

# NDT. A Model-Driven Approach for Web Requirements

María José Escalona and Gustavo Aragón

**Abstract**—Web engineering is a new research line in software engineering that covers the definition of processes, techniques, and models suitable for Web environments in order to guarantee the quality of results. The research community is working in this area and, as a very recent line, they are assuming the Model-Driven paradigm to support and solve some classic problems detected in Web developments. However, there is a lack in Web requirements treatment. This paper presents a general vision of Navigational Development Techniques (NDT), which is an approach to deal with requirements in Web systems. It is based on conclusions obtained in several comparative studies and it tries to fill some gaps detected by the research community. This paper presents its scope, its most important contributions, and offers a global vision of its associated tool: NDT-Tool. Furthermore, it analyzes how Web Engineering can be applied in the enterprise environment. NDT is being applied in real projects and has been adopted by several companies as a requirements methodology. The approach offers a Web requirements solution based on a Model-Driven paradigm that follows the most accepted tendencies by Web engineering.

**Index Terms**—Software Engineering for internet project, surveys of historical development of one particular area, requirements/specification.

## 1 INTRODUCTION

SOME years ago [22], a few research groups, started to analyze the characteristics of new types of software systems known as hypermedia systems which have since evolved into Web systems. It was the birth of a new line in software engineering that is currently called Web engineering.

Web engineering is the systematic, structured, and quantifiable application of methodological proposals to the development, evaluation, and maintenance of Web applications [12].

Web Engineering is a young line that is being widely studied. However, the research study in this field is mainly oriented to the analysis and design phases.

Surveys and comparative studies conclude that there are important gaps in this line [3], [49], [15], [18]. The requirements treatment is one of them. The correct treatment of requirements assures the quality of results in a Web project and this phase is now considered in the most recent approaches.

The high number of approaches without standard consensus, the lack of the use of standards, and the scarcity of both practical experience and tool support are other aspects that should be developed in more depth in this line.

In Section 2, a global analysis of the situation is presented. This paper analyzes the present situation of Web engineering in Section 2.1 and it presents how some

gaps are being solved with the use of Model-Driven Engineering (MDE).

A Model-Driven paradigm is a suitable solution for Web development and Web engineering is a specific domain in which MDE can be usefully applied [54]. The application of MDE in Web engineering is called Model-Driven Web Engineering (MDWE) and, as stated in Section 2.2, is offering very good results.

In Section 3, a combination of MDWE and classic Web engineering is presented in a Web requirements approach named Navigational Development Techniques (NDT) [14].

NDT is a methodology that covers the requirements and the analysis phases in Web developments. Thus, it tries to fulfill the lack of requirements treatment in Web engineering. Section 3 starts with a global vision of this approach and it present its life cycle and its scope. This approach is completely based on MDWE and, in Sections 3.2 and 3.3, NDT metamodels and transformations are introduced.

In order to offer a suitable support for the NDT application, a tool case, named NDT-Tool [13], is offered with the methodology. The presentation of NDT in Section 3 finishes with the introduction of this tool in Section 3.4.

NDT is also interesting because it has been widely applied in the business environment. In Section 4, this paper presents the most relevant applications of NDT and how they influence the theoretical research results.

This paper ends by drawing conclusions from these studies and future work along these lines is proposed.

## 2 RELATED WORK

NDT is an approach based on two important research lines. The first one is Web engineering approaches. NDT is not only a methodology focused on requirements and analysis, but it is also related with several other approaches.

The second one is the MDWE environment since its development process is based on this paradigm. In this

- M.J. Escalona is with the Departamento de Lenguajes y Sistemas Informáticos, ETS de Ingeniería Informática, University of Seville, Avenida Reina Mercedes S/N, 41012 Seville, Spain. E-mail: mjescalona@us.es.
- G. Aragón is with Everis Spain, C/Gregor J. Mendel, Edificio DaVinci, 41092 Seville, Spain. E-mail: gustavo.aragon@everis.com.

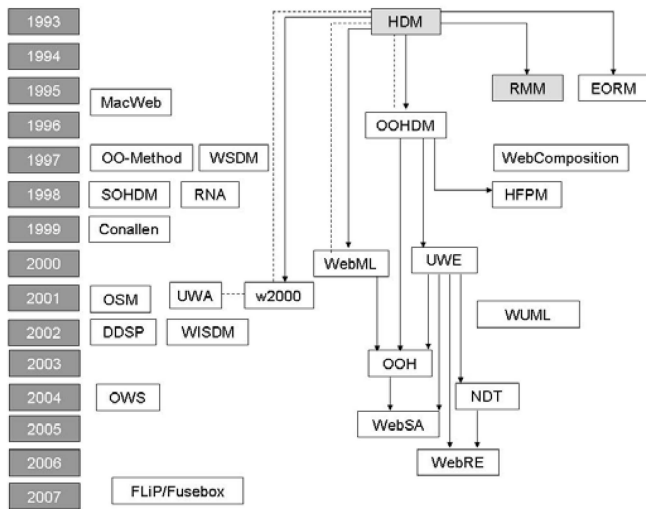


Fig. 1. Web methodologies.

section, a survey along the two lines is presented. Both lines are very popular today and there is quite recent work in this area; for this reason, the section only analyzes approaches similar to or connected to NDT.

## 2.1 Web Engineering Approaches

In recent years, the growing interest in the Internet has led to the generation of a high number of proposals which offer a frame of reference for the Web environment. Fig. 1 shows the most representative ones in chronological order.

In the diagram, continuous lines indicate that the latest methodologies are based on, or receive the ideas from, the previous ones. The dashed lines link the methodologies which have the same author.

In the beginning, the overall tendency was oriented toward the structured paradigm. Thus, the shadowed approaches, Hypermedia Design Model (HDM) [22] and Relation Management Method (RMM) [27], were structured. However, with the introduction of Enhanced Object Relationship Methodology (EORM) [34] and Object-Oriented Hypermedia Design Method (OOHDM) [50], this tendency moved to the object-oriented paradigm.

As can be deduced from some surveys and comparative studies [3], [49], [15], [18], the initial interest of groups was focused on offering new models and suitable techniques to deal with hypermedia system aspects. Thus, HDM was an extension of Entity-Relation Diagrams (ERDs) [8], [9] in the representation of hypermedia concepts. However, research work has since improved and approaches have started to offer a complete life cycle oriented to the new Internet systems.

Some of the approaches presented in Fig. 1 have not been developed over recent years, although others, such as UML-Based Web Engineering (UWE) [29], OOHDM, and Web Modeling Language (WebML) [7], are being reviewed in order to adapt and improve their approaches.

Another important fact is the analysis of the phases treated by each approach. Table 1 shows the route of the different phases of the life cycle treated by each methodology. In Table 1, phases of requirements treatment, analysis,

TABLE 1  
Web Phases Treated by Each Approach

	Requirements	Analysis	Design	Implementation	Tests
HDM					
RMM					
EORM					
MacWeb[42]					
OOHDM					
WSDM [11]					
OO-Method[46]					
WebComp.[23]					
SOHDM [35]					
RNA [4]					
HFPM [43]					
Conallen[10]					
WebML					
UWE					
W2000 [1]					
OSM[36]					
WUML[28]					
UWA <sup>1</sup>					
WISDM[55]					
DDSP[37]					
OOH[5]					
NDT					
OOWS[21]					
WebSA[38]					
WebRe[17]					
FLiP/Fusebox 5.1 <sup>2</sup>					

<sup>1</sup> www.uwaproject.org.

<sup>2</sup> www.fusebox.org.

design, implementation, and tests are presented in the columns and approaches are given in the rows.

It is necessary to add that the separation between phases is based on the comparative study presented in [18]. In Web engineering, there is no consensus and a clear separation between phases. In this table, ideas presented in this survey were followed.

A cell without color indicates that this approach does not consider this phase in its life cycle. The shadowed color points out that the proposal includes a phase based on classic proposals, but that it does not include special proposals for the Web. The dark color means that the proposal covers the whole phase, including specific methods and models for the Web environment.

By analyzing comparative studies, we can conclude that Web methodologies still present a relative lack of maturity and high ambiguity in some aspect such as in the definition of the covering of a life cycle, the performance of activities and tasks, the use of models and techniques, and handling of other aspects [5].

As can be observed in the table, most works have been focused on the design phase. However, this trend is changing. In recent years, research groups have detected the need to enrich their proposals with specific models for requirements processing on the Web. Thus, approaches like OOHDm, which did not consider the requirements phase in its first versions, now include a specific phase which proposes taking into account the use cases and its own technique named User Interaction Diagrams (UIDs) [56] to represent requirements suitable for Web systems.

Empirical experience [13] shows that, due to the growing complexity of Web systems, it is becoming more and more important to capture the requirements needed to model navigation and other important aspects for the Web. In this respect, some issues have arisen:

1. *Should the Web requirements be captured and treated in a specific way?*

Empirical results show that requirements should be treated carefully. Web systems are becoming more and more complex and it is necessary to know the requirements needed as soon as possible or to at least control their growth to guarantee the quality of the system. Moreover, the special characteristics of Web systems require special necessities.

2. *Assuming the treatment of Web-specific characteristics is improved in the first phases of the life cycle, which specific techniques must be considered?*

According to the survey [5], the use of techniques, such as use cases, which capture those requirements should suffice. However, as some groups and research works have shown [26], [56], in many cases, use cases are inadequate and too ambiguous to capture all of the required details for the definition of navigation.

3. *Is it necessary to include the user or client in the treatment of Web requirements?*

Empirical experience shows that, when it is possible and final users or, at least a group of expert users are known, it could be very useful. Web systems, and also the navigational systems on which

they are based, are becoming more and more complex. Users or clients, experts in the business environment, can offer important knowledge for the development team. That is why the techniques proposed for Web requirements have to be detailed enough to be useful to the development team and to offer a flexible validation system of the results for the user and client.

Another important issue is the need to use standard aspects [17]. As already mentioned, there are too many methods, techniques, and terminologies defining the same aspect. Work has to be oriented to assume accepted points and to propose new ideas only for those aspects which are really necessary.

Last, an important aspect which has not yet been mentioned should be emphasized here. Web systems change very quickly. They have to be updated and they are characterized by different types of users, who are, in many cases, undefined. That is why, from the modeling and methodological environment points of view, it is necessary to ease the maintenance and development of those systems with suitable tool support.

Most Web proposals lack the support of CASE tools, which support the technique application and the generation of results, as well as the process of maintenance and validation of models and final results. Some approaches, such as WebML, which offers WebRatio,<sup>1</sup> UWE, which offers ArgoUWE,<sup>2</sup> or FLIP Methodology, which is supported by a framework, Fusebox, are working along these lines. Obviously, the need to offer systematic or even automated definition, generation, or result validation is imminent in this type of environment and these tools provide the appropriate support.

## 2.2 Model-Driven Web Engineering

As introduced in the previous section, empirical experience shows that, due to the growing complexity of Web systems, it is becoming more and more important to capture the requirements needed to model specific characteristics of Web environments. However, empirical studies also conclude that it is sometimes very difficult to translate users' necessities into Web analysis models [13].

Maintaining consistency between models in different phases of the life cycle is of prime importance to assure the quality of results. This aspect becomes critical in the requirements phase. Requirements are frequently defined using nonformal models and the translation into analysis models depends on the analyst's experience.

Current research whose aim is to maintain consistency is in the use of Model-Driven approaches [51]. This tendency, named MDWE, is the application of MDE to the domain of Web system development. As introduced in [41], Web engineering is a specific domain in which MDE can be usefully applied.

In MDE, concepts have the greatest importance, independent of their represented action. MDWE proposes representing concepts using metamodels. The development process is supported by a set of transformations and

1. <http://www.webratio.com>.

2. <http://www.pst.informatik.uni-muenchen.de/projekte/argouwe>.

relations between concepts that leads to agile developments and assures consistency between models.

The power of MDWE is provoking classic approaches into evolving into this new paradigm. Thus, in [40] and [51], metamodels for WebML are presented. In [1], a metamodel for W2000 is offered. Even some initial Model-Driven approaches, such as UWE, are evolving into the new standard defined by the Object Management Group (OMG), for example, the use of QVT [48] for defining transformations [31].

The application of MD systems within the Web environment is increasing daily. As mentioned earlier, it offers important benefits in development and consistency in addition to the definition of concepts. This trend is also followed for the requirements phase.

One of the most recent pieces of research is [53]. This paper presents an approach for the transformation of a Web requirements model into a set of prototypes whereby requirements treatment is based on the task metaphor. Valderas et al. offer an extension of this approach to deal with the specific characteristics of Web requirements and then present a way to derive the navigational model of OOWS. First, they propose defining requirements as tasks and these tasks are translated into an AGG Graph. Using Graph transformations, analysis models are obtained. This approach is supported by a tool which enables the application of the complete cycle. This work is very interesting since it offers a suitable solution for transformation supported by a tool. However, its transformations are not based on OMG tendencies. This provokes incompatibility with other similar approaches.

In [17], the power of metamodels is presented. In comparative studies on Web approaches, a general conclusion is that similar concepts are used or represented with a different number of models, techniques, or artifacts. Thus, for instance, navigational classes are presented with different elements in UWE, OOHDM, NDT, and W2000. Escalona and Koch show how a metamodel can represent a concept independently of its representation or notation; only concepts are important. A metamodel for Web requirements, named WebRe, which represents requirements models of W2000, NDT, OOHDM, and UWE, is presented. In [30], their work is continued using QVT to obtain analysis models from this metamodel. These papers are interesting since they are based completely on UML [45] and on QVT, standards defined by OMG. However, these papers can be considered to be excessively theoretical.

This tendency to use metamodels and transformations to make different approaches compatible is applied in recent work under the name of MDWEnet [54]. MDWEnet is an initiative started by a representative group of MDWE researchers in an effort to find a common approach which allows different approaches to be represented and handled. This group is currently analyzing OO-H, UWE, and WebML with special focus on design models and, although this work is very recent, they are working toward the attainment of a common language in Web engineering.

Fernández and Mozón [20] present the possibilities of working with metamodels and tools. Thus, they present how a requirements metamodel can be easily defined in

Integral Requisite Analyzer (IRqA).<sup>3</sup> IRqA is a commercial tool that helps in the definition of metamodels for requirements. In this way, this paper presents the power of tools that support metamodels because they are suitable for any approach defined using metamodels. This work is very practical in fact, although it is not an approach for the Web. Metamodels do not offer specific artifacts to deal with the Web environment since only an approach for classic requirements treatment is offered.

However, although these studies are specifically for requirements, other classic approaches are being used in the MDE environment. For instance, in [40] and [51], some metamodels for WebML can be found whereby metamodels can represent classic concepts independent of the artifact used in their representation.

In [38], Meliá and Gómez present an approach called Web Software Architecture (WebSA). WebSA provides the designer with a set of architectural models and transformation models which can be used to specify a Web application. Although these models work in the design phase, this approach is very interesting since Model-Driven Architecture (MDA) [44] and QVT are employed in a very exhaustive way.

To conclude MDA and MDWE in a more general way, the use of metamodels and MDE are areas of software engineering that are becoming widely accepted by the Web engineering community as a solution for classic problems.

### 3 NAVIGATIONAL DEVELOPMENT TECHNIQUES

As mentioned in the previous section, several comparative studies have proven that one of the least studied phases in Web engineering is the requirements phase [1], [49], [15], [18]. Most approaches in Web engineering are focused on analysis and design phases. They usually propose using classic requirements techniques, such as use cases, in order to capture and define requirements on the Web.

Although use cases are a suitable technique for dealing with requirements and are very easily understood by the user, they are frequently very ambiguous [56], [26]. For this reason, several research groups have been working on specific requirements treatment for the Web environment.

Another conclusion drawn from comparative studies is that, in Web engineering, different aspects of software are treated in separate ways. This idea is followed in the analysis and design phases by several approaches. OOHDM, WSDM, WebML, and OOH, where content, navigation, interaction, etc., are modeled differently, are only a few examples. This idea of concept separation can be moved to the requirements phase in order to obtain these advantages. Thus, UWE deals separately with information requirements, functional requirements, etc., and W2000 defines different use cases for functional and navigational requirements.

Finally, another important fact detected by the comparative studies is that requirements are sometimes defined in a very ambiguous way and it is very difficult for analysts to translate the knowledge from the requirements definition to the analysis models.

3. <http://www.irqaonline.com/>.

With these ideas, NDT was proposed [14]. NDT is a methodological approach which deals with requirements in Web environments. NDT was proposed in order to support the requirements engineering and the analysis phase of Web systems and is based on the MDE paradigm.

Thus, it proposes an MDE approach in order to offer a suitable environment for the capture, definition, analysis, and validation of Web requirements. In this section, NDT will be presented. In Section 3.1, a brief view of its life cycle and its structure is shown. In Section 3.2, metamodels for NDT are introduced. In Section 3.3, transformation of its Model-Driven approach is presented.

NDT also has its own tool support. In Section 3.4, NDT-Tool, its associated tool, is presented.

This section is completely focused on the theoretical presentation of NDT. However, NDT is not only a theoretical approach. It is being applied in several real projects by companies in Andalusia, Spain. Thus, in Section 4, the introduction of NDT ends with the analysis of its practical experience and evolution.

### 3.1 NDT Life Cycle

NDT was born from exhaustive comparative studies between Web proposals. NDT proposes a complete and detailed requirements phase which systematically obtains the analysis models of Web systems. It is an approach for the specification and analysis of Web information systems.

The NDT development process can be defined as a bottom-up process. The development process is focused on a very detailed requirements definition, guided by objectives, which covers three subphases: *requirements capture*, *requirements definition*, and *requirements validation*. NDT only covers the first phases in the life cycle. It is also necessary to emphasize that workflows in NDT, which go from requirements to analysis, are systematic. These workflows are defined using the MDE paradigm, as will be presented later in this paper. The necessity of offering a systematic process in order to develop Web design models has been detected by several research groups. These workflows may even be automatic if the development team uses its associated tool of NDT, the NDT-Tool.

The process starts by defining objectives. Using the procedure described below, requirements are captured and defined. Requirements are classified according to their nature: *information storage requirements*, *actors' requirements*, *functional requirements*, *interaction requirements*, or *nonfunctional requirements*. In this way, NDT follows the idea of concept separation used by other approaches in the design phase.

These requirements are described in NDT using some especially defined patterns. A pattern is a template with specific fields which must be completed by the developer. In Section 3.2, an example of a pattern and its connection to the metamodel is introduced. When requirements are validated, the NDT process continues by defining three models:

- *The content model*, which is a class diagram. It expresses the static view of the system.
- *The navigational model*, which shows how users can navigate through the system.

- *The abstract interface model*, which shows the abstract interface of the system.

One of the main contributions of the NDT process is that it offers a systematic way of obtaining analysis models from requirements, thereby making each model independent. It is systematic because NDT offers transformations which indicate how each model must be obtained from the requirements definition. In NDT, different models are related to one another since each of them represents a different aspect of the same system. However, they are independent because each one can be obtained independently from the requirements. When the development team applies these transformations, they obtain the basic models: the *basic content model*, the *basic navigational model*, and the *basic abstract interface model*.

These basic models can be modified in order to obtain models more suitable to the system. In this way, they obtain the *final content model*, the *final navigational model*, and the *final abstract interface model*. The step from basic models to final models is not automatic because it requires the analysts' experience to be applied. However, NDT offers some guidelines and heuristics for their generation.

The navigation and the content models are class diagrams. However, the abstract interface model is a set of evaluated prototypes. Users and customers can evaluate these prototypes. This is the last phase in the NDT life cycle. Starting with developed models and assuring that they are correct, the development team can apply other Web methodologies, such as UWE and OOHDM, to deal with other later phases, such as design and implementation.

The main objective of NDT is to offer systematic processes to build models which other Web methodologies can use as the starting point of their development process and to guarantee the quality of these models. Thus, we can conclude that NDT is not a complete methodological Web approach. It is a methodological procedure to obtain content, a navigational and an abstract interface model from the users' requirements. This methodological procedure is based on the MDE paradigm. It allows consistency between requirements and concepts in the analysis phase to be assured.

To conclude this short presentation, in Fig. 2, an activity diagram describes the NDT development process. In this diagram, two special stereotypes are included. The first, *QVTTransformations*, presents the QVT Transformations defined by NDT in order to attain basic analysis models from requirements. The second, *NDTSupport*, shows that these activities are supported by guidelines and heuristics offered by NDT for the development of the final models while maintaining consistency with requirements.

### 3.2 NDT Metamodel

As presented in the previous section, NDT offers systematic ways to advance from the requirements model to the analysis models. These systematic ways are possible with the use of the Model-Driven paradigm.

Following the architecture defined by MDA, the requirements artifacts of NDT are defined using a Computational Independent Model (CIM). In the analysis phase, a group of Platform Independent Model (PIM) is also defined. With a

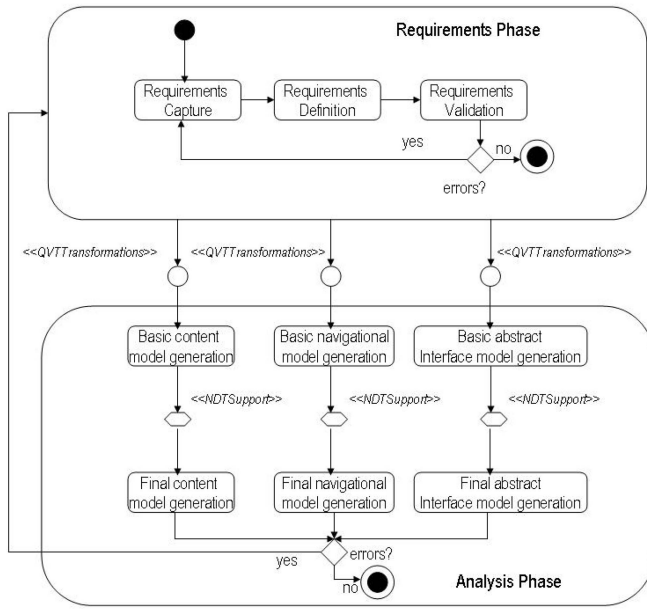


Fig. 2. NDT development process.

group of formal transformations, the group of metamodels in the PIM level can be systematically derived from the metamodel in the CIM level.

In classic Web engineering, ideas detected in the requirements phase must be translated into concepts by

analysts in the analysis models. An analyst's experience is essential and critical in this step. In the MDWE paradigm, the definition of the process is different. In each phase of the life cycle, concepts are studied, detected, and defined using a metamodel. Concepts in each metamodel are studied and related to other concepts in other metamodels. Thus, for instance, in the requirements phase of NDT, the concept storage information requirement is defined and, in the analysis phase, the concept content class is detected. Storage information requirements defined by the user are translated in a conceptual class in the analysis content model. These rules and relations allow the definition of the set of systematic transformations.

Transformations not only ease the translation of concepts but also assure the consistency between phases in the life cycle. Thus, using the same example, each storage information requirement has to be translated into a class in the analysis. If a storage information requirement is not represented by a class in the analysis content model, a requirement is lost in the development process.

In the first version, NDT metamodels were not MOF metamodels and transformations were not based on any formal language for transformations. However, NDT metamodels are currently defined using a UML profile and transformations are being defined using QVT.

The proposed metamodel for NDT in the requirements phase is presented in Fig. 3. This requirements metamodel follows the structure of WebRe [16]. As shown in Section 2.2,

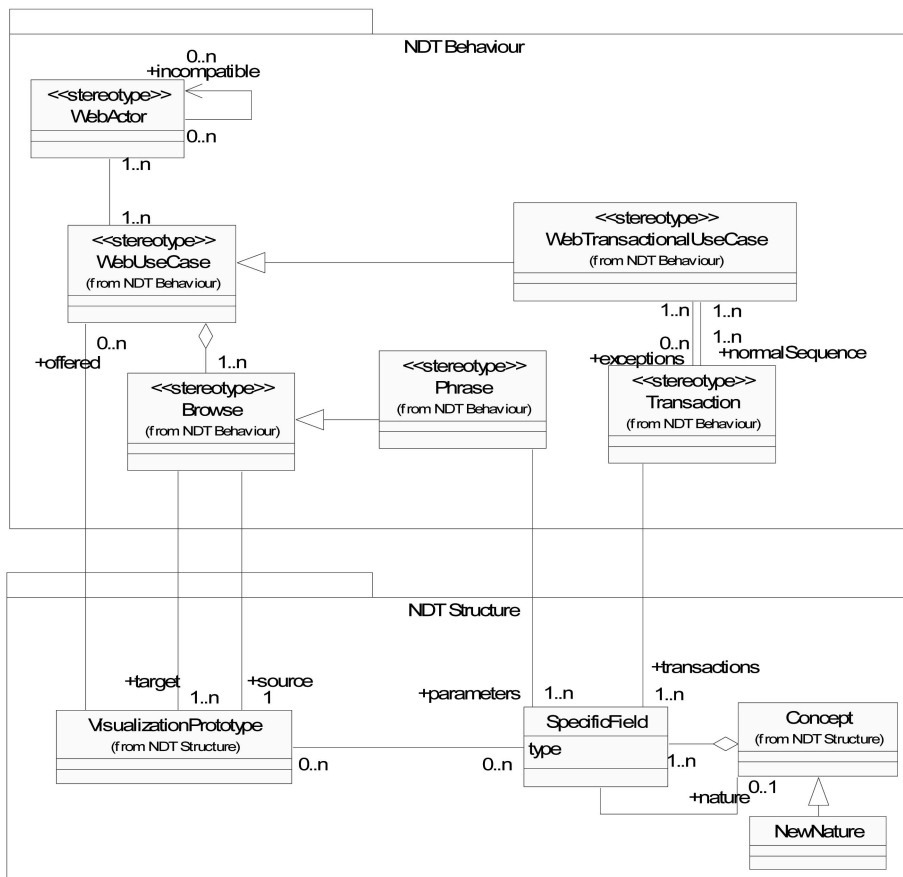


Fig. 3. NDT requirements metamodel.

WebRe is an approach for metamodels for requirements in Web engineering which groups concepts of different approaches: NDT, W2000, UWE, and UIDs of OOHDM.

However, in this case, this metamodel has been extended in order to support all of the aspects of the NDT requirements phase. The metamodel is divided into two packages: the *Behavior* and the *Structure*.

In the first package, concepts related to the behavior of the system are presented and classes represent conduct aspects. Thus, the *WebActor* class represents any actor in the system. As stated earlier, the study of actor requirements is a phase of NDT. In NDT, the study of actors is very complex. It not only detects different users in the system but also studies their relations. For this reason, in the metamodel, the relation of incompatibility is included. Two actors are incompatible when their roles cannot be played at the same time for the same person.

Another class in the behavior package is the *WebUseCase* Class. This class represents the group of functional requirements of the system. When a functional requirement includes any transactional activity, this functional requirement is an instance of *WebTransactionalUseCase*.

Finally, in the behavior package, a group of classes that represent activities is included. These are *Browse*, *Phrase*, and *Transaction*. *Browse* represents navigation activities in the system without any transaction activity. *Phrase* is a kind of activity that defines queries and search activities. Finally, the *Transaction* class represents activities where transactions are executed.

In the structure package, *Concept* class represents any information storage by the system. Each *Concept* is composed of a group of *SpecificFields*. Each *SpecificField* defines each piece of information stored by the *Concept*.

NDT defines a special concept named *NewNature*. *NewNature* is a kind of *Concept* that represents global data structures in the company, not for the system. For instance, the definition of personal data of users must be stored in the same way in each system in the company. Thus, the structure of this information is of a new nature.

The last concept, *VisualizationPrototype*, represents any point of interaction between the user and the system.

This metamodel is based on UML artifacts. For this reason, its profile can be defined. In Fig. 4, this profile is presented.

Although these concepts could be represented with UML artifacts, the selected technique in NDT for the definition of requirements is patterns. Patterns are tables with specific fields for the collection of any piece of information specific for each concept.

In NDT, the structured way to define requirements offered by the patterns is important. In Table 2, an instance of an NDT pattern is presented. This pattern represents the definition of a storage information requirement taken from a real project defined completely in [57]. It represents the necessity of storing information about patients in a medical system.

In this pattern, special fields which are necessary for the definition of storage requirements are found. The first is the identifier of the requirements, RA-01, and its name. The *associated objectives* field stores the objectives of the system

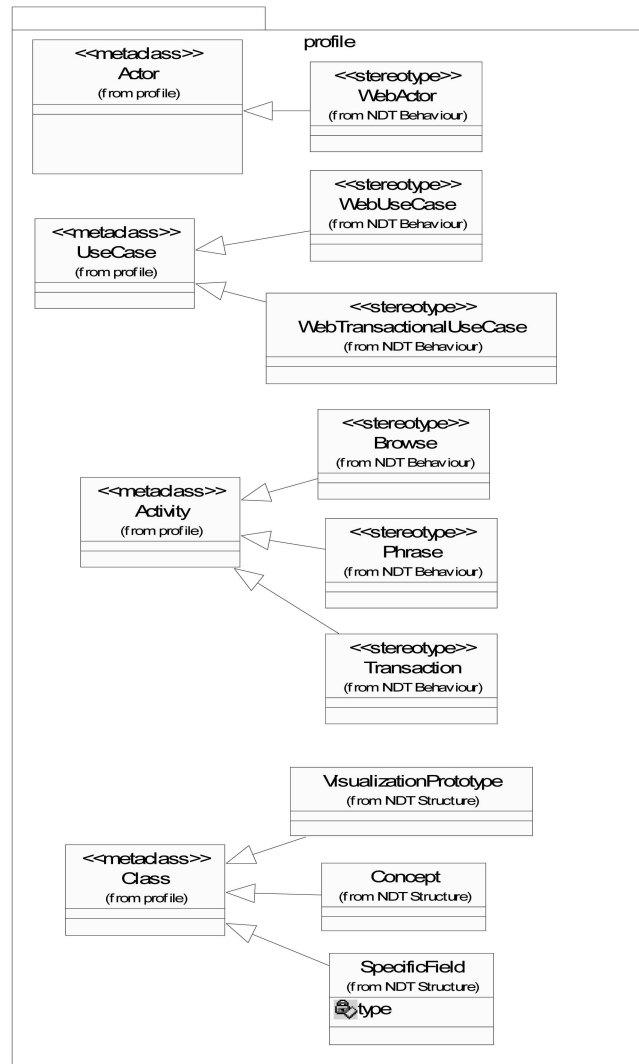


Fig. 4. NDT requirements profile.

that are later obtained (or partially obtained) by the implementation of the requirement. Finally, in the field of *specific fields*, the specific piece of data storage for each patient is presented. Similar patterns are defined for each kind of requirement in the project.

Patterns are essential throughout the life cycle of NDT. They offer a structured definition of requirements, very useful for analysts during the analysis phase. Moreover, they also offer a suitable definition for users. Patterns are completely defined using the user's language and, therefore, can be easily evaluated.

Furthermore, patterns offer a suitable way to represent each concept and relation in the metamodel. Thus, storage information requirements are instances of the *Concept* class. With the field *Specific Field*, the relation between each *Concept* and *Specific Field* is represented.

In fact, in NDT, patterns are defined in a more generic way. In Table 3, the pattern followed to define the previous example is presented. In the pattern, some fields are obligatory and others are optional (they are presented by \*). In addition, some sentences are fixed and others (sentences

TABLE 2  
An Instance of the Storage Requirements Pattern

<b>RA-01</b>	<b>Patient's basic data</b>	
<b>Associated Objectives</b>	OBJ-01: To manage the degree of handicap automatically.	
<b>Description</b>	The storage information has to be confidential and is controlled by a high security system.	
<b>Specific Fields</b>	<b>Name and description</b>	<b>Nature</b>
	Name: patient's name	String
	DNI: patient's identification number	String
	Date of Birth: the date when the patient was born	String
	Phone number: the patient's contact phone number	Cardinal Number: 1..n
	E-mail: patient's email	String
	Weight: patient's weight	Real

TABLE 3  
A Storage Requirements Pattern

<b>&lt;identifier&gt;</b>	<b>&lt;requirements name&gt;</b>	
<b>Associated Objectives</b>	<the group of objectives that are covered partially or totally by the requirements>	
<b>Description</b>	<storage information requirements definition>	
<b>Specific Fields</b>	<b>Name and description</b>	<b>Nature</b>
	<name of the specific field>: <description of the specific field>	<nature> Cardinality: <cardinality>
	...	...
<b>Comments*</b>	<some additional comments for the requirements>	

between <> ) have to be completed by the development team.

In the next phase of NDT, the analysis phase, more metamodels are used. However, most approaches of Web engineering offer models and techniques for the analysis phase. For this reason, NDT does not offer new models and uses previous experiences based on standards. For the content model, NDT proposes the development of a class diagram following UML. For this reason, the one content metamodel of NDT is the UML class metamodel.

For the navigational model and abstract interface model, NDT also looked for previous suitable metamodels. After some comparative studies [15], [18], NDT assumes the UWE metamodel for the analysis phase. The selection of UWE is based on the use of UML. UWE metamodels are completely based on UML and they have a suitably defined profile. In [32], these metamodels are presented and defined.

### 3.3 NDT Model-Driven Transformation

Continuing on from the previous section, requirements are located in the CIM level and the analysis artifact in a PIM level. Between them, transformations defined with QVT are defined in order to derive analysis models from the requirements models.

According to the taxonomy of transformation approaches proposed by Czaenecki and Helsen [6], NDT defines a set of transformations *model-to-model*. Since NDT

only works with the requirements and the analysis phases, NDT does not cover the other kind of transformations defined in this taxonomy: *model-to-code*.

QVT Transformations offered by NDT are grouped into three transformations:

- *Requirements2Content*: This transformation allows the generation of the basic content model from the requirements model.
- *Requirements2Navigational*: This transformation allows the generation of the basic navigational model from the requirements model.
- *Requirements2Prototypes*: This transformation allows the generation of the basic abstract interface model from the requirements definition.

Presenting all of these transformations in this paper is very complex. For this reason, only the first one is presented, see Table 4.<sup>4</sup>

It expresses how each storage information requirement is to be translated into a content class in the analysis phase. Each storage information system generates a class and each specific field generates an attribute.

In relation R1, each Concept and each NewNature in the requirements is translated into a Class in the Content

4. Complete information about NDT and its transformations can be found at [www.lsi.us.es/ndt](http://www.lsi.us.es/ndt).



TABLE 4  
Requirements2Content Transformation

```

Transformation Requirements2Content (ndt:NDT, uml: UML){
top relation R1 {
    checkonly domain ndt c:Concept {name = n};
    enforce domain uml cl: Class {name = n};
    relation R2 {
checkonly domain ndt sf : SpecificField
{namespace = cp:Concept {}, name = sfn};
enforce uml at: Attribute
{namespace = cc: Class {}, name = atn};
where {sf.nature->size () = 0; }
when {R1(cp, cc);}
relation R3 {
checkonly domain ndt sf : SpecificField
{namespace = cp:Concept {}, name = sfn, nature = conc:Concept {} };
    enforce uml as: Association
{name = asn};
    where {asn.connection = {cp, conc};}
    when {R1(cp, cc);}
}
}
}

```

Model. In relations R2 and R3, SpecificFields are treated. If a SpecificField has a basic Nature, that is, when its relation with Content has cardinality 0, they are translated into Attributes with relation R2. If its nature is Content, it is translated into Association with relation R3.

Similar structures can be defined for navigational and abstract interface models.

These transformations can be implemented by suitable tools such as SmartQVT<sup>5</sup> or Moment [47], as shown later in Section 5. These transformations were presented in Fig. 2 as QVTTransformation activities, although NDT includes another support also represented in Fig. 2: NDTSupport.

These transformations could be applied automatically. However, the analysts' experience could add some ideas to basic models in order to make them more suitable. For this reason, in NDT, analysts can change basic models, but these changes must be controlled. For instance, if, in the basic model, the analyst introduces a heritage, it must not contradict the transformation. However, if a new class is added, the analyst must consider if it is a lost storage requirement.

Thus, NDT offers a set of rules that must be considered by the development team to control the consistency between final models and requirements.

In [14], the complete list of these rules is defined.

In conclusion, NDT is a Web approach focused on the first phases of the life cycle. It tries to offer a Model-Driven development process in order to assure the consistency of the model and to ease the development of each model in each phase. It is completely oriented to the interaction between the development team and the user with the use of patterns and its development process is supported by a tool named NDT-Tool. In this sense, NDT attempts to cover a gap in Web engineering by offering a detailed requirements approach that is compatible with other accepted approaches, mainly UWE.

### 3.4 NDT-Tool

Another important gap detected in Web engineering is the lack of tools to support its development. NDT and its

definition based on metamodels and transformations can be translated into a tool, named NDT-Tool. One of the most important advantages of NDT is that its tool not only support the definition of models or results, like other tools in Web engineering, it supports the complete life cycle of NDT, it lets patterns and model be defined, and it guarantees the traceability using model-driven support. It has been shown that NDT is being used profusely in practice, as is presented in Section 4.

This tool is developed with J2EE using the Model-View-Controller pattern. Its architecture is presented in Fig. 5.

The application interface, NDT-Tool, receives requests from the user and translates them to the controller. It interacts with the model, developed using JavaBeans and EJBs. Furthermore, the controller sends requests to the View, modeled using JSP and TagLibs. The view makes some queries to the model and presents the results to the user using the interface.

Nowadays, there is a special collaboration between two companies, *Telvent* and *everis*, to finish the new tool that was presented at the end of September 2007. NDT-Tool is very suitable for applying NDT because it supports its complete life cycle. It allows the attainment of all NDT results automatically: requirements and analysis documents and, also, system prototypes. Not only are transformations between requirements and analysis models completely implemented in NDT-Tool, but the heuristics and controls of NDT also support the development of final models in the analysis phase.

NDT-Tool is a free tool.<sup>6</sup> However, it is not the only option for NDT. It is an ad hoc tool. For large projects or special adaptations of NDT, this tool can be considered to be too closed. For instance, in a real project developed with NDT named Diraya [19], NDT-Tool was not useful because this project was very large and it needed a special identification of requirements and modularization of the system that was impossible to achieve using only NDT-Tool. The definition of NDT using metamodels and profiles as a formal extension of UML allows the adaptation of any commercial tool that supports the definition of profiles into the NDT approach. Thus, in the Diraya Project, Enterprise Architecture was the solution selected, as seen below.

### 3.5 NDT in the Web Engineering Community

To conclude with the NDT presentation, it is necessary to present how NDT covers most of gaps presented in Section 2:

- About the life cycle, NDT is focused on requirements and analysis. Thus, it covers the gap in the life cycle presented in Web engineering.
- About the compatibility and standards, NDT is compatible with other approaches, mainly UWE and approaches based on standards like UML.
- About the tool support, NDT offers NDT-Tool, a case tool that supports not only the generation of results but also the complete life cycle and the traceability using model-driven paradigm.

5. <http://sourceforge.net/projects/smartqvt>.

6. Available at [www.lsi.us.es/ndt](http://www.lsi.us.es/ndt).

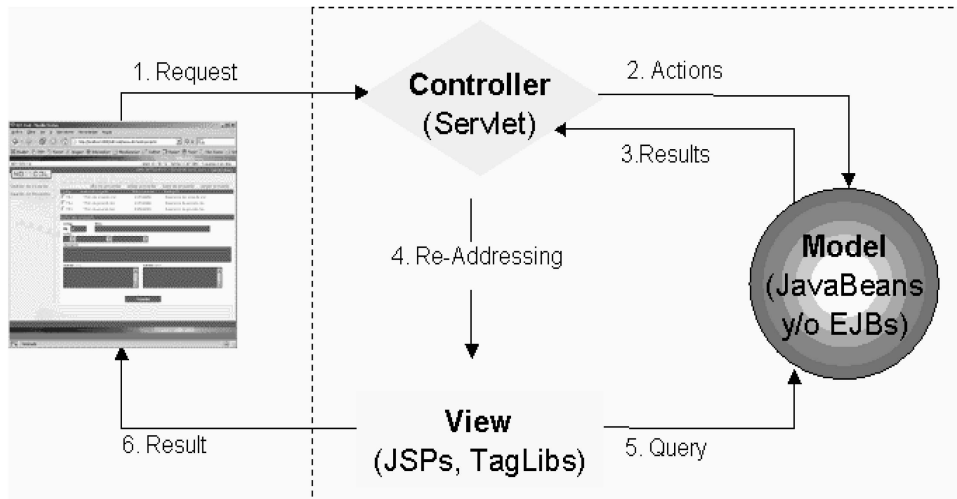


Fig. 5. NDT-tool adaptation.

TABLE 5  
The Project Analyzed

Name of project	Date	Client
<i>Systems to manage information on grants</i>		
The system to manage information on cultural grants.	2000-2001	Andalusian Government
The system to manage information on international aid.	2001-2002	
<i>Systems to manage historical patrimony information</i>		
The system of Movable Heritage in Andalusia.	1997-1999	Government of Andalusian Culture
The Thesaurus of Historical Heritage.	2003-2004	
The system of Historical Heritage Authors.	2005-2006	
<i>Medical Systems</i>		
The system to measure the grade of handicap.	2004-2005	Alcer Foundation
Specialized Diraya Project	2006-...	Andalusian Health Service

- About the automation of processes, NDT proposes the use of the model-driven paradigm and it reduces the development time using NDT-Tool.

However, there is another important gap that has not yet been treated. This is the practical application. NDT has been a very used approach; in fact, it is being used by several companies and development teams. Its practical applications have influenced the methodology definition, as is presented in Section 4.

#### 4 NDT PRACTICAL EVOLUTION

Following the introduction to the theory of NDT, which can be seen in more depth in its reference manual [14], the historical evolution of NDT is presented in this section.

NDT was created from comparative and analytic studies of Web engineering. One of the conclusions we have drawn from these studies stands out above all the others: that Web engineering has been greatly underemployed in the enterprise environment.

For this reason, the feedback obtained for NDT using practical experience is an important factor in our research work. In Table 5, the most relevant practical experiences for NDT are numbered.

Although the projects presented here are grouped by theme, it is important to analyze the order in which they were developed to see the evolution of NDT. In Table 5, an abstract with the name of each project, the date of development, and the client company is shown. This short schema provides the base for the general analysis of the conclusions.

##### 4.1 Grant Management Projects

When NDT was first applied in 2000, only the requirements metamodel was defined. Patterns and models in the requirements phase were applied in a system to manage the information on grants given by the Andalusian Government for cultural activities. NDT patterns were changed, corrected, and modified during this first collaboration. NDT divides requirements into five groups: storage information, actors', functional, interaction, and

nonfunctional requirements. The system developed was of critical importance because it had to manage a great amount of information that depended on a complex administrative process where there were a high number of different roles with different necessities. To this end, it was deemed necessary to offer a way to define new data structures. However, the idea of data structure was not oriented to a programming concept.

The data structures that had to be included in NDT had to be oriented to the user. They had to offer a suitable way to define a set of structured data according to the user's view. For instance, the necessity of storing the identification data of different people, users, managers, etc., appeared several times in the system. These data always included the same information: name, address, etc. Instead of defining these data every time, we worked on the idea of offering a way to define this structure in the requirements phase.

With this idea, in NDT, the concept of new natures was introduced. A nature is a new special kind of requirement, with its own pattern, that allows the definition of these data structure requirements.

Another important idea that was added in NDT during the development of this project was the use of graphics. At the beginning, the requirements phase of NDT was only based on patterns. However, we detected that, at the beginning of the project, it was more interesting to organize meetings based on graphical notations. We decided to enrich the basic definition of NDT with UML use cases. The use case model is a standard notation widely known and easily understood by nonexperts. The textual description of patterns was kept in NDT since use cases became very ambiguous for the capture of all the necessary information.

These changes in NDT provoked an additional cost in the project, which was subsequently delayed by several months. However, the final results were good and, some months later, a new similar collaboration was initiated.

Again, the project was a system to manage information on grants in the Andalusian Government. However, in this case, grants were oriented to international aid. The users were not the same and the new users did not know NDT or the patterns. Nevertheless, with this new experience, it was shown that the changes introduced were very suitable.

This new double way of presentation, involving both patterns and diagrams, offered a more flexible way of working with users. However, an important barrier to the application of NDT was found in this collaboration; keeping patterns updated was a very difficult task. Patterns were interrelated and a single change or modification in one pattern could produce changes in several others. It was at this point that the necessity of developing a tool to support NDT was detected.

## 4.2 Heritage Management Projects

The oldest project where NDT was applied was the system of Movable Heritage in Andalusia (1997-1999). This system allows the management and distribution of information on movable heritage in Andalusia. The initial patterns in NDT were completed and first tested here. However, the essential contribution of this project to NDT was the special treatment of the different roles.

This system took into account the different user roles and the system changed completely depending on each role. The information shown was different depending on whether the user was an archaeologist, an artist, a tourist, or other. Moreover, the same user could navigate in the system playing several roles at the same time. Thus, if a user was an archaeologist, an artist, or an archaeologist-artist, the navigation, functionality, and interface were completely different.

The complexity of this role motivated us to find simple but powerful ways to study roles and their relationships. From this idea, new models based on inheritance and traceability matrices were added to NDT during this project.

These new ideas were again tested in a similar project with the same group of users in 2003. In this case, we applied NDT to develop a Thesaurus of Historical Heritage<sup>7</sup> and obtained very good results.

This project also demonstrated the advantages of working with users that know the NDT patterns. Patterns are very intuitive for users and nonexpert people in software engineering. When a user works with a pattern, it is very easy for him/her to use it again.

Recently, the evolution of NDT and its practical improvements have been appreciated in a newly completed project with the same group of users. This new project consists of the development of a system to manage and distribute, through the Internet, information about authors who worked on the Andalusian Historical Heritage. Both development time and cost are lower in this project than in the other projects for several reasons.

On one hand, the users and the development team knew the development environment very well. Since the users understood NDT well, it was quite easy to apply patterns and requirements techniques. On the other hand, NDT-Tool had already been fully developed and was available via the Web, making the management of the information and the attainment of the results easier.

Good results obtained in these collaborations and other similar projects with the Cultural Andalusian Government ensured that NDT was chosen as the methodology to be applied in all its Web projects.

Today, a special collaboration between the University of Seville and the Culture Andalusian Government controls the application of NDT to each project. At the moment, 11 projects are being developed using the NDT approach.

## 4.3 Medical Systems

Two important medical systems have been developed with NDT.

The first one, a system to measure the grade of handicap, was one of the first where NDT was applied [57]. In this case, NDT was a consolidated methodology. For this reason, in the application of NDT to this system, patterns, models, techniques, and NDT-Tool could be applied without any changes and very good results were obtained. Moreover, the project was essential for another aspect which had been forgotten in the research environment but was essential in some enterprise environments.

7. [www.juntadeandalucia.es/iaph](http://www.juntadeandalucia.es/iaph).

The system to manage the grade of handicap is a system developed with the Alcer Foundation.<sup>8</sup> This system enables the application of the Royal Decree 1971/1999 (23/12) of the Spanish Government and dictates how the grade of handicap of a patient must be measured by a medical tribunal.

This project offered two important challenges to NDT. First, the environment, the group of users, and the terminology were completely new for NDT and the development team since there is no similar system in Spain. In this sense, the application of NDT was very successful because it eased communication with users.

Furthermore, since this line of research is a relatively new concept, interest was generated. In other projects, the validation of requirements was made with techniques such as reviews, traceability matrices, and the study of the terminology with glossaries.

Nevertheless, this system worked in a specific environment with a very complex argot. The ambiguity of the terminology in this kind of project could cause serious administrative disasters.

Work was started on the development of more powerful techniques to validate requirements. We made a comparative study and worked with the University of Nice on this aspect. The university developed a tool, named fuzzy thesaurus [39], which allows the development team to test ambiguities in software analysis models. After a joint study, this tool was adapted to NDT and it is now being included in NDT-Tool. The application of the fuzzy thesaurus enables any errors in the requirements definition provoked by terminology ambiguities to be found in a systematic way. The application of this tool is not always enforced in NDT, but it is a good technique for the validation of requirements in complex projects with complex environments.

However, if a project has to be selected as the most complicated for NDT, it would have to be the Specialized Diraya Project.

The Diraya Project is a very ambitious project developed by the Andalusian Health Service (SAS, in Spanish).<sup>9</sup> SAS is the public administration agency in Andalusia which manages hospitals, health centers, and other public health systems.

Some years ago, the Diraya project started to manage Andalusian primary health attendance and was called Primary Diraya. However, it is now being extended to include specialized health attendance and is consequently called Specialized Diraya.

Specialized Diraya is a Web system to manage all of the information on patients that visit any hospital in Andalusia, independently of whether it is through a specific appointment or in an emergency situation.

Specialized Diraya will be installed in 29 hospitals in Andalusia and will be used by more than 62,000 users composed of doctors, nurses, etc. The complete health information in Andalusian hospitals will be managed by this new system.

The project started some months ago and it is being developed by six large software companies in Andalusia: Telvent, Indra, Everis, Tecnova, Accenture, and Isoft. These

companies are working together in order to attain the best results. These are experts in healthcare systems and their prior knowledge is being pooled during the development process.

Specialized Diraya development has been divided into two phases. The first phase was presented in June 2007 and is being installed in a hospital for its evaluation.

Although Specialized Diraya development is based on Métrica,<sup>10</sup> a methodology proposed by the Spanish government for public software projects, Métrica and NDT are compatible. Consequently, NDT was the environment used in the requirements and in the analysis phase.

The group of analysts is composed of 13 people from the different companies who are now working together in order to obtain a consistent product. The number of members of the development team, including people from companies, analysts, designers, SAS software experts, etc., exceeds 40.

The magnitude of Diraya is a very interesting example. The high number of users, analysts, designers, etc., the high number of requirements, and the high number of different roles in the development form a complex and real example that offers important feedback for our research results.

The application of NDT to this project is very successful, as presented in [19]. The Model-Driven paradigm used in NDT transformations reduced the development time since it was a systematic translation. Moreover, some inconsistencies were found among the workings of each company.

However, a new handicap for NDT was detected in this project. Although NDT-Tool is a suitable tool for NDT, it is an ad hoc tool and, hence, when a project requires special adaptation, such as ids configuration, and special patterns to generate documents, NDT-Tool becomes an insufficient tool to fulfill these adaptations.

For this reason, SAS and the University of Seville were working in order to find a suitable tool for Specialized Diraya. After looking for and comparing different possibilities, Enterprise Architecture was selected.

Enterprise Architecture supports UML and offers the possibility of extending the initial definition of UML with its extension mechanisms. The NDT requirements metamodel and also the UWE analysis model are defined with a UML profile and, therefore, it was very easy to adapt the tool for the group. With Enterprise Architecture's profile option, the profile of NDT was defined and companies used this profile to define requirements and analysis artifacts.

More recently, the first release of this project concluded with such good results that SAS followed the decision of the Government of Andalusian Culture and has implanted NDT in all of their Web projects.

## 5 CONCLUSIONS AND FUTURE WORK

Web engineering is currently a new and an important research line in software engineering. Some questions asked years ago on whether the Web environment really needed special processes, methods, and techniques [33] are not posed today. Web engineering has become an accepted research environment, although important gaps are still present in this line of research.

8. [www.alcer.info](http://www.alcer.info).

9. [www.juntadeandalucia.es/sas](http://www.juntadeandalucia.es/sas).

10. [www.map.es](http://www.map.es).

The lack of standard notation, the high number of approaches, models, and techniques, the necessity of offering tools to support the development, and the lack in the study of Web requirements are only a handful of ideas enumerated from comparative studies in this paper.

In order to cover some of these gaps, in this paper, NDT was shown. NDT is a specific approach for the requirements treatment. It is based upon the use of UML and other standard approaches, such as QVT. Furthermore, it proposes the use of metamodels to make itself compatible with other suitable techniques, such as UWE, and it comes with a tool, NDT-Tool, which permits the application of its complete life cycle results to be automatically processed and the automation of its procedures and transformations.

NDT also follows some general ideas accepted by the Web engineering community. Thus, it proposes dividing the treatment of requirements in accordance with the idea of concept separation, which is followed by most of the other Web approaches in the design phase.

The lack of practical experience in Web engineering is also covered by NDT. It has been and is now being applied in real projects, thereby generating important feedback for its development and evolution, as presented here.

However, NDT is still under development and this research is being continued with our future work.

In the theoretical line, NDT is being enriched with a new approach to the derivation of test cases from its requirements model. The requirements metamodel of NDT is being analyzed and reviewed in order to incorporate new artifacts which support the automatic generation of test cases. Important results in this area have been obtained and can be consulted in [25].

Another important line of research emerged from the Diraya Project. The use of a tool is essential for Web systems. NDT-Tool is very good support, but it is also too closed in some cases. Companies and Web projects have special necessities and corporative rules; thus, it is necessary to offer different possibilities. With the Diraya Project, we adapted Enterprise Architecture to support NDT metamodels where the use of profiles is essential. Thus, other tools such as Rational Rose or StartUML, which support UML profile definition, are easily adapted. However, they do not support QVT languages in their current versions and the definition of NDT transformations is very complex.

For this reason, in order to define suitable tool environments for NDT, we are looking at using tools such as ATL, SmartQVT, or Moment, which are based on transformation languages.

Finally, one of our most important lines of work is to continue with practical collaboration. The Government of Andalusian Culture and SAS have assumed NDT as their requirements and analysis approach. The continued application of NDT is fundamental to verifying research results.

## ACKNOWLEDGMENTS

The authors wish to thank Lesley Burrige and Rafael Ordoñez for helpful suggestions and English comments. This work was supported in part by project QSimTest of the Ministerio de Educación y Ciencia (TIN2007-67843-C06\_03), Spain and by the RePRIS project of the Ministerio de Educación y Ciencia (TIN2007-30391-E), Spain.

## REFERENCES

- [1] L. Baresi, F. Garzotto, and P. Paolini, "Extending UML for Modelling Web Applications," *Proc. 34th Ann. Hawaii Int'l Conf. System Sciences*, pp. 1285-1294, 2001.
- [2] L. Baresi, F. Garzotto, and M. Maritati, "W2000 as a MOF Metamodel," *Proc. Sixth World Multiconf. Systemics, Cybernetics, and Informatics*, 2002.
- [3] C. Barry and M. Lang, "A Survey of Multimedia and Web Development Techniques and Methodology Usage," *IEEE Multimedia*, pp. 52-56, Apr.-June 2001.
- [4] M. Bieber, R. Galnares, and Q. Lu, "Web Engineering and Flexible Hypermedia," *Proc. Second Workshop Adaptive Hypertext and Hypermedia*, 1998.
- [5] C. Cachero, "Una Extensión a los Métodos OO para el Modelado y Generación Automática de Interfaces Hipermediales," PhD dissertation, Univ. of Alicante, 2003.
- [6] K. Czaeński and S. Helsen, "Classification of Model Transformation Approaches," *Proc. OOPSLA Workshop Generative Techniques in the Context of Model-Driven Architecture*, 2003.
- [7] S. Ceri, P. Fraternali, and P. Bongio, "Web Modelling Language (WebML): A Modelling Language for Designing Web Sites," *Proc. Ninth Int'l World Wide Web Conf., Computer Networks*, vol. 33, nos. 1-6, pp. 137-157, May 2000.
- [8] P. Chen, "The Entity-Relationship Approach: Towards a Unified Behavior of Data," *ACM Trans. Database Systems*, vol. 1, no. 1, pp. 9-36, Jan. 1976.
- [9] E.F. Codd, *The Relational Model for Database Management*. Addison-Wesley, 1992.
- [10] J. Conallen, *Building Web Applications with UML*. Addison Wesley, 1999.
- [11] O. De Troyer and C. Leune, "WSDM: A User-Centered Design Method for Web Sites," *Proc. Seventh Int'l World Wide Web Conf., Computer Networks and ISDN Systems*, pp. 85-94, 1998.
- [12] Y. Deshpande, S. Marugesan, A. Ginige, S. Hanse, D. Schawabe, M. Gaedke, and B. White, "Web Engineering," *J. Web Eng.*, vol. 1, no. 1, pp. 3-17, 2002.
- [13] M.J. Escalona, J. Torres, M. Mejías, and A.M. Reina, "NDT-Tool: A Tool Case to Deal with Requirements in Web Information Systems," *Proc. Fourth Int'l Conf. Web Eng.*, pp. 212-213, 2003.
- [14] M.J. Escalona, "Modelos y Técnicas para la Especificación y el Análisis de la Navegación en Sistemas Software," PhD dissertation, Univ. of Seville, 2004.
- [15] M.J. Escalona and N. Koch, "Requirements Engineering for Web Applications: A Comparative Study," *J. Web Eng.*, vol. 2, no. 3, pp. 193-212, 2004.
- [16] M.J. Escalona, J.J. Gutierrez, D. Villadiego, A. León, and A.H. Torres, "Practical Experience in Web Engineering," *Proc. 15th Int'l Conf. Information Systems Development*, 2006.
- [17] M.J. Escalona and N. Koch, "Metamodelling the Requirements of Web Systems," *Proc. Second Int'l Conf. Web Information Systems and Technologies*, pp. 310-317, 2006.
- [18] M.J. Escalona, J. Torres, M. Mejías, J.J. Gutiérrez, and D. Villadiego, "The Treatment of Navigation in Web Engineering," *Advances in Eng. Software*, vol. 38, pp. 267-282, 2007.
- [19] M.J. Escalona, C.L. Parra, F.M. Martín, J. Nieto, A. Llergo, and F. Pérez, "Diraya Project. The Power of Metamodels in Real Experiences with Web Engineering," *Proc. 16th Int'l Conf. Information Systems Development*, 2007.
- [20] J.L. Fernández and A. Monzón, "A Metamodel and a Tool for Software Requirements Management," *Reliable Software Technologies, Ada-Europe*, 2000.
- [21] J. Fons, V. Pelechano, M. Albert, and O. Pastor, "Development of Web Applications from Web Enhanced Conceptual Schemas," *Proc. 22nd Int'l Conf. Conceptual Modeling*, pp. 232-245, 2003.
- [22] F. Garzotto, D. Schwabe, and P. Paolini, "HDM-A Model-Based Approach to Hypermedia Application Design," *ACM Trans. Information Systems*, vol. 11, no. 1, pp. 1-26, 1993.
- [23] H.W. Gellersen, R. Wicke, and M. Gaedke, "WebComposition: An Object-Oriented Support System for the Web Engineering Lifecycle," *Computer Networks and ISDN Systems*, vol. 29, pp. 1429-1437, 1997.
- [24] A. Gu, "Extending Object-Oriented Modelling Languages for Web Applications," MSc thesis, Univ. of Technology, 2001.
- [25] J.J. Gutierrez, M.J. Escalona, M. Mejías, and J. Torres, "Derivation of Test Objectives Automatically," *Proc. 15th Int'l Conf. Information Systems Development*, 2006.

- [26] E. Insfrán, O. Pastor, and R. Wieringa, "Requirements Engineering-Based Conceptual Modelling," *Requirements Eng. J.*, vol. 7, no. 1, 2002.
- [27] T. Isakowitz, E. Stohr, and P. Balasubramanian, "RMM: A Methodology for the Design of Structured Hypermedia Applications," *Comm. ACM*, vol. 38, no. 8, pp. 34-44, 1995.
- [28] G. Kappel, B. Pröll, W. Retschitzegger, and W. Schwinger, "Modelling Ubiquitous Web Applications—The WUML Approach," *Proc. Int'l Workshop Data Semantics in Web Information Systems*, 2001.
- [29] N. Koch, "Software Engineering for Adaptive Hypermedia Applications," PhD dissertation, FAST Reihe Softwaretechnik, vol. 12, Uni-Druck Publishing, 2001.
- [30] N. Koch, G. Zhang, and M.J. Escalona, "Model Transformations from Requirements to Web System Design," *Proc. Sixth Int'l Conf. Web Eng.*, pp. 281-288, 2006.
- [31] N. Koch, "Transformations Techniques in the Model-Driven Development Process of UWE," *Proc. Second Int'l Workshop Model-Driven Web Eng.*, vol. 155, 2006.
- [32] A. Kraus and N. Koch, "A Metamodel for UWE," Technical Report 0301, Ludwig-Maximilians-Universität München, Jan. 2003.
- [33] M. Lang, "Hypermedia System Development. Do We Really Need New Methods?" *Site—Where Parallels Intersect. Informing Science*, pp. 883-891, 2002.
- [34] D. Lange, "An Object-Oriented Design Approach for Developing Hypermedia Information Systems," *Proc. 31st Ann. Conf. Systems Science*, Sprague R, 1995.
- [35] H. Lee, C. Lee, and C. Yoo, "A Scenario-Based Object-Oriented Methodology for Developing Hypermedia Information Systems," *Proc. 31st Ann. Conf. Systems Science*, pp. 121-138, 1998.
- [36] S.W. Liddle, D.W. Embley, and S.N. Woodfiel, "A Seamless Model for Object-Oriented Systems Development," *Proc. First Int'l Workshop Web-Oriented Software Technology*, 2001.
- [37] D. Lowe and J. Eklund, "Client Needs and the Design Process in Web Projects," *Web Eng. Track of the WWW Conf.*, 2002.
- [38] S. Meliá and J. Gómez, "The WebSA Approach: Applying Model-Driven Engineering to Web Applications," *J. Web Eng.*, vol. 5, no. 2, 2006.
- [39] I. Mirbel, "Un Mécanisme d'Intégration de Schemas de Conception Orientée Object," PhD dissertation. Laboratory IS3, Univ. of Nice, Dec. 1996.
- [40] N. Moreno, P. Fraternali, and A. Vallecillo, "A UML 2.0 Profile for WebML Modelling," *Proc. Second Int'l Workshop Model-Driven Web Eng.*, 2006.
- [41] N. Moreno, J.R. Romero, and A. Vallecillo, "An Overview of Model-Driven Web Engineering and the MDA," *Web Eng. and Web Applications Design Methods*, chapter 12, 2007.
- [42] J. Nanard and J. Nanard, "Hypertext Design Environments and the Hypertext Design Process," *Comm. ACM*, vol. 38, no. 8, pp. 49-56, 1995.
- [43] L. Olsina, "Building a Web-Based Information System Applying the Hypermedia Flexible Process Modelling Strategy," *Proc. Workshop Hypermedia Development Processes, Methods, and Models*, 1998.
- [44] OMG: MDA Guide, version 1.0.1, <http://www.omg.org/docs/omg/03-06-01.pdf>, 2003.
- [45] OMG. Unified Modeling Language: Superstructure, version 2.0, Specification, Object Management Group, <http://www.omg.org/cgi-bin/doc?formal/05-07-04>, 2005.
- [46] O. Pastor, E. Insfran, V. Pelechano, J. Romero, and J. Meseguer, "OO-METHOD: An OO Software Production Environment Combining Conventional and Forma Methods," *Proc. Ninth Int'l Conf. Advanced Information Systems Eng.*, 1997.
- [47] P. Queralt, L. Hoyos, A. Boronat, J.A Carsí, and I. Ramos, "Un Motor de Transformación de Modelos con Soporte para el Lenguaje QVT Relations," *III Taller sobre Desarrollo de Software Dirigido por Modelos, MDA y Aplicaciones (DSDM)*, 2006.
- [48] Query QVT-Merge Group, Revised submission for MOF 2.0 Query/Views/ Transformations RFP. 2004, Object Management Group, <http://www.omg.org/cgi-bin/apps/doc?ad/04-04-01.pdf>, 2004.
- [49] W. Retschitzegger and W. Schwinger, "Towards Modeling of Data Web Applications—A Requirements Perspective," *Proc. Am. Conf. Information Systems*, vol. 1, pp. 149-155, 2000.
- [50] G. Rossi, "An Object-Oriented Method for Designing Hypermedia Applications," PhD dissertation, Univ. of PUC-Rio, 1996.
- [51] A. Schauerhuber, M. Wimmer, and E. Kapsammer, "Bridging Existing Web Modeling Languages to Model-Driven Engineering: A Metamodel for WebML," *Proc. Second Int'l Workshop Model-Driven Web Eng.*, 2006.
- [52] D.C. Schmidt, "Model-Driven Engineering," *Computer*, Feb. 2006.
- [53] P. Valderas, V. Pelechano, and O. Pastor, "A Transformational Approach to Produce Web Application Prototypes from a Web Requirements Model," *Int'l J. Web Eng. and Technology*, 1476-1289, 2006.
- [54] A. Vallecillo, N. Koch, C. Cachero, S. Comai, P. Fraternali, I. Garrigós, J. Gómez, G. Kappel, A. Knapp, M. Matera, S. Meliá, N. Moreno, B. Pröll, T. Reiter, W. Retschitzegger, J.E. Rivera, W. Schwinger, M. Wimmer, and G. Zhang, "MDWenet: A Practical Approach to Achieving Interoperability of Model-Driven Web Engineering Methods," *Proc. Third Int'l Workshop Model-Driven Web Eng.*, pp. 246-254, 2007.
- [55] R. Vidgen, "WISDM: Constructing a Web Information System Development Methodology," *Information Systems J.*, vol. 12, no. 3, pp. 247-261, 2002.
- [56] P. Vilain, D. Schwabe, and C. Sieckenius, "A Diagrammatic Tool for Representing User Interaction in UML," *Proc. Third Int'l Conf. Unified Modeling Language*, 2000.
- [57] D. Villadiego, M.J. Escalona, J. Torres, and M. Mejías, "Application of NDT to the System for the Recognition, Declaration and Qualification of the Handicap Grade," Internal Report LSI-2004-02, Univ. of Seville, 2004.