

# Applying model-driven paradigm: CALIPSONeo experience

M.J.Escalona<sup>1</sup>, J.A. García-García<sup>1</sup>, F. Mas<sup>2</sup>, M. Oliva<sup>2</sup>, C. del Valle<sup>1</sup>

<sup>1</sup>Department of Computer Languages and Systems. University of Seville. Spain  
mjescalona@us.es, julian.garcia@iwt2.org, carmel@us.es

<sup>2</sup>Airbus, Spain {fernando.mas, manuel.oliva}@airbus.com

**Abstract:** Model-Driven Engineering paradigm is being used by the research community in the last years, obtaining suitable results. However, there are few practical experiences in the enterprise field. This paper presents the use of this paradigm in an aeronautical PLM project named CALIPSONeo currently under development in Airbus. In this context, NDT methodology was adapted as methodology in order to be used by the development team. The paper presents this process and the results that we are getting from the project. Besides, some relevant learned lessons from the trenches are concluded.

**Keywords:** Model-Driven Paradigm, Product Life-Cycle Management (PLM), industrial Digital Mockup (iDMU)

## 1. Introduction

The model-driven paradigm is a paradigm that combines the power of concepts and its relations, using abstracts models, in order to offer suitable mechanisms for software development. This paradigm is focused on concepts and how these concepts evolved in the life cycle of a product. We can refer to model-driven engineering (MDE) [1], when this paradigm is applied in the context of engineering. In the last years, software engineering community has applied MDE in different contexts of software engineering, getting suitable results. For instances, it was used in the context of the web engineering, in methodologies such as UWE (UML Web Enginnering) [2] or WebML (Web Model-Languages) [3]; in the context of software testing, with approaches such as [4][5]; in software product lines, such as [6]; or even in more specific contexts like software architecture, with the approach WebSA [7] or in the context of RIA (Rich Internet Applications) [8] These researches have produced suitable results that could be considered an inspiration source for the industrial community. However, very few experiences were reported from the enterprise [9].

This paper presents a real application of MDE in an industrial context. NDT (Navigational Development Techniques) [10] is a MDE methodology that was applied in a large number of real projects. Nevertheless, in this situation, it was adapted and applied in a new context, the aeronautical one, which entails modifying the methodology and its management policies. The paper presents this version and a global view of the experience. It analyses learned lessons and future works and

concludes that the application of the model-driven paradigm in aeronautical contexts can be a relevant area and can offer very successful results.

## **2. CALIPSONeo**

CALIPSONeo (advanced Aeronautical solutions using PLM processes & tools) is an ambitious project developed in Airbus that has as a main target the definition through a comprehensive requirements collection process, using PLM (Product Life-Cycle Management) software existing in the market, that will allow the industrial engineers to define, simulate, optimize and validate the aeronautical assembly processes in a 3D virtual environment before they are implemented in a real shop floor. Through the years, many analyses were performed on PLM and its benefits for Airbus [11]. From this information, the project team has developed the project requirements documentation, the requirements management plan, and the requirements traceability matrix that satisfies the PLM business needs. CALIPSONeo covers the design of a new PLM methodology to conform to a PLM collaborative design and the required development of the software that satisfies that concept [11]. CALIPSONeo is subdivided into three individual subprojects, in order to effectively manage the work needed to complete it. It will let the project team more effectively manage the project's scope as they work on the tasks necessary to complete the project. Each of the three projects is subdivided in work packages. Tasks are the next level below work package, which are the minimal entity the project can be structured into.

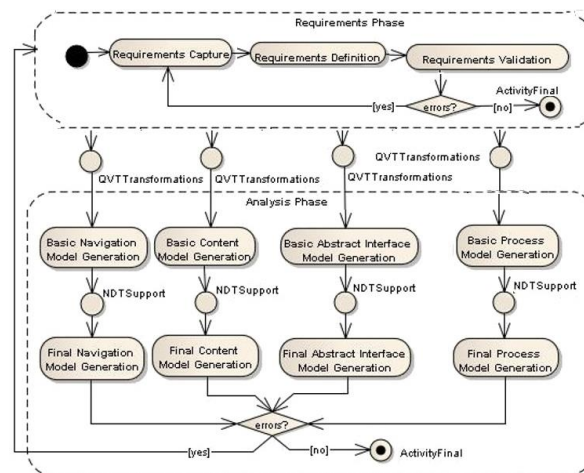
Following this structure, CALIPSONeo consists of three subprojects: MARS (autoMated shop-floor documentation updating System), PROTEUS (PROcess sTructure gEneration and Use) and ELARA (gEneraLization to assembly oriented authoring Augmented ReAlity). These subprojects are independent and teams involved in each of them also differ. However, subprojects have to be coordinated and they have to be correctly integrated because they have common actors who demand common functionality. Thus, the main objective ELARA must achieve is to develop a system which should provide workers in assembling tasks, with augmented reality technology, still valid on any airframe, using the 3D information contained in the iDMU (industrial Digital Mock Up) [13] or information coming from MARS. The final result will be an industrial prototype to be used in the assembly process for the FanCowl product in the A320Neo program. PROTEUS is responsible for the Product, Process and Resources structure definition and the their interrelations to create an iDMU and MARS is responsible for exploiting the iDMU in order to obtain automatically the documentation needed at shop floor for the airplane assembly. Thus, despite each of them have different functions and aims; it is required to guarantee that they are incorporated, connected and well-integrated.

## **3. Background**

### **3.1 NDT-Navigational Development Techniques**

Initially, NDT was defined as a MDE methodology focused on requirements and analysis processing. At the beginning, it dealt with defining a set of formal metamodels for the requirements and analysis phases. In addition, NDT defined a set

of derivation rules, stated with the standard QVT (Query View Transformations)[14], which generated analysis models from requirements model. The main goal of the Requirements phase of NDT is to build the catalogue of requirements containing the needs of the system to be developed. It is divided into a series of activities: capture, definition and validation of requirements. Requirements can also be classified according to their nature: information storage requirements, functional requirements, actor requirements, interaction requirements and non-functional requirements. NDT defines derivation rules to generate the analysis phase models, once the requirements specification phase has been completed and the catalogue of system requirements has been drafted and validated. Figure 1 shows this idea through the stereotype «*QVTTransformation*». The transition between the requirements and the analysis model is standardized and automated. It is based on QVT transformations, which translate the concepts of requirements metamodels into the first versions of the analysis models. These models are known in NDT as basic models of analysis. For example, the basic conceptual model of analysis is obtained from the storage requirements defined during the requirements phase. Thereafter, the team of analysts can transform these basic models to enrich and complete the final model of analysis.



**Figure 1** Transformations from Requirements to Analysis

As this process is not automatic, the expertise of an analyst is required. Transformations are represented in Figure 1 through the stereotype «*NDTSupport*». NDT controls these transformations by means of a set of defined rules and heuristics, to ensure consistency between requirements and analysis models. This idea, which was only applied in the requirements and analysis phases, was extended in the last years and, currently, this context supports the complete life cycle: viability study, requirements, analysis, design, implementation, maintenance and testing under different life cycles, such as classical, agile or the one based on prototypes<sup>1</sup>. To sum up, NDT offers an environment conducive to Web systems development, completely covering the software development life cycle. Recently, NDT has been applied to many practical environments where it has succeeded due to the application of transformations among models, which has reduced the development time.

<sup>1</sup> You can get more information about NDT full life cycle in [www.iwt2.org](http://www.iwt2.org)

### 3.2 NDT-Suite

The application of MDE and, particularly, the application of transformations among models, may become monotonous and very expensive if there are no software tools that automate the process. Therefore, NDT has defined a set of supporting tools called NDT-Suite<sup>2</sup> [15] to meet this need. Currently, the suite of NDT comprises the following main free Java tools: (a) NDT-Profile is a specific profile for NDT, developed using Enterprise Architect. NDT-Profile offers the chance of having all the artifacts that define NDT easily and quickly as they are integrated into a tool called Enterprise Architect [16]. (b) NDT-Quality is a tool that automates most of the methodological review of a project developed with NDT-Profile. It checks both, the quality of using NDT methodology in each phase of software life cycle and the quality of traceability of MDE rules of NDT. (c) NDT-Driver, allows transformation can be automatically applied every phase of the life-cycle. (d) NDT-Prototype is a tool designed to automatically generate a set of XHTML prototypes from the navigation models, described in the analysis phase, of a project developed with NDT-Profile. (e) NDT-Merge uses the comparison among metamodels to identify syntactic and semantic inconsistencies among different versions of the same requirements catalogue. In addition, NDT-Suite has more tools: NDT-Report, NDT-Glossary, NDT-Checker or NDT-Counter. You can see the purpose of these tools on IWT2 website.

### 3.3 NDTQ-Framework

In the last years, NDT has evolved again and now, in order to offer a suitable and a global solution for the real application of NDT, a global framework named NDTQ-Framework<sup>3</sup> was developed. NDTQ-Framework comprises a set of processes involving development processes, management processes, quality processes, testing processes and security processes. This environment is based on different reference models like CMMi (Capability Maturity Model Integration) [17] and ITIL (Information Technology Infrastructure Library) [0], and its application in real projects are certificated under different standards like ISO 27001, ISO 9001:2008, UNE EN 16602 and ISO 14000. This paper does not aim to present NDTQ-Framework in detail, but you can download more information from IWT2 website. However, for this paper is necessary to know that the initial management processes defined by NDTQ-Framework have to be adapted to support the development and the management of CALIPSONeo.

## 4. Adapting NDT and management rules

NDT has been applied to several real projects, since it was used in CALIPSONeo. However, CALIPSONeo provides a new environment with very relevant aspects: (1) CALIPSONeo is, in fact, a set of three projects. NDT and its management rules have to be adapted for improving and supporting integration and coordination among these projects. (2) CALIPSONeo is a special project for NDT application because, in fact, it

---

<sup>2</sup> In IWT2's website ([www.iwt2.org](http://www.iwt2.org)), you can read more about NDT-Suite and its tools.

<sup>3</sup> In IWT2's website ([www.iwt2.org](http://www.iwt2.org)), you can read more about NDTQ-Framework.

produces very “few” software. With CALIPSOneo, we are going to obtain a product that mainly has to cover both a set of requirements and a set of policies to be applied to PLM. This project is more centered in requirements and work policies than in code. In this sense, it offers a suitable environment to confirm whether MDE can support this kind of development. (3) CALIPSOneo has a very heterogeneous team project. In fact, the team project is distributed around different cities in Spain. This situation even complicates coordination policies and communication among the different teams. Finally, several aspects were modified in order to cope with all the aforementioned points

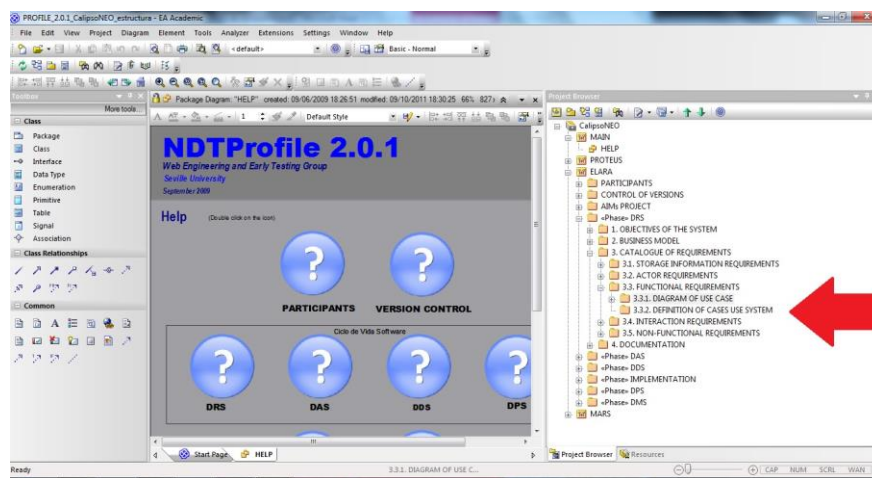


Figure 2 NDT-Profile for CALIPSOneo

#### 4.1 Managing the coordination

NDT-Profile, NDT-Quality and NDT-Driver were adapted to coordinate these three teams. Besides, supporting this change involved some management processes in NDTQ-Framework. Firstly, NDT-Profile was modified extending the original profile to offer a “project composed by subprojects”. Figure 2 represents the main interface of NDT-Profile for CALIPSOneo. On the right margin, we can see three different folders, each of them for one subproject. In the figure, only Elara is opened and the rest remains the same. In fact, subfolders included in each subprojects constitute the same folders that we could find in the original NDT-Profile. In this sense, the original metamodel of NDT and its profile were extended to support the possibility of a “project composed by projects”. The inclusion of this new aggregation in the metamodel, caused the necessity of extended NDT-Suite and its tools to support this new association correctly. In contrast, if we do not offer management rules to teach our teams how to work in liaison, modifying the profile and tools will not be enough. Thus, teams in each project work under some specific coordination rules, which are later explained in section 4.3.

#### 4.2 Managing “Software” products

There are more experiences related to code generation in the context of MDE. In fact, there is a historical interest in the code among the software community. However, in

CALIPSONeo a full PLM is being developed and in this context, the “code” is no so relevant. In the context of CALIPSONeo, requirements and analysis models acquire a very relevant role because they are the key to guarantee the quality of the results. As a matter of fact, despite we have to develop code, the most important point is how this new code and PLM tools will work together to get the final PLM. In this sense, NDT-Profile is offering a very suitable common repository for teams. In the project, we have an ECM<sup>4</sup>-based documental repository for storing each document, which stores each document, PLM policy or software piece of document integrated in NDT-Profile and referenced via “links”. In addition, different automatic traceability matrixes were generated in the profile to connect each artifact between phases. Thus, NDT-Profile integrates the great deal of knowledge of the project.

### 4.3 Managing “management”

As previously mentioned, these technical mechanisms are not useful without providing suitable mechanisms for managing the project. In this sense, a set of management processes were defined in the context of CALIPSONeo to guarantee that we can manage MDE mechanisms as well as the quality of results.

Thus, as MDE concerns, teams work with the same Enterprise Architect File. Nevertheless, as they work in different periods and each subproject has different advances, we have designed a “master file” that is only managed by the quality management team. Approximately every month, this team gathers all the work executed by Elara, Proteus and Mars teams in the master file. This involves three different steps:

1. Firstly, a quality review with the new version of NDT-Quality is executed. In this step, inconsistencies among models or problems in projects interconnections are identified. Besides, NDT-Quality checks traceability matrixes in order to guarantee that subprojects keep consistency between phases and artifacts
2. Secondly, if the teams develop a new phase, NDT-Driver will be executed in order to “translate” the acquired knowledge along in phases. For instance, in March 2013, the teams finished the requirements phase in the third integration of CALIPSONeo. The quality team used NDT-Driver and generated analysis models following the idea presented in Figure 1. Thus, teams received in this third integration a first version of the analysis with the automatic generation of their requirements knowledge thanks to transformations. These transformations not only generated analysis models, but they also produced traceability matrixes to save the origin of each artifact in the analysis in relation to requirements.
3. Finally, a new baseline of the master file is generated and stored in Alfresco to use at different teams’ disposal.

This mechanism of integration is quite useful because every team, even, everyone in the project, can work independently. However, using the stored connection among elements generated from NDT-Driver transformations and baselines will help us easily manage each team’s work. Thus, the integration process does not take the quality team more than one hour per month. Obviously, if inconsistencies are identified in step 1, this time will be extended, as the quality team will have to work with teams to face them.

---

<sup>4</sup> ECM is acronym of Enterprise Contents Management and within CALIPSONeo, the Alfresco solution has been used. You can find more information about Alfresco in its website.

## 5. Learned lessons

As previously introduced in the abstract, CALIPSOneo is a real project where MDE can show all its strengths. Nowadays, although the project is in the analysis phase and there are long future ways to go, we can offer some initial learned lessons. The first one is about MDE. The context of the model-driven paradigm is not always easy to understand by project teams. In fact, concepts like metamodels, transformations and some others, can be too abstract for the enterprise context. In contrast, if we enrich this context with suitable tools, such as NDT-Profile or NDT-Driver, they can become powerful for companies, since they reduce the cost of the project and the number of errors and inconsistencies between the different phases. In MDE, knowledge is “transferred” from phases and transformations guarantee this transfer, becoming a powerful resource for companies, if carried out automatically.

Additionally, MDE is not exclusively used in code generation. We have the intention to demonstrate with CALIPSOneo that we are using it in other phases, being very profitable in environments not mainly oriented towards code.

Nevertheless, it is clear that MDE, mainly in big projects with heterogeneous and distributed teams, requires management rules like every software project, which must ensure the quality of results.

## 6. Final conclusions and future work

This paper has pointed out how a MDE paradigm is being used in the context of a real project named CALIPSOneo. As it was presented, CALIPSOneo gives us a very interesting opportunity to practically measure the power of this paradigm in a context that offers special characteristics, such as multiple and heterogeneous work teams without experience or knowledge in MDE, or a context not specifically oriented towards software. This has let us demonstrate the power of this paradigm in different contexts and explore new ways to improve our research in this line.

Thus, we are working in a new version of NDT-Merge, which allows us to automatically detect errors in general, not only those dealing with quality that we usually identify with NDT-Quality. We are working in the effective comparison of different instances of requirements definition to find out syntactic and semantic problems among different versions of requirements.

The “back” is also another open line. NDT-Driver and NDT-Quality are prepared to detect changes in previous phases. It means that after the analysis generation, requirements definition can evolve and they can lose the consistency with previous analysis artifacts generated. Our tools check these evolutions and present these inconsistencies to the quality team. Moreover, NDT-Driver is prepared for executing transformations only in new versions of artifacts, keeping the same form in the original project. Therefore, we have to keep on working in this line because it would be very interesting that these evolutions could be detected by NDT-Profile, while the teams are working on it, for instance, through a warning to the engineer.

To conclude, we would like to insist on the importance of this kind of experience in the context of our research. The effective possibility to test our tools, approaches and ideas in real contexts with real projects is a required and essential practice to help us propose new approaches suitable for the enterprise context.

## Acknowledgements

This research has been supported by the Tempros project (TIN2010-20057-C03-02), and by the project NDTQ-Framework (TIC-5789) of the Junta de Andalucía, Spain and CALIPSOneo Project.

## References

1. Schmidt, D.C. "Model-Driven Engineering," IEEE Computer, Computer Society, vol. 39, no. 2, pp. 25-31, 2006.
2. UWE (UML-based Web Engineering). <http://uwe.pst.ifi.lmu.de/>. Last accessed May 2013.
3. WebML (Web Model Language) . <http://www.webml.org>. Last accessed May 2013.
4. Gutierrez, J.J., Nebut, C., Escalona, M.J., Mejías, M., Ramos, I. Visualization of use cases through automatically generated activity diagrams. LNCS 5301. pp. 83-96. 2008.
5. Robles, E., Grigera, J., Rossi, G. Bridging Test and Model-driven Approaches in Web Engineering. LNCS. 5648 (2009) pp.130-150. 2009.
6. Bertolino, A., Gnesi, S. PLUTO: A Test Methodology for Product Families. LNCS. pp 181-197. 2004.
7. Melia, S., Gómez, J. The webSA approach: applying model driven engineering to web applications. Journal of Web Engineering 5 - 2, pp 121-149. 2006
8. Robles, E., Escalona, M.J., Rossi, G. Modelling the requirements of rich internet applications in WebRE. Communications in Computer and Information Science. 2012.
9. Mohagheghi, P., Gilani, W., Stefanescu, A., Fernandez, M. A. An empirical study of the state of the practice and acceptance of model-driven engineering in four industrial cases. Empirical Software Engineering, 18(1), 89-116. 2013.
10. Escalona, M.J., Aragón, G. NDT: A Model-Driven Approach for Web requirements, IEEE Transactions on Software Engineering. Vol. 34. N° 3, pp 370-390, 2008.
11. Mas, F., Rios, J., Menendez, J.L., Gomez. A., A process-oriented approach to modeling the conceptual design of aircraft assembly lines, International Journal of Advanced Manufacturing Technology 2012, Vol. 62,
12. Mas, F., Menéndez, J.L., Oliva, M., Ríos, J., Collaborative Engineering: an Airbus case study. 5th Manufacturing Engineering Society International Conference. 2013.
13. Mas, F., Gomez. A., Menendez, J.L., Rios, J., Proposal for the conceptual design of aeronautical final assembly lines based on the iDMU concept. 10th International Conference on Product Lifecycle Management. 2013.
14. Query/View/Transformation. [www.omg.org/spec/QVT/1.1/](http://www.omg.org/spec/QVT/1.1/). Release 1.1. 2011. Accessed 05/2013.
15. García-García J. A., Alba M., García-Borgoñón L., Escalona M. J.: NDT-Suite: A Model-Based Suite for the Application of NDT, LNCS 7387, pp. 469–472, (2012)
16. Enterprise Architect. [www.sparxsystems.com.au](http://www.sparxsystems.com.au). Accessed in May 2013.
17. M.B. Chrissie, M. Konrad, S. Shrum, "CMMI® for Development: Guidelines for Process Integration and Product Improvement," Editorial Pearson Education. 2011.  
A. Jong, A. Kolthof, "Fundamentos de ITIL, Volumen 3," Van Haren Publishing, ISBN 9087530609, 2008.