern. There is an increasing need to understand customers' needs and to adapt as quickly as possible to their preferences.

Secondly, the value chain is widening, giving more and more relevance to content provision, new service creation and fast service deployment. This means a wider service portfolio, as well as improved Time-To-Market and Time-To-Connect.

Finally, the balance sheet also becomes a key element, so adequate management of the available human and financial resources must be achieved.

Business management must pay close attention to new technologies. The operators that lead the adoption of the best solutions will be the leaders of the market. But considering the attractiveness of technology alone, not taking into account the social changes and the pace of the market can be dangerous and lead to failure. For that reason the solutions must be affordable and scalable. To achieve these goals the operators need to have at their disposal very powerful tools to optimise all the steps in the provisioning chain.

One of the most critical steps in that chain, and the subject matter of this article, is service creation and deployment. The efficient development of robust and complex services as well as their fast and seamless deployment on Telefónica's network has been the aim of the company's R&D effort in the last decade.

Of the many relevant examples of this effort, a truly significant one is discussed in detail herein. Our case study reviews the past, present and future of Telefónica's research in the field of assisted and automated equipment configuration and the software development paradigms applied, with different levels of success, to the creation of such configuration tools and systems.

2 First Failed Attempts at Modelling

Back in 1999 a new broadband access technology revolutionized the Spanish Internet market. The Asymmetric Digital Subscriber Line (ADSL) brought for the first time a high bandwidth connection at a relatively low cost and set off an explosion in the number of Internet users that irreversibly changed the market.

In early 2000 it was evident that new management and configuration tools were needed to cope with the rapidly growing customer base and the associated home networks and devices. In particular, the development of tools that allowed easy ADSL modem-router configuration for use by operations personnel, as well as end users, became a necessity.

Also the number of equipment vendors that entered the new ADSL market started to grow and soon the heterogeneity of configuration protocols and Command Line Interfaces (CLI) of the different ADSL modem-routers raised the question of how to come up with a simple and unified user interface that could hide the underlying complexity to the non-tech-savvy customer.

The development department in charge of residential Internet access management and configuration tools at Telefónica started to build a framework that could support all that heterogeneity and, as with any new project, several approaches were tested before actual implementation began.

With the release of UML 1.1 [1] in 1997, some development teams had started to use this new modelling language as a way to capture information about requirements and system architecture in the initial stages of analysis. In this project that approach was also tried but with the more ambitious aim of creating much more detailed and complete blueprints of the system and use those as a foundation for guiding the implementation process and creating detailed technical documentation that would make the evolution and maintenance of the system much easier in the future.

The main focus of the modelling effort was put into the specification of the interaction between the configuration tool and the network equipment (ADSL modem-routers, in this case). This configuration protocol modelling was seen as key to the success of the approach, as it was the core functionality of the system. For that purpose, multiple interaction use cases were detailed by means of sequence diagrams, but soon the team realized that complete specification of the protocols was not feasible in that manner, as the number of possible scenarios was almost infinite.

Statechart diagrams were then tried but a full protocol modelling was not achieved this time either. The cause can be found in the combination of several circumstances, where the lack of experience with UML undoubtedly played a role, but the notational complexities of this modelling language, as well as the inability to maintain the models in synch with the code without a great effort, are also reasons worth mentioning. Tools for code generation from UML models were not state-of-the-art by then; hence the modelling effort was seen as wasted if going beyond a first high-level Platform Independent Model (PIM). Also, on later refinement iterations, code was directly updated without any further modelling, and in the absence of roundtrip engineering support, that led to the out-of-synch problem.

Finally, the development process went on without any significant or measurable benefits obtained from the modelling effort, and even if the best practices in object oriented analysis and design were applied and the final results were quite satisfactory, the whole model-based experience was deemed as unsuccessful.

3 A Past Success Based on a DSL Approach

Towards the end of 2000 the first framework for building equipment configuration systems was ready. It had been constructed with a focus on component reusability and, as a result, it produced some C++ libraries and other assets that remain in use today.

However, even with the help of those high-level components and libraries, the task of creating the complex configuration systems that the unstoppable growth of the market demanded was not suited for everyone. Highly skilled engineers that could cope with all the variables involved in the process were scarce and costly. On the other hand, whenever systems' development was carried out by less-than-well-trained staff, the increasing number of software defects led to additional testing iterations and failure to meet the deployment deadlines, which in turn was also costly not only in terms of loss of business, but also in lack of credibility.

Due to the shortcomings found with UML 1.x (the OMG did not release UML2 until 2004) and the relatively recent failure to build a framework based on models, in early 2003 the team decided to try a different approach to solve the problem. A selected group of technical experts with extensive experience in the domain were put in charge of finding a new development paradigm that was at the same time robust and not as error prone for less skilled developers.

After long consideration, the idea that prevailed was one of building on top of existing assets and creating a programming tool that was specifically designed to address the problem domain, that is, network device configuration.

The main requirements on this tool were: a) adequate management of metadata, b) simple capabilities for flow control and decision-making, c) full support for all the protocols used in device configuration, and d) easily extensible to add support for new protocols (e.g.: MDAP, which is a multicast protocol used by Thomson equipment).

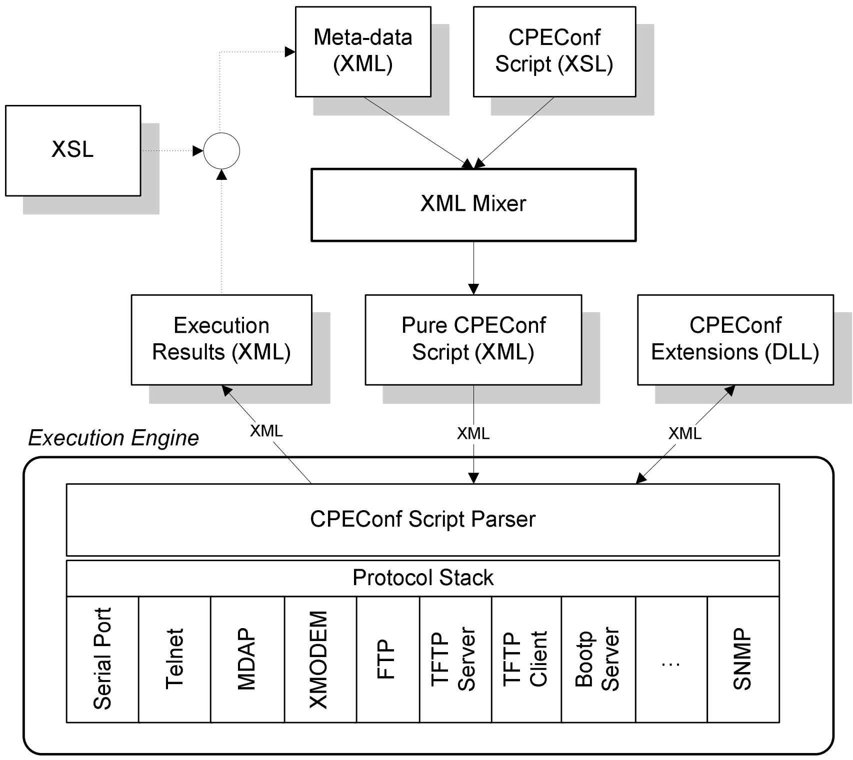

Fig. 1. High-level architecture of the DSL-based solution for network equipment configuration developed in Telefónica.

Instead of creating a fully functional language from scratch, an intermediate approach was adopted. A first stage would get the metadata in XML form (the specific configuration and scenario description data, such as IP address and network mask or DHCP parameters) and, relying on XSL [2] and XSLT [3], a configuration script would be generated. With this approach, the language did not have to support complex constructs and the implementation was simpler and focused on configuration flow control and specific support for protocols and other domain concepts. The syntax of the new language was also XML based and it was quickly nicknamed CPEConf (Customer Premises Equipment Configuration).

To bridge the gap between this newly created Domain-Specific Language and the supporting framework and platform, an interpreter engine was developed. Due to the fact that the first step was accomplished by an XSL transformation, this was the most sensible option. Compared with an approach based on code generators and compiled code, it also had the advantage that new and updated configuration scripts could be easily deployed and run directly without further processing. Figure 1 shows the main components of this framework.

A simplified script fragment for configuring a Cisco router via XMODEM protocol might look like the following example:

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
<xsl:output method="xml" doctype-system="cpeconf.dtd" />
[…]

<!-- Open serial port COM1 at 9600 baud, 8 data bits,
**   1 stop bit, no parity and no control handshake
-->
<sequence id="{$start}">
  <command>OPEN_SERIAL "COM1" 9600 8 "1" "NO" "NO"</command>
  <on_fail><xsl:value-of select="$error"/></on_fail>
  <on_timeout><xsl:value-of select="$timeout"/></on_timeout>
</sequence>
[…]

<!-- Send the xmodem command to the router console
-->
<sequence id="{$start_xmodem}">
  <command>SEND_COMMAND "copy xmodem: startup-config"</command>
  <command>SLEEP 2000</command>
  <response_ok>
    <goto="$open_xmodem">Ready to copy...</goto>
  </response_ok>
  <on_fail><xsl:value-of select="$closex_modem"/></on_fail>
</sequence>

<!-- Initiate the XModem session
**   and then transfer the configuration file to the router
-->
<sequence id="{$open_xmodem}">
  <command>OPEN_XMODEM "COM1" 9600 8 "1" "NO" "NO"</command>
  <command>PUT_FILE_XMODEM "configuration.txt"</command>
  <msg_on_error>Error transmitting config file.</msg_on_error>
  <on_fail><xsl:value-of select="$closex_modem"/></on_fail>
  <on_timeout><xsl:value-of select="$retry_xmodem"/></on_timeout>
</sequence>
[…]
```

The advantages of this new programming language started to show clearly with the early versions. Engineers with some technical knowledge of routers and basic notions of XSL transformation were able to write configuration scripts quickly without the need for any previous C++ or other object-oriented language expertise.

This tool also forced some changes in the organization. Instead of having several teams devoted to the same kind of development projects, a special permanent team had to be created with the responsibility of maintaining and evolving the tools that the rest of the groups were using. These organizational changes and their implications will be discussed in further detail in Section 5.

This language has been extensively used and evolved in our department since its creation and is still in use today. In these five years of use, the amount of code written for it is, by large, well in the range of the hundreds of thousands of lines. Given the synthetic properties of domain-specific languages, this can be estimated as several millions of lines of C++ code that would have been produced otherwise.

The following table summarizes some estimated metrics regarding the productivity improvement and cost/effort savings that the tool has produced during that period.

Table 1. Productivity improvement and cost/effort savings in the department of equipment configuration tools at Telefónica by using a DSL approach. The data are based on a typical mid-complexity service.

	With DSL	Without DSL	Comparison
Lines of code (configuration code only)	2.000	30.000	15 times less
Critical bugs (configuration related)	Less than 5	10 to 20	2 - 4 times less
Effort (in person-months, whole project)	15	25	down to 60%
Development cost (in 2008 K€, whole project)	110	230	down to 50%
Time-To-Market (in months)	3.5	5	down to 70%
Total lines of code produced (2003-2008)	250.000	≈ 3.75 Mloc	--

As this data clearly attests, the incursion into the world of DSLs has proved highly beneficial to the company in terms of product quality and Time-To-Market (TTM) improvement, while generating significant cost and effort savings, but the approach is not completely free of some weak aspects, the main one being the fact that CPEConf scripts are vendor specific and, in some cases, even device specific and thus, it is necessary to create separate scripts for each network device capable of providing a specific service or combination of services, which, with the current explosion in the number of services and devices that need to be supported, is becoming an increasingly important concern. Also, even if the CPEConf language is adequate for technical staff it is not easy to understand for domain experts.

4 Current Work on MDE and Domain-Specific Modelling

Several years have passed and, as we pointed out at the end of the previous section, the services that customers demand from our company are becoming more and more complex, while the pressure to deploy these new services faster keeps increasing.

To address these issues, the company decided to participate in the MODELPLEX project (IST-FP6-2006 Contract No. 34081), co-funded by the European Commission as part of the 6th Framework Programme. Since the end of 2006 and in the context of this project, a team of researchers has been assigned the task of finding a new paradigm for telecom service development, with a focus in the following aspects: a) progressive migration to a remote “zero touch” configuration approach, b) industrial support to the adoption of new device-independent configuration standards, c) more streamlined service development processes by creating tools usable in all the phases of the lifecycle (as opposed to only in the implementation phase) and by all the roles involved in the process (as opposed to only by technical experts) and finally, d) reuse and build, again, on top of existing assets.

With those objectives in mind, the company composed an industrial case study centred on the concept of a CPE Management and Configuration System (CMCS), which basically is a server for automated and remote equipment configuration. Telefónica expects that a system like this will help the company to greatly improve the Time-To-Connect (TTC) of our services, that is, the time spent since a user makes a subscription request until the service is fully operative. By reducing human intervention in the configuration process to a minimum, the system will also help reduce the Operating Expenses (OPEX).

To achieve the adoption of device and vendor-independent configuration protocols, the company also started a strategy of full support and commitment to the DSL Forum's TR-069 family of standards [4] (please note that, in this case, DSL stands for Digital Subscriber Line). In the context of this strategy, Telefónica is putting a lot of pressure on network equipment vendors to adhere to these standards in their new product releases.

In the fulfilment of the last two objectives is where the MDE paradigm in general and the tools developed in MODELPLEX in particular are taking a predominant role. The core of our work in the project consists of a DSM graphical tool that will permit the modelling of the basic concepts in the telecommunication domain, such as interfaces, devices, networks, protocols and services.

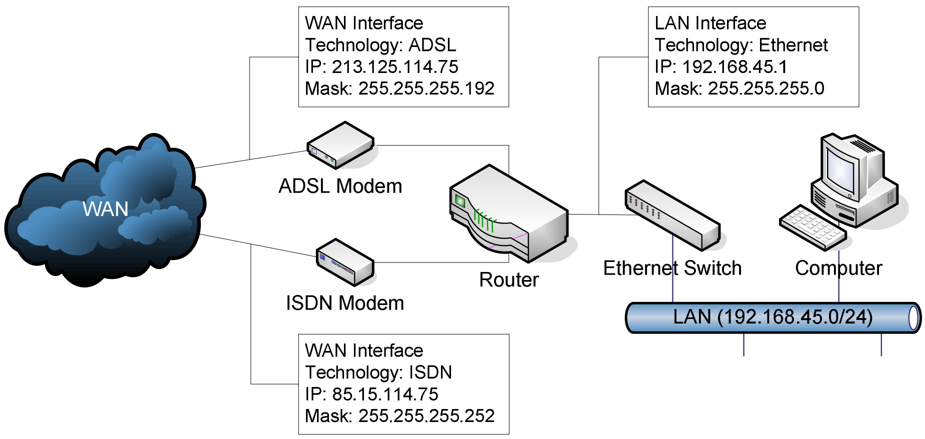

Fig. 2. This example of a DSM depicts a simple ADSL service with ISDN backup line. The level of abstraction is notably higher than in a direct CIM model. The annotations show some of the domain-specific properties of the model.

After an intense state-of-the-art analysis, the Common Information Model (CIM) from the Distributed Management Task Force (DMTF) [5] was chosen as the meta-model for the DSM tool. This choice provides strong foundation for the tool, building on top a solid industry standard. A first prototype of the tool has already been developed with the assistance of MODELPLEX partner, and expert in DSM tooling, Xactium Ltd [6]. This first implementation supports a direct mapping from DSM elements to CIM concepts. Current and future work is focused on higher abstraction layers (see Figure 2) and on integration of a model verification language that could be either the Object Constraint Language (OCL) [7] from the OMG or the Epsilon Validation Language (EVL), part of the Epsilon family of development tools [8] from (also MODELPLEX partner) University of York.

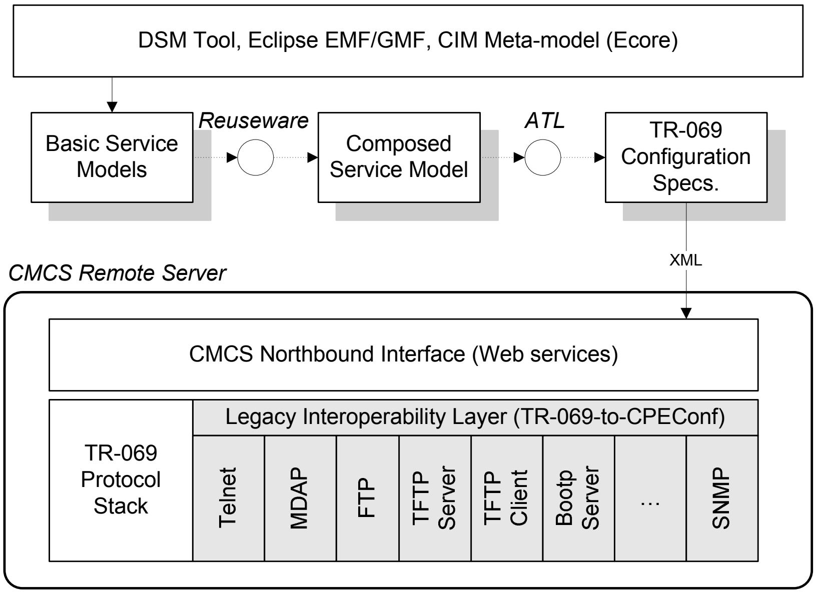

Fig. 3. General architecture of the DSM-based tooling. The support for legacy, CPEConf asset reutilization is shown in gray, as it is currently out of the scope of the MODELPLEX project.

From this modelling framework we foresee two benefits, compared with current service development state-of-the-art in the company. Firstly the creation of a tool that can substitute service requirement specification by business and domain experts, currently done via documentation, by a model-based specification, thus optimizing the communication between the different roles in the chain. Using the same tool, more technically-oriented experts can further refine the specifications, finally obtaining a complete model of the service. Secondly the fully-automated generation of device configuration scripts by transformation of the original model. For this purpose we are developing such transformations with the help of other partners' tools like Reuseware Composition Framework [9] from Technische Universität Dresden for model (service) composition, ATLAS Transformation Language (ATL) [10] from INRIA for model-to-model transformation (both for TR-069 and also, in the future, CPEConf script generation) and MOFScript [11] model-to-text transformation language from SINTEF for generation of documents and other artefacts like system configuration files.

All this work will take us as close as possible to a DSM-based full code generation approach (see [12]), while entering the realm of automated remote configuration. It also puts us on the road to integrating existing DSL assets (the CPEConf Language). It is important to note, however, that this integration is out of the scope of the MODELPLEX project. The resulting framework can be seen in Figure 3.

As this is a work in progress we cannot yet provide concrete measurements of the benefits that MDE will bring to our business but further improving TTM and reducing development costs are foreseen to a certain extent.

5 Challenges on the Road to MDE Adoption

As we have mentioned in the previous section, when faced with changing the paradigm for software development in a company, there usually exists a clear need for reusing and integrating already available assets into the new paradigm. As others have previously pointed out [13], success in the adoption must be based on evolution instead of revolution.

The use of tools for reverse engineering and model understanding can mitigate the impact of the move to MDE by helping the company to migrate those assets and raise the level of abstraction, which may prove highly valuable to feed the newly obtained models into the MDE chain with the intention of redesigning them or regenerating the code for a different platform.

Of course, even with the help of reverse engineering, the complexity and size of such a task must not be taken lightly; hence the possibility of just building an MDE chain on top of those assets seems to permit bridging the gap and start obtaining benefits quickly. This reasoning led the research team at Telefónica to the conclusion that a DSM solution as the one explained in Section 4 could be successful.

There are, however, additional factors to consider that can compromise the success of MDE adoption, adequate training of the technical staff not least among these. The MDE paradigm brings with it so many new tools and technologies that to produce or hire well-trained professionals becomes an important concern and can lead to higher costs that can even exceed the benefits in productivity improvement. To address this problem, the solution is again (see our previous experience about this in Section 3) the creation of special teams, composed of experts in those technologies from everywhere in the organization, that can lead innovation and creation of tools and frameworks that other less-skilled engineers can then use to generate code and build reliable systems. This strengthens our opinion that DSM approaches can be greatly beneficial.

But the creation of such teams at a large scale can also prove difficult because organizational barriers that some companies may have. The hierarchical structure of our company, divided in departments at several aggregation levels, which are mainly organized around specific areas in the domain, makes it complex to share expertise in crosscutting or horizontal technologies. This is not usually a problem when those technologies are simpler or widespread, like Java programming language or Service Oriented Architecture (SOA) but, as technologies become more and more diverse and less spread it can be difficult, if not impossible, to have all types of experts available in all departments. Our current experience is that finding experts in the different MDE technologies, like model transformation, model-based testing (MBT) or simulation, to name just a few, is a challenging task even for a big company.

Furthermore, we have to take into consideration the problems and risks associated with the adoption of technologies that are not fully mature yet, which is the case in our modest opinion with MDE. There are some challenges still present in the field, like real support for traceability and roundtrip engineering. Also the multiplicity of similar tools being developed and the lack of really integrated tool suites can have a negative impact in the widespread adoption of the paradigm. The convergence of tools around the Eclipse framework is certainly promising, but for big companies to commit to a technology, soundness is not the only concern; a clear support behind the tooling is also essential.

Finally, it is our belief that commitment to MDE, like with any other technology, must be carefully assessed by a company before it starts investing on it, because what works for someone might not necessarily work for others. In our case, for example, we have presented in detail our expectations about Domain-Specific Modelling solutions, but at the same time we have important concerns regarding the limitations of general purpose modelling languages, as UML, for code generation. There is no doubt, however, that for clearly defined scopes, this approach can and should be successfully tried.

6 Conclusions

From our experience of this last decade in Telefónica we can conclude that, as society and technologies evolve, businesses have to move faster and faster to cope with that change. In a domain like ours, that of telecommunication service provisioning, this trend is even clearer.

Quick reaction to the market implies the modernization and evolution of key systems and business processes, but this modernization has to be cost effective and thus, reutilization of available assets has to be taken into account.

Can the MDE paradigm help a company like Telefónica address these changes in a more effective manner? We have little doubt that the answer to this question is affirmative, but we realize as well that, to be successful in the transition, more changes are necessary that must be conveniently analyzed and weighted.

Acknowledgments. Part of the work presented in this paper is being carried out in the context of the MODELPLEX project (IST-FP6-2006 Contract No. 34081), co-funded by the European Commission as part of the 6th Framework Programme. The opinions expressed, however, are those of the authors only, and do not necessarily correspond to those of other members of the consortium.

We would also like to thank all the MODELPLEX consortium partners for their collaboration in our research.

References

1. Unified Modeling Language (UML) Proposal to the OMG, Version 1.1, 1st September 1997, http://www.omg.org/cgi-bin/doc?ad/97-08-11
2. W3C Extensible Stylesheet Language (XSL) Recommendation, Version 1.0, 15th October 2001, http://www.w3.org/TR/2001/REC-xsl-20011015/
3. W3C XSL Transformations (XSLT) Recommendation, Version 1.0, 16th November 1999, http://www.w3.org/TR/1999/REC-xslt-19991116
4. DSL Forum Website, Technical Reports, http://www.dslforum.org/trlist/trlist.php
5. DMTF's Common Information Model Website, http://www.dmtf.org/standards/cim/
6. Xactium Ltd. website, http://www.xactium.com/
7. Object Constraint Language (OCL) Specification from the OMG, Version 2.0, http://www.omg.org/cgi-bin/doc?formal/2006-05-01
8. Kolovos, D., Paige, R., Polack, F.: Epsilon Development Tools for Eclipse. In: Eclipse Modeling Symposium, Eclipse Summit Europe 2006, Esslingen, Germany, 2006.
9. Reuseware Composition Framework Website, http://www.reuseware.org/
10. Jouault F., Kurtev I.: Transforming Models with the ATL. In: Proceedings of the Model Transformations in Practice Workshop at MoDELS 2005, volume 3844 of LNCS, pages 128–138, Montego Bay, Jamaica, October 2005.
11. MOFScript Official Website: http://www.modelbased.net/mofscript/
12. Kelly, S., Tolvanen, J.: Domain-Specific Modeling: Enabling Full Code Generation. John Wiley & Sons, Inc., New Jersey (2008)
13. Vogel, R., Mantell, K.: MDD Process for an SME: Evolution, not Revolution – Phase I. In: Proceedings of C2M Workshop at ECMDA-FA 2005, Nuremberg, Germany.