

MDAI: Model based Design in Automobile Industry

João Carlos Amaro Ferreira

Abstract—It is proposed a new approach based on a methodology, assisted by a tool, to create new products in the automobile industry based on previous defined processes and experiences inspired on a set of best practices or principles: it is based on high-level models or specifications; it is component-based architecture centric; it is based on generative programming techniques. This approach follows in essence the MDA (Model Driven Architecture) philosophy with some specific characteristics. We propose a repository that keeps related information, such as models, applications, design information, generated artifacts and even information concerning the development process itself (e.g., generation steps, tests and integration milestones). Generically, this methodology receives the users' requirements to a new product (e.g., functional, non-functional, product specification) as its main inputs and produces a set of artifacts (e.g., design parts, process validation output) as its main output, that will be integrated in the engineer design tool (e.g. CAD system) facilitating the work.

I. INTRODUCTION

This paper is a review of research design and development of new products in automobile industry based on models that we call MDAI. New product development is a transformation of a market opportunity and a set of assumptions about product technology into a product available for production and later to sale. Our proposal is to use past experience and promote knowledge reuse based on the models that assist this complex development process. This approach is intended primarily for two audiences. First, we hope to benefit from young designers or product developers as well as to speed up the complete development process.

Designing a automobile is a complex task as it requires close coordination and inputs from a number of disciplines in order to develop the different systems and sub-systems in the vehicle. Furthermore, the numerous systems in a automobile must fit within the confined automobile space, perform the function and provide the customers an acceptable combination of all relevant elements of the vehicle. The objective of this research is to model the process business using UML (Unified Modeling Language) [1] and develop a design tool that can be used on the early stages of the design of a new product development. UML will play two major rules: (1) describing process business activity; (2) MDA (Model Driven Approach) [2] oriented for the automatic software development and consequently for

the automatization of repetitive jobs. Thus, the research focuses on the development of a conceptual framework and software development to perform design and analyse tasks in the following areas: (1) Plan of vehicle's roominess (study of Occupant and Mechanical Packaging, Ergonomics, Human Factor); (2) Packaging definition containing technical, ergonomic and legal bonds; (3) Check the model parameters and the possibility to compare it with competitors' vehicles by using an Excel program and benchmarking & teardown databases integrations on CAD (Computer Aided Design) systems; (4) Comparison with competitors' vehicles by using CAD maths and by loading several vehicles profiles from existing databases, like "Autograph". The first requirement in automobile design is to ensure that occupants are placed safely and comfortably. This process of "Ergonomic vehicle packaging" consists of several competing activities. In the preliminary stage, most attention is focused on the manikin in its seated position. A basic package is achieved by establishing front and rear seat geometry, front and rear seat entry and egress as well as seat environment (visibility, reach). The automotive occupant packaging process relies on a "human factors" database that defines spatial locations for the driver's eye and head, hand reach, preferred seat positions and other related dimensions. Many of the studies involved hundreds of human subjects and have resulted in standardized drafting templates and CAD procedures that are used to develop seating packages for combined populations of male and female vehicle users. The necessity for speed in the preparation of these renderings prohibits a full package study (all components). Even with the help of a CAD system, producing a full package requires days and even weeks of intensive work and has to be repeated many times during the development of a vehicle. At each stage of this process, each component must be carefully checked. If a given component originates changes in the vehicle interior dimensions, all other components have to be checked. A typical component study consists of 10 to 20 variables and about 10 relationships. Certain variables are common to all studies, creating complex dependencies among activities. Each change to a basic dimension needs to be reflect on all the other components. The traditional CAD approach does not support iteration very well. When a change to a package occurs late in the process, the costs of repackaging are very high considering that each individual study has to be verified. In practice, such an alternative is not even considered. Shortcuts are often taken in order to expedite the process. The need to improve this process is the key for producing vehicle packages quickly and accurately. Any improvement in this process has to provide a way of capturing each activity's goals, variables and relationships in an active way.

Manuscript received February 1, 2009.

João Ferreira is with the GuIAA-ISEL, Lisbon Portugal. Rua Conselheiro Emídio Navarro 1 1900-049 Lisboa; e-mail: jferreira@deetc.isel.ipl.pt.

The model, developed as software integration (Visual Basic or C++) on a parametric CAD system, has new researchers entering the field of product development. We also hope this review will be valuable to experienced researchers who are interested in learning about the range of research in product development, perhaps to identify new research opportunities or to locate issues that intersect their current interests.

II. RESEARCH CHALLENGES

The main benefits of the model based approach are: (1) reduce time-to-market; (2) reduce cost of development and maintenance; (3) improve quality or productivity, or any other observed benefits.

We searched the following publication channels for industrial studies related to Model based application, from books [3, 4, 5, 6], to the reports of the Software Engineering Institute (SEI) [7] and the Fraunhofer Institute for Experimental Software Engineering (IESE) [8], and to papers in various technical journals and conferences, such as: (1) ACM digital library; (2) IEEE Xplore; (3) Empirical Software Engineering Journal; (4) Journal of Systems and Software; (5) Journal of Information Science; (6) IEEE Transactions of Software Engineering (TSE); (7) IT Professional; (8) ACM Computing Surveys (CSUR); (9) Journal of Research and Practice in Information Technology; (10) IEEE International Conference on Software Maintenance (ICSM); (11) International Software Product Line Conference; (12) International Conference on Software Engineering (ICSE); (13) International Conference on COTS-Based Software Systems (ICCBSS).

We identified 40 papers and reports that claim industrial application of a model based approach. Most of these papers are single projects with description of a project or development method and discussion of experiences [9-15, 17, 18-19, 21, 26-30, 33-35, 36]. Of these, only two include some quantitative data from the projects (both from Motorola). Only seven papers report experiences from completed projects [9, 11, 16, 21, 31, 33, 35], while the others are from pilot studies or ongoing projects at the time of reporting.

Main motivations for applying Model based approach are:

- Increasing productivity and shortening of development time: as in [12, 14, 16, 25, 26, 29];
- Improve quality: improving the quality of the generated code [31, 33, 35], improving the quality (assurance) of system requirements [13] and managing requirement volatility [28], improving the quality of intermediate models [21], and earlier detection of bugs [20, 33, 35];
- Automation: generating code and other artifacts and introducing automation into the development process [9-12, 15-17, 19-21, 23, 27, 29, 34], and model-based simulation and testing [19];
- Standardization and formalism: providing a common framework for software development across the company and phases of the lifecycle [10, 30, 31], formalize and organize software engineering knowledge

at a higher level of abstraction [35], and common data exchange format [26];

- Maintenance and evolution concerns: maintaining the architecture intact from analysis to implementation [31], evolution of legacy systems [20], concerns over software method and tool obsolescence [14], verification of system by producing models from traces [34] and that PIMs have long lifespan [22];
- Improved communication and information sharing: between stakeholders [25, 30] and within the development team [20, 32-33] and ease of learning [33, 35].

Additional motivations are traceability throughout software development artifacts [25, 32], early assessment [28, 32], promoting reuse [10, 25, 30, 35], porting of solutions to new platforms [20, 21], and the ability to estimate costs based on the models [28, 32].

The main motivation of model based approach is the increasing of productivity (and shortening development time) and improving quality may be regarded as the ultimate reasons for applying model based approaches.

By using transformations model based approach is given emphasis to generating models, code and other artifacts from models, in addition to the verification and validation on the model level.

The current state of model based approach is far from mature. There is a varying degree of automation and it is mostly applied for code generation. Tools have been improved during the recent years but several papers still discuss the lack of a coherent model based approach environment and tool chain. Tools should scale to large-scale development and support the domain-specific approach more effectively. Other challenges in adopting model based approach are the complexity of modeling itself, developing PIMs that are portable to several platforms and using model based approach together with legacy systems.

III. DOMAIN SPECIFIC LANGUAGE: VDML

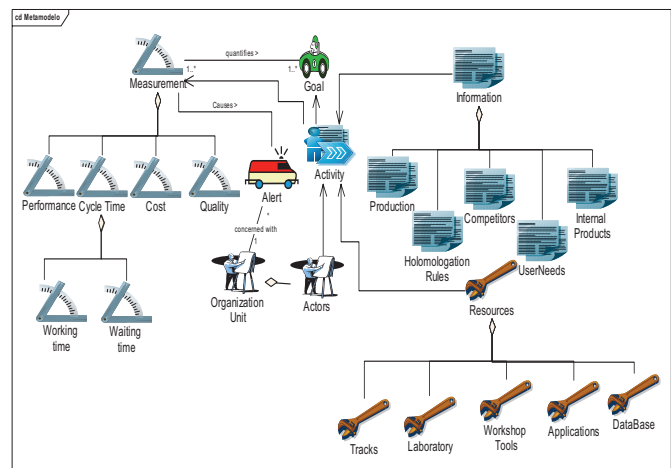


Figure 1: VDML-UML Profile for New Vehicle Process Development.

Based on UML we propose a meta-model for Vehicle Development Model Language (VDML), based on a new

stereotype specific for design business [37]. **VDML** [37] is a set of coherent UML extensions, Figure 1 shows a proposal, that allow a high-level, visual modeling way to design interactive systems. The main idea is to have a standard language UML which could act as: (1) standardization and systematization of concepts and notation in the new product development; (2) establishment of rules and guides for code generation through appropriate templates, which transform XML system models in written code expressed in a predefined chosen language. This templates are constructed based on the knowledge of the final code on an inverse engineering process; (3) automation of job routine process; (4) central database could store models and templates; (5) common interface between modules; (6) store knowledge in the model.

IV. MDAI PROPOSAL

Figure 2, describes the main principles of MDAI proposal. Basic and specific VDML models were created through a modeling tool for UML (e.g. Enterprise Architecture) and converted automatically into XML format. Models are created in high level following VDML language in different levels of complexity stored in a central place for future use. This action is performed on a UML tool (Enterprise Architecture 5.0) which has capacity to produce the XMI (XML Metadata Interchange) code from the model identified. The XMI is a standard of OMG (Object Management Group) for data exchange between platforms. Also the UML and the XMI are the base for MDA (Model Driven Architecture) which the goal (Figure 4) is to define a set of rules and techniques used to modify one model in order to get a different new model. Mappings are used to transform PIM (Platform Independent Model) into PSM (Platform Specific Model) and the opposite. The UML profiles play an important role in MDA, since MDA leverage the usage between models. There are several techniques to perform this transformation and generate automatic code. Templates are a pre-build form, having as input a simplified XML of each model, and the output is the source code of that model. To each model, it is necessary a template for a specific platform. Models and templates are stored in a central database for future use. Models can be used also to store information about certain processes, so we can refer as knowledge. Available models and templates are stored in a central database for future reuse. Programers can contribute with templates or models, and product development engineers contribute with available models and templates to create appropriate actions. The proposed

approach allows the usage of the knowledge captured in previous modeling processes by using “applicational templates”. This allows accelerating the process by employing the concept of develop-by-reuse [38] and minimizing error introduction through a constant quality improvement process of the captured expertise. This same approach, but according to another perspective, can be applied having as its main goal the creation of a template repository, thus supporting the concept of develop-for-reuse [38], whose main objective is to define a requirement document template catalog/library. These templates, “architectural templates”, are based on re-factored requirements, hence improving their range of possible application, assuring simultaneously that they incorporate the best practices of document requirements and use cases of writing patterns [39].

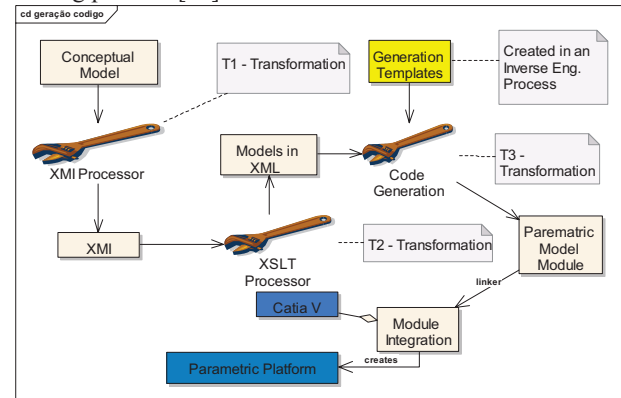


Figure 3: MDAI transformation process.

Among other best practices and well known patterns in the software engineering, we argue that the application of these approaches in the new product development based on MDAI proposal would shorten the time to develop and consequently the time to market new products and would improve the overall quality and robustness of these products. On the other hand, this approach obliges a learning investment from product developer’s stakeholder’s, mainly on modeling process and in the usage of VDML. To reach such goals the methodology requires the following elements: (1) MDAI platform is a tool to support the CAD and CAS Engineer on their job activity. The MDAI platform is sustained by a repository that keeps related information, such as models, templates, drawing information, generated artifacts, technological data base and even information concerning the process itself. The MDAI platform is a web based, multi-user, multi-application and multi architecture tool; (2) VDML/UML profile is a set of coherent UML

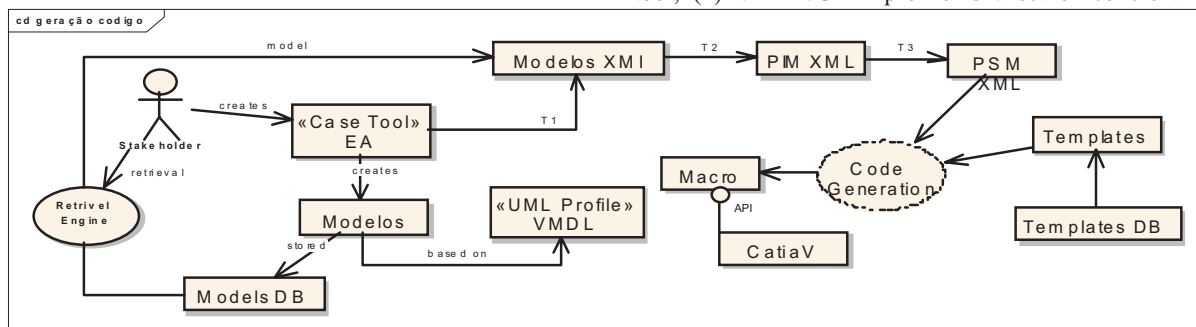


Figure 2: MDAI approach.

extensions that allow a high-level, visual modeling method to design information systems. This model acts, as libraries of standards; (3) MDAI/XML language is a XML language that provides a textual, structured, understandable and compact way of specifying information systems. While MDAI/UML profile allows the definition of visual models, XIS/XML [40] is its textual counterpart that allows the definition of information systems with XML-based, textual specifications.

The quality of the starting models would grant a higher precision output from the forming simulation than that available today, giving a higher degree of acceptance to the following recycle steps.

The design tool (MDAI) is an integrated environment that supports the central tasks of the new product development activity and it is integrated on a CAD tool.

In Figure 3 in the white boxes are the main modules that present the high-level MDAI components architecture and shows the steps to transform models in modules (small pieces of programming code) and link them to build a job routine process, or identify important job related information.

The main tasks to be performed are: (1) **Components**, conceptual models are transformed into XMI (T1) and simplified in XML (T2). From these XML models, and with appropriate templates created in an inverse engineering process, we generate the modules of proposed platform, which are integrated on CAD design tool; (2) **Integration**, the components generated from models through templates and the platform components used should be integrated into a CAD system.

A. Modularity

This methodology makes sense if we make available small pieces, which allow the diverse construction of the new product like a LEGO construction (Figure 4) through the availability of small modules and templates. If we have an: (1) increasing number of models and templates available at central system database; (2) creation of templates for different CAD tools; (3) improve existing templates and models; (4) Retrieve 'desirable' drawings. With this methodology we can use the same modules and approaches to suport the development of different products.



Figure 4: MDAI modular approach.

V. APPLICATION IN AUTOMOBILE INDUSTRY

A. New Product Developers Stakeholders

Human resource and skill management has become an object of growing interest as their importance. Skill management intends to give competitive advantages on organizations by taking into account human resources as strategic assets and integrating them in the business process goals. This perspective already has been studied in knowledge management studies [41-43]. Skill management plays an important role in different organizational activities, such as: (1) personnel recruitment; (2) expert finding; (3) project management; (4) alerts for missing goals; (5) personnel development. In this view we intend to capture relations among actors which participate on the activity process, capture in a contextualizing way the communication between actors', display actors responsibilities and skills.

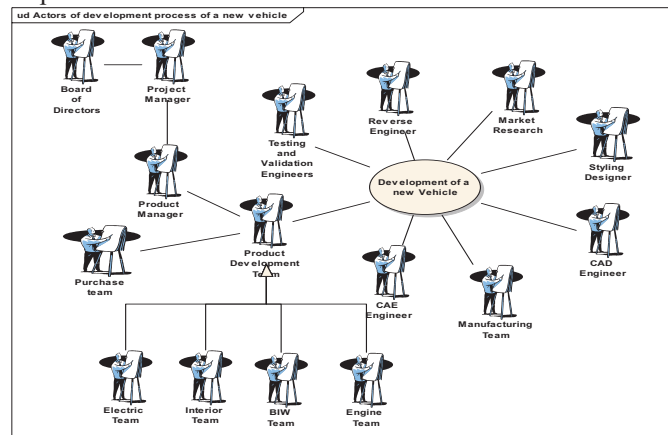


Figure 5: Identification of main stakeholders.

In the development of a new product in the automobile industry the main actors, illustrated in Figure 5 and for a detailed description see [44].

B. Case Study

We already applied this approach on the new product development with a set of tools that automates many of the tasks associated with the feasibility process of an automobile in a CAD (Computer Aided Design) or CAS (Computer Aided Style) system. This software tool will provide assistance to certify the vehicle for compliance with all governmental regulations and standards. It allows the users to access the ergonomic performance of the vehicle. All these tools will be written in the style of macro SCRIPT (Visual basic or c++), wizards and buttons will be created as well as the instructions to complete these, often very complex, tasks. Parameterization of the output geometry allows easily the update of the different vehicle parts. The developed software will be fully used in a parameterized CAD\CAS environment. More information about this case study see [44-45]. The most relevant results of this project are:

- Time reduction in the setting phase of the project;

- Ergonomic studies and vehicle setting phases are unified in one parametric product;
- Standardization of the vehicle process setting and establishing a common language and notation;

The main advantage of the tool is to unify and integrate in a single environment, integrated in a major CAD/CAS platform, the typical functionalities of several programs. Both the parameters relating to ergonomics and the ones concerning vehicle settings are linked together. It was developed an “open parametric environment” that was able to integrate all automotive knowledge packages when developed.

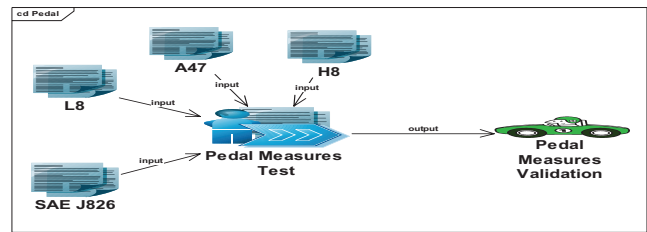
C. Automation of ergonomic checks validation

We apply this approach to most relevant Ergonomics flow checks by modelling this activity using VDMl was performed in EA (Enterprise Architecture), a detailed list can be found at [45]. This information is converted automatically to XMI format and through an XSTL transformation to XML. This XML represents PIM (Platform Independent Models) and are stored in knowledge repositories. Later we can use this XML PIM models and convert then to PSM (Platform Specific Models), oriented to the final platform (our case CatiaV) and through appropriate templates (created in an inverse engineering process) we can generate code to a specific platform. In our case study we generate Catia V macros to perform automatic ergonomic checks. Since this templates to generate code (this case macro, to interact with CatiaV API) don't change much we save programming time using this approach, and high level models allow easy customization. A detail list can be found at [44-45]. Here we illustrate a small example: Pedal check activity, where we need to check maximum and minimum values of three parameters: A47, H8 and L8 defined in ergonomics rule SAE J826. EA model, correspondent XML (PSM model) and output performed is illustrated on Figure 6. A macro created to integrate in Catia V can be found at [46].

VI. CONCLUSIONS

We propose a new methodology for new product development using a novel model-based approach as well as generative programming techniques. The vision underlying the methodology is not new by itself. It is effectively a revisitation of existing expectations that in the past did not have so much success. Applying the proposed methodology to the development of new products bring us the following benefits: (1) reduce the time to create new products; (2) introduction of automatic validations; (3) automation of standard jobs routines; (4) re-use of models and templates at high-level in order to realize this potential; (5) promote the collaboration among different research groups; (6) knowledge reuse; and (7) better productivity, which means less time and lower costs in development activity.

Obviously, these are very ambitious ideas, that can be detailed and focused in a medium term research agenda and they are the application of Software Engineer main approaches to the new product development area.



PIM model of this activity performed in EA using VDMl.

```
<PedalMeasuresTest name = "PedalActivity">
  <SAEJ826>
    <A47>
      <MinAllowedValue>0</MinAllowedValue>
      <MaxAllowedValue>77</MaxAllowedValue>
    </A47>
    <L8>
      <MinAllowedValue>400</MinAllowedValue>
      <MaxAllowedValue>500</MaxAllowedValue>
    </L8>
    <H8>
      <MinAllowedValue>0</MinAllowedValue>
      <MaxAllowedValue>100</MaxAllowedValue>
    </H8>
  </SAEJ826>
</PedalMeasuresTest>
```

PSM model in XML format

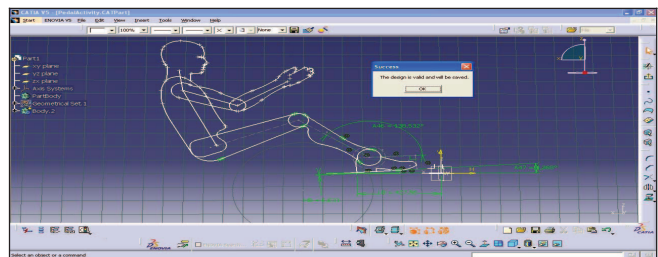


Figure 6. Integration of pedal check activity from modulation (PIM model) to PSM model and automatic check on Catia V.

REFERENCES

- [1] Cris Kobryn, "UML 2001: A standardization odyssey", Communications of the ACM, October 1999, vol.42, No 10.
- [2] Object Management Group, (2003) "MDA Guide Version 1.0.1", June 2003. <http://www.omg.org/cgi-bin/apps/doc?omg/03-06-01.pdf>
- [3] Bosch, J. 2000. Design and Use of Software Architecture: Adopting and Evolving a Product-Line Approach. Addison-Wesley.
- [4] Clements, P. and Northrop, L.M. 2001. Software Product Lines: Practices and Patterns. Addison-Wesley.
- [5] Jacobson, I. Griss, M. and Jonsson, P. 1997. Software Reuse: Architecture, Process and Organization for Business Success. ACM Press.
- [6] Karlsson, E.-A. (Ed.) 1995. Software Reuse, a Holistic Approach. John Wiley & Sons.
- [7] SEI 2005. <http://www.sei.cmu.edu/>.
- [8] IESE - Fraunhofer-Institute for Experimental Software Engineering - <http://www.iese.fhg.de/fhg/iese/>
- [9] Anonsen, S.: Experiences in Modeling for a Domain Specific Language. In: Satellite Activities at the Unified Modeling Language, 7th International Conference (UML 2004), LNCS, vol. 3297, pp. 187-197, Springer (2004)
- [10] Bahler, L., Caruso, F., Micallef, J.: Experience with a Model-Driven Approach for Enterprise-Wide Interface Specification and XML Schema Generation. In: Seventh IEEE International Enterprise Distributed Object Computing Conference (EDOC'03), pp. 288-295 (2003)

- [11] Baker, P., Loh, P.S., Weil, F.: Model-Driven Engineering in a Large Industrial Context - Motorola Case Study. In: ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems (MoDELS/UML 2005), LNCS, vol. 3713, pp. 476-491, Springer (2005)
- [12] Biffl, S., Mordinyi, R., Schatten, A.: A Model-Driven Architecture Approach Using Explicit Stakeholder Quality Requirement Models for Building Dependable Information Systems. In: 5th International Workshop on Software Quality (WoSQ'07) at ICSE'07, IEEE, 6 p. (2007)
- [13] Bloomfield, T.: MDA, Meta-Modeling and Model Transformation: Introducing New Technology into the Defense Industry. In: 1st European Conference on Model Driven Architecture: Foundations and Applications (ECMDA-FA'05), LNCS, vol. 3748, pp. 9-18, Springer (2005)
- [14] Brambilla, M., Ceri, S., Fraternali, P., Acerbis, R., Bongio, A.: Model-Driven Design of Service-Enabled Web Applications. In: ACM SIGMOD International Conference on Management of Data, pp. 851-856 (2005)
- [15] Burgstaller, B., Wuchner, E., Fiege, L., Becker, M., Fritz, T.: Using Domain Driven Development for Monitoring Distributed Systems. In: 1st European Conference on Model Driven Architecture: Foundations and Applications (ECMDA-FA'05), LNCS, vol. 3748, pp. 19--24, Springer (2005)
- [16] Deng, G., Lu, T., Turkay, E., Gokhale, A., Schmidt, D., Netchypurenko, A.: Model Driven Development of Inventory Tracking System. In: 3rd OOPSLA Workshop on Domain Specific Modeling (DSM'03), 6 p. (2003)
- [17] Jonkers, H., Stroucken, M., Vdovjak, R.: Bootstrapping Domain-Specific Model-Driven Software Development within Philips. In: 6th OOPSLA Workshop on Domain Specific Modeling (DSM'06), 10 p. (2006)
- [18] Jouenne, E., Normand, V.: Tailoring IEEE 1471 for MDE Support. In: Satellite Activities at the Unified Modeling Language, 7th International Conference, LNCS, vol. 3297, pp. 163--174, Springer (2004)
- [19] MacDonald, A., Russell, D., Atchison, B.: Model-Driven Development within a Legacy System: an Industry Experience Report. In: Australian Software Engineering Conference (ASWEC'05), pp. 14-22, IEEE (2005)
- [20] Mattsson, A., Lundell, B., Lings, B., Fitzgerald, B.: Experiences from Representing Software Architecture in a Large Industrial Project using Model Driven Development. In: 2nd Workshop on SHaring and Reusing architectural Knowledge Architecture, Rationale, and Design Intent (SHARK-ADI'07) at ICSE 2007, 6 p., IEEE (2007)
- [21] Middleware Company. Model Driven Development for J2EE Utilizing a Model Driven Architecture (MDA) Approach. Productivity Analysis. Report by the Middleware Company on behalf of Compuware, http://www.omg.org/mda/mda_files/MDA_Comparison-TMC_final.pdf (2003)
- [22] MODELWARE D5.3-1 Industrial ROI, Assessment, and Feedback-Master Document. Revision 2.2, URL: <http://www.modelware-ist.org> (2006)
- [23] MODELWARE D5.3-5 Western Geco ROI, Assessment, and Feedback. Revision 0.3, URL: <http://www.modelware-ist.org> (2006)
- [24] MODELWARE D5.3-2 Enabler ROI, Assessment, and Feedback. Revision 1.1, URL: <http://www.modelware-ist.org> (2006)
- [25] Pagel, M., Brörkens, M.: Definition and Generation of Data Exchange Formats in AUTOSTAR. In: 2nd European Conference on Model Driven Architecture: Foundations and Applications (ECMDA-FA'06), LNCS, vol. 4066, pp. 52--61, Springer (2006)
- [26] Presso, M.J., Belaunde, M.: Applying MDA to Voice Applications: an Experience in Building an MDA Tool Chain. In: 1st European Conference on Model Driven Architecture: Foundations and Applications (ECMDA-FA'05), LNCS, vol. 3748, pp. 1--8, Springer.
- [27] Raistrick, C.: Applying MDA and UML in the Development of a Healthcare System. In: Satellite Activities at the Unified Modeling Language, 7th International Conference (UML 2004), LNCS, vol. 3297, pp. 203--218, Springer (2004)
- [28] Safa, L.: The Practice of Deploying DSM, Report from a Japanese Appliance Maker Trenches. In: 6th OOPSLA Workshop on Domain Specific Modeling (DSM'06), 12p. (2006)
- [29] Shirtz, D., Kazakov, M., Shaham-Gafni, Y.: Adopting Model Driven Development in a Large Financial Organization. In: 3rd European Conference on Model Driven Architecture: Foundations and Applications (ECMDA-FA'07), LNCS, vol. 4530, pp. 172-183, Springer (2007)
- [30] Staron, M., Kuzniarz, L., Wallin, L.: Case Study on a Process of Industrial MDA Realization: Determinants of Effectiveness. Nordic Journal of Computing 11(3), 254-278 (2004)
- [31] Staron, M.: Adopting Model Driven Software Development in Industry- a Case Study at two Companies. In: ACM/IEEE 9th International Conference on Model Driven Engineering Languages and Systems (MoDELS/UML 2006), LNCS, vol. 4199, pp. 57--72, Springer (2006)
- [32] Trask, B., Paniscotti, D., Roman, A., Bhanot, V.: Using Model-Driven Engineering to Complement Software Product Line Engineering in Developing Software Defined Radio Components and Applications. In: ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA'06), pp. 846--853 (2006)
- [33] Ulrich, A., Petrenko, A.: Reverse Engineering Models from Traces to Validate Distributed Systems- an Industrial Case study. In: 3rd European Conference on Model Driven Architecture: Foundations and Applications (ECMDA-FA'07), LNCS, vol. 4530, pp. 185-193, Springer (2007)
- [34] Weigert, T., Weil, F.: Practical Experiences in Using Model-Driven Engineering to Develop Trustworthy Computing Systems. In: IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06), pp. 208--217 (2006)
- [35] Wegener, H.: Agility in Model-Driven Software Development? Implications for Organization, Process, and Architecture. URL: <http://www.softmetaware.com/oopsla2002/wegenerh.pdf> (2002)
- [36] Wegener, H.: Balancing Simplicity and Expressiveness: Designing Domain-Specific Models for the Reinsurance Industry. In: 4th OOPSLA Workshop on Domain Specific Modeling (DSM'04), 12 p. (2004)
- [37] Ferreira, João; Furini, Francesco, Silva, Nuno (2007). "VDML: A Modeling Language for New Vehicle Development". Proceedings of International Conference on systems Engineering and Modeling (ICSEM'07), March 20-23, 2007 in Herzeliya and Haifa - Israel.
- [38] XIS/XML - <http://berlin.inesc-id.pt/projects/xis/>
- [39] Cybulski, J., "Application of Software Reuse Methods to Requirements Elicitation from Informal Requirements Texts", PhD Thesis, La Trobe University, Australia, March 2001.
- [40] Examples of requirements document's templates: <http://www2.ics.hawaii.edu/~johnson/413/lectures/5.2.html>, <http://www.devdaily.com/java/misc/ReEnableExternalUser/node1.shtml>
- [41] C. Woodruffe. What is meant by a competency? Leadership & Organization Development Journal, vol.4, n°1,93.
- [42] F. Stone. New Definition of Corporate Competencies. Management Review, vol. 84, no. 6, June 1995.
- [43] G. Hamel, C. Prahalad. The core competence of the corporation. Harvard Business Review, pages 79-91, 1990.
- [44] Ferreira, João; Furini, Francesco, Silva, Nuno (2007). "Framework for an Advanced Design Vehicle Process Development". Proceedings of IASTED International Conference on Modelling, Identification, and Control (MIC 2007), February 12-14, 2007 in Innsbruck - Austria.
- [45] Ferreira, João; A. Tidstam; F. Furini (2007). "Ergonomic Tools in Automobile Product Development". Proceedings of International Conference on Ergonomics 2007, December 3-5, 2007 in Kuala Lumpur Malaysia.
- [46] www.deetc.isel.ipl.pt/matematica/jf/macro.pdf