Dealing with dependencies among functional and non-functional requirements for impact analysis in Web engineering

José Alfonso Aguilar
Computer Science Faculty, University of Sinaloa, Mexico
Irene Garrigós
Department of Software and Computing Systems (DLSI), University of Alicante, Spain
Jose-Norberto Mazón
Department of Software and Computing Systems (DLSI), University of Alicante, Spain

ABSTRACT

Due to the dynamic nature of the Web, as well as its heterogeneous audience, Web applications are more likely to rapidly evolve, thus leading to inconsistencies among requirements during the development process. Importantly, to deal with these inconsistencies Web developers need to know dependencies among requirements. Furthermore, the understanding of requirement dependencies also helps in better managing and maintaining Web applications. Therefore, in this paper, an algorithm has been defined in order to analyze dependencies among functional and non-functional requirements, thus understanding which is the impact derived from a change during the Web application development process. This impact analysis would support Web developer in selecting requirements to be implemented, thus ensuring that Web applications finally satisfy the audience

Keywords: Impact analysis, goal-oriented requirements engineering, Web engineering.

INTRODUCTION

Requirements in Web engineering tend to rapidly evolve due to the dynamic nature of the Web (Ginige, 2002). This continuous evolution may lead to inconsistencies among requirements which hinder Web developers from understanding the impact of a change in the Web application. Impact analysis of requirements is therefore a crucial task in Web engineering order to know how changes in a Web requirement affect the other requirements, thus supporting Web developers in making the right design decisions.

Impact analysis is the task of identifying the potential consequences of a change, or estimating what needs to be modified to accomplish a change (Arnold & Bohner, 1993). We define a "change" as any modification on a Web requirement. Usually, impact analysis has been done intuitively by Web applications developers, after some cursory examination of the code and documentation. This may be sufficient for small Web applications, but it is not enough for

sophisticated ones. In addition, empirical investigation shows that even experienced Web applications developers predict incomplete sets of change impacts (Lindvall & Sandahl, 1998). Therefore, to effectively analyze the impact of requirements in Web applications, dependencies among requirements should be explicitly considered, thus better managing changes in Web applications. Actually, inconsistencies are defined as negative dependencies among the set of requirements, caused by the fact that requirements often originate from stakeholders with different or conflicting viewpoints (Zhang, Mei, & Zhao, 2005). Usually, impact analysis is performed only by considering dependencies on functional requirements (FR), leaving aside the non-functional requirements as shown in (Zhao, 2002) and (Gupta, Singh, & Chauhan, 2010). In the software engineering area, FR describe system services, behavior or functions, whereas NFR, or quality requirements, specify a constraint on the system or on the development process (Sommerville, 2005). According to (Ameller, Franch, & Cabot, 2010), we believe that the NFRs must be considered from the very beginning of the development process, also in the development of Web applications. Therefore, impact analysis should be done on both kinds of requirements: FR and NFRs.

Interestingly, the recent inclusion of goal-oriented techniques (Nuseibeh & Easterbrook, 2000) in Web requirements engineering, (Bolchini & Mylopoulos, 2003), (J. A. Aguilar, I. Garrigós, J. N. Mazón, & J. Trujillo, 2010), (Garrigós, Mazón, & Trujillo, 2009) offer a better analysis of requirements in Web application development, due to the fact that requirements are explicitly specified in models, thus supporting developers in evaluating the implementation of certain requirements for desigining successful software. In the goal-oriented requirements engineering field, FR are related to goals and sub-goals whereas NFRs are named softgoals, commonly used to represent objectives that miss clear-cut criteria. Specifically, finding right tradeoffs for dependent NFRs is an important step when impact of requirements is being analyzed (Boehm & In, 1996) and (Elahi & Yu, 2009), i.e., a Web application without passwords is usable, but not very secure, increased usability reduce security or increased security reduce usability. Unfortunately, finding these tradeoffs among dependent Web requirements is not a trivial task due to the dynamic nature of the Web.

Therefore, this paper presents a goal-oriented requirements engineering proposal for supporting Web developers in analyzing the impact of a change by considering both FR and NFRs in the Web applications. To this aim, an algorithm has been defined to support Web developer in (i) clearly identifying dependencies among FR and NFRs, and (ii) automatically performing the impact analysis in Web engineering by determining right tradeoffs. The main benefit of our approach is that provides information about the different design alternatives, thus allowing developers to make more informed design decisions for implementing a Web application that fully-satisfies FR, while there is a tradeoff among NFRs.

This paper is as extension of our recent work (Aguilar, Garrigós, Mazón, 2011) about the importance of considering impact analysis in a goal-oriented Web requirements analysis approach. In particular, the novelty of our ongoing work presented in this paper consists of: (i) the implementation of the UML-Profile to adapt the i* framework in the Web domain as a metamodel, (ii) the development of a prototype tool for the Web requirements specification as a proof of concept of our approach, (iii) the implementation of the transformation rules (with a high degree of automation) to derive the Web application conceptual models, and (iv) the implementation of the algorithm for impact analysis in goal-oriented models.

The remainder of this paper is structured as follows: Section 2 presents some related work relevant to the context of this work. Section 3 describes the proposal for goal-oriented requirements analysis where is found the contribution of this work and introduces a running example for demonstration purposes. The algorithm for impact analysis in goal-oriented requirements is presented in Section 4. The application of the algorithm to perform the impact analysis is described step by step in Section 5. In Section 6 is presented the current implementation of this approach. Finally, the conclusion and future work is presented in Section 7.

RELATED WORK

In our previous work (J. A. Aguilar, I. Garrigós, J.-N. Mazón, & J. Trujillo, 2010), a systematic literature review has been conducted for studying requirement engineering techniques in the development of Web applications. Our findings showed that most of Web engineering approaches focus on the analysis and design phases and do not give a comprehensive support to the requirements phase (such as OOHDM (Schwabe & Rossi, 1995), WSDM (Troyer & Leune, 1998) or Hera (Casteleyn, Woensel, & Houben, 2007)). Moreover, we can also conclude that the most used requirement analysis technique is the UML (Unified Modeling Language) use cases. This technique has proved to be successful in the common software development process to deal with the requirements specification in both textual and diagram. But unfortunately, this technique is not enough to deal with aspects such as navigation in the Web application development process. Even though, this technique is applied by some of the most remarkable Web engineering approaches such as OOWS (Pastor, Fons, Pelechano, & Abrahão, 2006), WebML (Ceri, Fraternali, & Bongio, 2000), NDT (Maria J. Escalona & Aragón, 2008) and UWE (Koch, Knapp, Zhang, & Baumeister, 2008).

Furthermore, none of the aforementioned Web engineering approaches perform the analysis and modeling of the users' needs for ensuring that the Web application is not overloaded with useless functionalities, while important functionalities are not missed. We believe that these facts are an important issue that limits a broaden use of these approaches. In this sense, to the best of our knowledge, the only approaches that use goal-oriented requirements analysis techniques for Web engineering have been presented in (Bolchini & Paolini, 2004) and (Molina, Pardillo, & Toval, 2008). Unfortunately, although these approaches use the i* modeling framework (Yu, 1995, 2002) to represent requirements in Web domain, they do not benefit from every i* feature because don't use all the expressiveness of the i* framework to represent the special type of requirements of the Web applications such as the related with navigational issues. To overcome this situation, our previous work (J.A. Aguilar, et al., 2010) adapts the well-known taxonomy of Web requirements presented in (M.J. Escalona & Koch, 2004) to be used within the i* modeling framework.

On the other side, with regard to approaches that consider NFRs from early stages of the development process, in (Molina & Toval, 2009) the authors propose a metamodel for representing usability requirements for Web applications. Moreover, in (Ameller, et al., 2010) the authors present the state-of-the-art for NFRs in model-driven development, as well as an approach for integrating NFRs into a model-driven development process by considering them from the very beginning of the development process. Unfortunately, these works overlook how

to analyze and evaluate the impact among FR and NFRs. However, some interesting works have been done in this area (Horkoff & Yu, 2009a) and (Horkoff & Yu, 2009b). These works evaluate i* models based upon an analysis question (whatif) and the human judgment. To this aim, this procedure uses a set of evaluation labels that represent the satisfaction or denial level of each element in the i* model. First of all, initial evaluation labels reflecting an analysis question are placed in the model. These labels are then propagated throughout the model by using a combination of set propagation rules and the human judgment. The results of this propagation are interpreted in order to answer the stated question. Unfortunately, these general approaches have not been adapted to Web engineering.

The motivation with regard this proposal relies on the deficiencies from the works previously mentioned. Since, they are focused on how to analyze i* models with which to answer a particular question (what-if) without considering the goal satisfaction (the organizational objectives). Thus, our proposal is focused on how to evaluate the impact derived from a change in the i* requirements model in order to use it to offer to the designer a better form to analyze design options from the Web application. For this purpose, the requirements are classified according the taxonomy of Web requirements defined in (M.J. Escalona & Koch, 2004) in the i* requirements model. Moreover, our proposal brings out alternative paths to satisfy the goals bearing in mind the softgoals tradeoff, thus considering the softgoals from the beginning of the Web application development process.

In summary, there have been many attempts to provide techniques and methods to deal with some aspects of the requirements engineering process for the development of the Web applications. However, there is still a need for solutions which enable the designer to know which requirements are affected because of a change in the Web application design besides of considering the NFRs involved in the development process.

GOAL-ORIENTED REQUIREMENTS ANALYSIS IN WEB ENGINEERING

This section describes our proposal to specify requirements in the context of the A-OOH (Adaptive Object-Oriented Hypermedia) Web modeling method (Garrigós, 2008) by using goal-oriented models (J.A. Aguilar, et al., 2010).

A-OOH is an extension of the OOH (Object-Oriented Hypermedia) (Gómez, Cachero, & Pastor, 2000) method with the inclusion of personalization strategies. The development process of this method is founded in the MDA (Model Driven Architecture) (Melia, Cachero, & Gomez, 2003). MDA is an OMG's standard and consists of a three-tier architecture with which the requirements are specified at the Computational Independent Model (CIM), from there are derived the Web application conceptual models which corresponds with the Platform Independent Model (PIM) of the MDA. Finally, the Web application conceptual models are used to generate the implementation code; this stage corresponds with the Platform Specific Model (PSM) from the MDA standard.

The A-OOH approach uses five models at the PIM level to define a Web application, namely:

- Domain model (DM). This encapsulates the structure and functionality required of the relevant concepts of the application domain and reflects the static part of the Web application, is represented such as UML (Unified Modeling Language) class diagram.
- Navigation model (NM). This model aims to specify the structure and behavior of the navigation view over the domain data defined. It defines each path on which users can navigate through the Web application.
- Presentation model (PM). This model defines the layout of the Web application, i. e. the page style, the font color, etc.
- Personalization model (PM). Personalization strategies (Adaptive Web Sites) are specified in this model. Web pages are personalized based on the context, the socioeconomic level and the interest of an individual user. Personalization implies that the changes are based on the items purchased or in the pages viewed.
- User model (UM). This enables the users to be described in terms of their personal information and their relations with a particular application domain as well as the navigational actions performed at execution-time. The structure of the information needed for the personalization strategies is also described in this model.

On top of these models, the A-OOH approach defines Web requirements in a CIM (J.A. Aguilar, et al., 2010). From this CIM, a several transformation rules have been defined for supporting the derivation of the navigational (NM) and domain (DM) conceptual models with a high level of automation (J.A. Aguilar, et al., 2010).

Requirements specification at the computational independent model

The requirements specification in the A-OOH method is performed by means of the i* modeling framework (Yu, 1995, 2002). As a goal-oriented analysis technique, the i* framework focuses on the description and evaluation of alternatives and their relationships to the organizational objectives.

The i* framework consists of two models: the strategic dependency (SD) model (⊸▷⊸) to describe the dependency relationships among various actors in an organizational context, and the strategic rationale (SR) model (◌), used to describe actor interests and concerns and how they might be addressed. The SR model provides a detailed way of modeling internal intentional elements and relationships of each actor (◯). The essential elements to perform a goal-oriented requirements analysis using the i* modeling framework are described next.

- Goal (◯) represents an (intentional) desire of an actor; regrettably, goals are not enough for describing how the goal will be satisfied.
- Means-end links are a kind of links representing alternative ways for fulfilling the goals.
- Task (◇) describes some work to be performed in a particular form. The task is decomposed in the necessary intentional elements (tasks and resources) to be performed. This decomposition is by means of the decomposition links (⎯⏉⎯).
- Resource (▭) represents some physical or informational entity required for the actor, in particular by the task element.
- A softgoal (⬭) is a goal whose their satisfaction criterion is not clear cut.

In which form contributes an intentional element to the satisfaction or fulfillment of a softgoal is determined via contribution links (→). The possible labels for a contribution link are "make", "some+", "help", "hurt", "some-", "break", "unknown", this labels are used to indicate the strength of the contribution made, this could be positive, negative or unknown.

Even though the i* modeling framework provides good mechanisms to model stakeholder's (actors) and relationships between them, the Web applications have some particular requirements that differs from the traditional ones. These new requirements are defined in the seminal work of Escalona and Koch (M.J. Escalona & Koch, 2004). In this work, the authors put forward the argument that FRs for Web engineering are related to three main features of Web applications: navigational structure, user interface and personalization capability, and that the data structures required by the Web application should also be specified. Therefore, to adapt the i* modeling framework to the Web engineering domain we use the clasification described in the work of Escalona and Koch. As a strong point, with this clasification is possible to reflect special Web requirements that are not taken into account in traditional requirement analysis approaches. An overview of each kind of requirement for Web engineering is described next:

– Content Requirements. With this type of requirements, is defined the website content presented to users. For example, in a book on-line store some examples might be: "book information" or "book categories".
– Service Requirements. This type of requirement refers to the internal functionality the system as Web application should provide to its users. Following the example of the Content Requirements, for instance: "register a new client", "add book to cart", etc.
– Navigational Requirements. These type of requirements refers to the navigational paths available for the users of the Web system. In this sense, some examples are: the user navigation from index page to "consult products by category" or to "consult shopping cart" options.
– Layout Requirements. Requirements can also define the visual interface for the users. For instance: "present a color style", "multimedia support", "the user interaction", among others.
– Personalization Requirements. The designer can specify the desired personalization actions to be performed in the final website (e.g. "show recommendations based on interest", "adapt font for visual impaired users", etc.)
– Non-Functional Requirements. These kinds of requirements are related to quality criteria that the intended Web system should achieve and that can be affected by other requirements. Some examples can be "good user experience", "attract more users", "efficiency", etc.

The A-OOH approach is UML-compliant, therefore we have used the extension mechanisms of UML in order to adapt the i* modeling framework to specific Web domain terminology. To do so, (i) we defined a profile to formally represent the adaptation of each one of the i* elements with each requirement type from the Web requirements clasification adopted (Garrigós, et al., 2009); and (ii) we implemented this profile in an EMF (Eclipse Modeling Framework) metamodel. Therefore, new EMF clases have been added according to the different kind of Web requirements (see Fig. 1): the Navigational, Service, Personalization and Layout requirements extends the Task element and the Content requirement extends the Resource class. It is worth

noting that NFRs can be modeled by directly using the softgoal element. The EMF metamodel for Web requirements specification using the i* framework has been implemented in the Eclipse IDE (Integrated Development Enviroment).
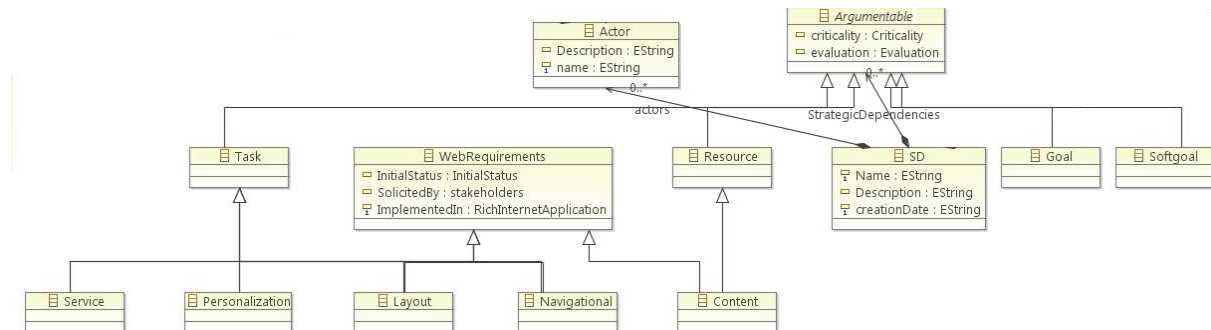

Figure 1. An extract of the i* metamodel for Web requirements

A sample application of the i* modeling framework for Web domain is shown in Figure 2, which represents the SR model of our running example for the Conference Management System (CMS). The purpose of the system is to support the process of submission, evaluation and selection of papers for a conference. The complete specification of the case study can be found at: http://users.dsic.upv.es/~west/iwwost01

A part of the CMS requirements specification (CMS Actor) is modeled in the Figure 2. The requirement specification is focused in the process of selecting the review procedure. Four actors participate in the CMS, for this example only the author, reviewer and system actors were considered. It is important to remark that each element from Figure 2 corresponds to a requirements type from the clasification previously mentioned (from Escalona and Koch), i.e., the content requirement (Content) from the taxonomy is displayed with the notation "Resource" from the i* modeling framework and the navigational (Navigational) and service (Service) requirements with the symbol "Task", both of them with their respective associations called decomposition-links. A decomposition-link between two elements means that a requirement ("Task") is decomposed in one or more sub-requirements ("Sub-Tasks") to be able to perform its function. The Figure 2 depicts a correct scenario of the requirement decomposition by introducing the navigational requirement "Blind Review Process", decomposed in two sub-requirements named "Download papers without authors' name" (Service) and "Review Paper" (Navigational). Besides, the labels (☑) and (×) are used in the requirements specification to represent the requirements that are currently implemented in the Web application conceptual models.
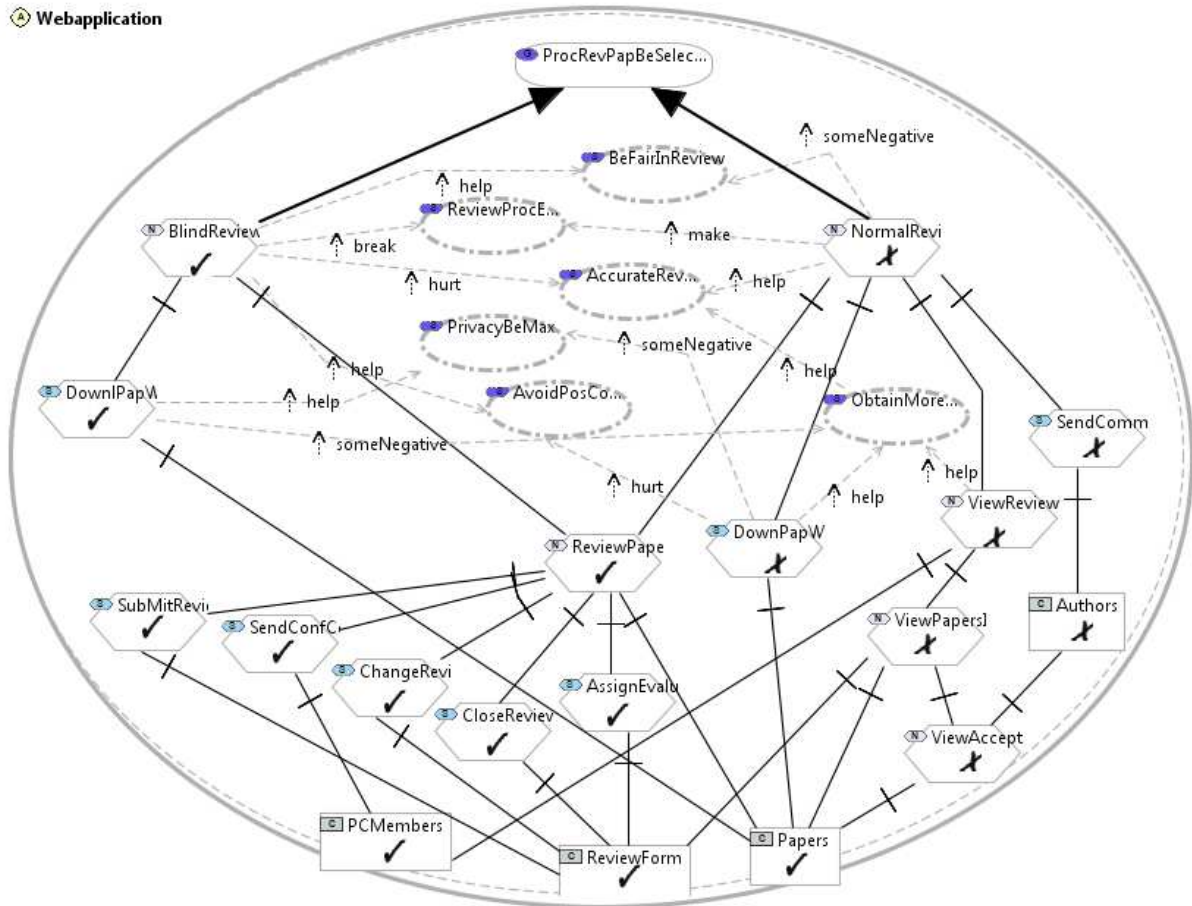
Figure 2. Part of the Conference Management System requirements expressed in a SR and SD Models

Applying the goal-oriented requirements analysis, we found that there are three actors detected which depends on each other to achieve their goal; these are namely "Reviewer", "Author" and "Conference Management System". The "Reviewer" actor needs to use the CMS to achieve its own goal which is "Review paper". The "Author" actor depends on the CMS actor in order to "Paper be reviewed". In Figure 2 are modeled these dependencies as well as the CMS actor by means of the SD and the SR models. The goal of the CMS actor is "Process of review of papers be selected". To achieve the goal, the SR model specifies that one of the two navigational requirements: "Blind review process" or "Normal review process" should be performed. In this running example, the path to achieve the goal of the CMS actor is through the navigational requirement "Blind review process", all the requirements needed to implement this path are labeled with (☑). Besides, we can observe in the SR model that some navigational and service requirements are decomposed in other requirements, some of them affects positively or negatively some NFRs, i.e., the service requirement "Download paper without authors' name" needs the information provided by the content requirement "Papers". Moreover, the service requirement affects positively the softgoal "Privacy be maximized" and in some negatively form the softgoal "Obtain more complete info". This fact is very important to see how to satisfy the goal "Process of review of papers be selected" considering the Web application softgoals.

Accordingly, maximizing or minimizing the contribution from requirements to softgoals is a viable solution to find a path to fully-satisfy the goal.

From the computational independent model to the platform independent model

Once the requirements have been defined they can be used to derive the Web application conceptual models. For the sake of understandability, the process for obtaining the navigational conceptual model will not be described in detail, and our main focus will be on the derivation of the domain model. For a broaden explanation about this process, we refer reader to (J.A. Aguilar, et al., 2010).

The A-OOH domain model is expressed as a UML-compliant class diagram. It encapsulates the structure and functionality required of the relevant concepts of the application and reflects the static part of the system. The main modeling elements of a class diagram are the classes (with their attributes and operations) and their relationships. We have implemented a set of rules using the ATL (Atlas Transformation Language) (ATL, 2011) to derive the domain model from requirements specification (model). In this transformation rules, the source model correspond to our CIM for Web requirements specification while the domain model as a representation model for the Web application domain is the PIM level, both of them in a MDA context.

- Content2DomainClass. By using this transformation rule, each content requirement detected in the requirements model is derived into one class of the domain model (see Fig. 3).
- Service2Operation. Detects a service requirement with an attached content requirement in the SR model, each service requirement is transformed into one operation of the corresponding class (represented by the content requirement).
- Navigation2Relationship. To generate the associations in the domain model we have to detect a navigational root requirement (i.e. task) in the SR model which can contain one or more navigational requirements attached. Each of the navigational requirement can have attached a resource (i.e. content requirement) thus creating an association from one content requirement to other one by means of the navigational requirements associated.

```
--For each Content element in the i* Requirements Model, a UML Class will be created in the Domain Model.
rule Content2Class
{
    from
        vContent : MMiStar!Content
    to
        vClass : UML!Class (name<-vContent.name),
 __traceLink : traceability!TraceLink (
    name <- 'Content2DomainClass',
    sourceElements <- Sequence {__LinkEnd_vContent},
    targetElements <- Sequence {__LinkEnd_vClass},
    model <- thisModule.__wmodel, ruleName <-'Content2DomainClass', description<-'From each Content element
    in RequirementsModel, this rule creates a UML Class in DomainModel'),
    __LinkEnd_vContent : traceability!TraceLinkEnd (element <- __elementRef_vContent ),
    __elementRef_vContent : traceability!ElementRef (ref <- vContent.__xmiID__, modelRef <- thisModule.__model_IN),
    __LinkEnd_vClass : traceability!TraceLinkEnd (element <- __elementRef_vClass ),
    __elementRef_vClass : traceability!ElementRef (ref <- vClass.__xmiID__, modelRef <- thisModule.__model_OUT)

}
```

Figure 3. The transformation rule Content2Class implemented by means of ATL

Figure 4 shows the domain model obtained from requirements specification (model). The domain model is showed by means of the tree editor for easier viewing, all the classes, operations, associations with their respective properties are encapsulated in a UML-Package.

Once the model is derived the designer has only to refine them, avoiding the task of having to create them from scratch. The designer will have to specify the most relevant attributes of the classes, identify the cardinalities and define (if existing) the hierarchical relationships.

IMPACT ANALYSIS ALGORITHM FOR GOAL-ORIENTED REQUIREMENTS IN WEB ENGINEERING

In this section, we present a proposal which provides a form to analyze the impact of a change in the requirements specification (requirements model) within the A-OOH method. Therefore, the Web developer will be able to evaluate the effect of the change and select a sub-set of requirements to implement to fully satisfy the goal. To do this, the designer must perform (find) a balance between the contributions made by the FRs to the NFRs.

The algorithm is designed to be applied in the requirements specification (the i* requirements model) considering the type of contributions made by the intentional elements to the softgoals. This algorithm allows evaluating the impact in the requirements model resulting from removing any element of the A-OOH conceptual models (it is worth noting that, in this paper, our focus is on the previously defined domain model). In this way, it is determined which new requirements should be included in the A-OOH conceptual models for maximizing or balancing the softgoal satisfaction although some requirements have been removed. To this aim, some heuristics have been defined.

Heuristics

Some heuristics have been defined for determining the impact of the contribution links between intentional element and softgoals. Table 1 summarizes some terms for understanding the heuristics presented in this section. These terms correspond to the most common types of contribution links of the i* modeling framework.

Heuristics Terms	i* Contribution Type
Strongly-positive	Help
Weakly-positive	Some +
Strongly-negative	Hurt
Weakly-negative	Some -
Dependent-negative	Break
Dependent-positive	Make

Table 1. The i contributions types*

```
platform:/resource/Requirements2DomainModel/OutputModel/DomainModel.uml
  <Package>
    <Class> Authors
      <Operation> SendComments2Authors ()
    <Class> Papers
      <Operation> DownPapWithAuthorName ()
      <Operation> DownlPapWithoutAuthName ()
    <Class> ReviewForm
      <Operation> AssignEvaluation ()
      <Operation> CloseReview ()
      <Operation> ChangeReview ()
      <Operation> SubMitReview ()
    <Class> PCMembers
      <Operation> SendConfComments2PC ()
    <Association> Papers--->Authors
      <Property> dst : Authors
      <Property> src : Papers
    <Association> Papers--->Papers
      <Property> dst : Papers
      <Property> src : Papers
    <Association> ReviewForm---->Authors
      <Property> src : ReviewForm
      <Property> dst : Authors
    <Association> ReviewForm---->Papers
      <Property> dst : Papers
      <Property> src : ReviewForm
    <Association> PCMembers---->Papers
      <Property> src : PCMembers
      <Property> dst : Papers
    <Association> PCMembers---->ReviewForm
      <Property> src : PCMembers
      <Property> dst : ReviewForm
```
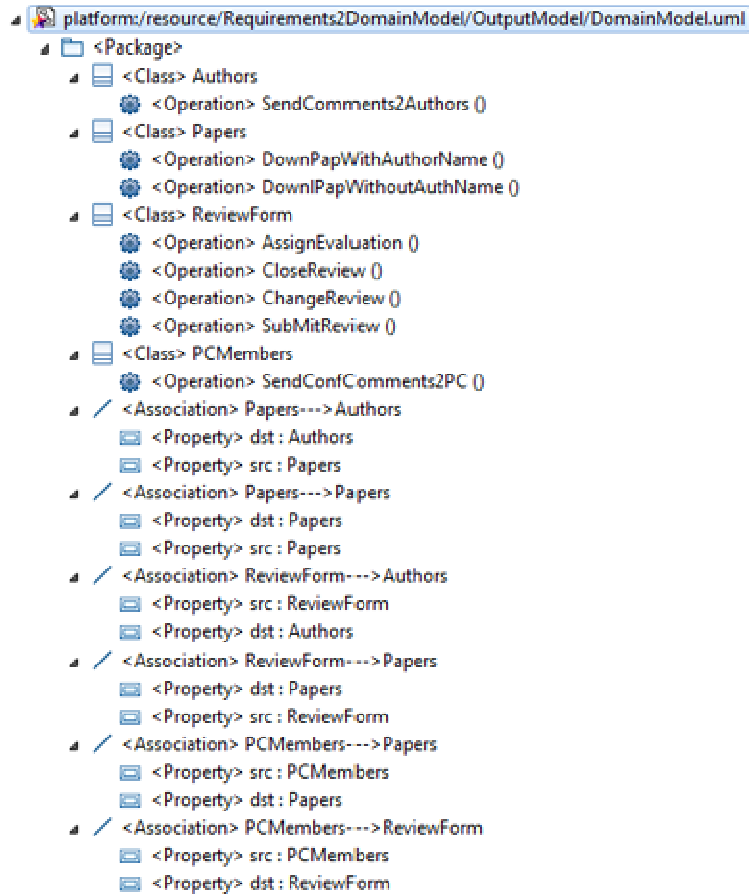
Figure 4. The the domain model derived from the requirements model

The "Help" contribution link is a partial positive contribution, not sufficient by itself to satisfy the softgoal. The contribution link named "Hurt" is partial negative contribution, not sufficient by itself to deny the softgoal. "Some +" is a positive contribution whose strength is unknown. "Some -" is the opposite contribution type to "Some +", is a negative contribution whose strength is unknown. The "Break" contribution link refers to a negative contribution enough to deny a softgoal. Finally, the "Make" contribution link is a positive contribution strong-enough to satisfy a softgoal.

The heuristics defined are as follows:

- H1. If the contribution of the requirement to remove is strongly-positive, and the contribution of the requirement to implement is strongly-negative, do not implement the requirement.
- H2. If the contribution of the requirement to remove is more stronglypositive than the contribution of the requirement to implement, but the contribution to be implemented is weakly-negative, the requirement could be implemented.

- H3. If the contribution of the requirement to remove is strongly-negative or weakly-negative, and the contribution of the requirement to implement is strongly-positive or weakly-positive, the requirement should be implemented.
- H4. If the polarity of the contribution of the requirement to remove is as negative as the contribution of the requirement to implement, the requirement to implement should not be implemented to maximize the satisfaction of the softgoal.
- H5. If the polarity of the contribution of the requirement to remove is as positive as the contribution of the requirement to implement, the requirement to implement should be implemented to maximize the satisfaction of the softgoal.
- H6. If the contribution of the requirement to remove is Dependent-positive, then the developer should consider whether that requirement should be removed or not considering the need to implement this softgoal.
- H7. If the contribution of the requirement to remove is Dependent-negative, then the developer should consider whether that requirement should be removed or not considering the impact of not implementing this softgoal.

Preconditions and postconditions

The preconditions should hold before the algorithm could be executed and they affects to the elements specified in the requirements model described in the section "Goal-oriented requirements analysis in web engineering". Specifically, these preconditions permit the execution of the algorithm when:

1. The requirement to remove does not affect the goal by the "means-end" contribution type.
2. When there is more than one "means-end" contribution type, it means that the impact analysis will be possible by means of the softgoals tradeoff.
3. The requirement to remove affects other requirements and these requirements (not shared) are not in the possible paths to satisfy the goal.

Moreover, there is one postcondition that must hold when a requirement has been selected to be implemented in the requirements model as an alternative solution for the satisfaction of the goal: if the requirement to be implemented has associated requirements, these requirements must be implemented automatically.

Impact Analysis Algorithm

This algorithm considers the contributions made by the intentional elements from the requirements model to find a path to fully satisfy, where possible, the main goal. To do this, the designer must have to find tradeoffs between the softgoals. The algorithm is presented next.

1. *FUNCTION TradeOffAlgorithm (RequirementsModel)*
2. *TR= task to remove; TI= task to implement;*
3. *SN= new softgoal; IEList= intentional elements list;*
4. *ASList= affected softgoals list;*
5. *TIList= list of task to implement; Value= false;*
6. *P = PreConditions ( ) ;*
7. *IF (P=true) THEN*
8. *IELi st= CreateIntentionalElement sList (RequirementsModel) ;*
9. *IF(TR. Contributes2Softgoal s ( )) THEN*
10. *ASList=Cr eateAffectedSoftgoa l sLi s t (TR) ;*
11. *FOREACH s FROM ASLi s t :*
12. *TI= SearchTaskToApply (IELi s t) ;*
13. *Value= Heur i s t i c s (ASList , IEList , TI) ;*
14. *TI . AddValue (Value) ;*
15. *TIList .Add(TI) ;*
16. *IF (TI . Contributes2Softgoal s (TI)) THEN*
17. *ASList . add (SN) ;*
18. *END IF*
19. *END FOREACH*
20. *FOREACH v FROM TILi s t :*
21. *CalculateAverage (v) ;*
22. *IF(CalculateAverage (v)) THEN*
23. *Implements (v) ;*
24. *END IF*
25. *END FOREACH*
26. *END IF*
27. *PostCondition ( ) ;*
28. *ELSE*
29. *ShowMessage (P.message ( )) ;*
30. *END IF*
31. *END PROGRAM*

1.1 Algorithm for impact analysis in goal-oriented Web engineering

A snippet of code of the algorithm for impact analysis in goal-oriented Web requirements specification is shown in 1.1. First, in lines 6 and 7 the pre-conditions are evaluated; these must be "true" to proceed with the execution of the algorithm. Next, from lines 8 to 19, the algorithm creates a list of intentional elements. In these lines, all types of requirements from the requirements model are stored in this list. The next step is to extract those softgoals that receive a contribution from the requirement to remove, for each softgoal from the list, finding a non-

implemented requirement and applying the heuristics introduced in the section "Impact Analysis Algorithm for Goal-oriented requirements in web engineering". Each of these requirements must be stored in the list, and if it contributes to a softgoal, the softgoal must be stored in the list too. Then, lines 20 to 29 are used to evaluate each element from the list according to the weight of each element assigned by the heuristics to determine when a requirement must be implemented. Finally, the postcondition is executed and the alternative path to fully satisfy the goal from requirements model is obtained.

PERFORMING THE IMPACT ANALYSIS

For the sake of understandability, the following scenario is assumed along this section: the Web developer decides deleting from the A-OOH domain model the elements that were created from the requirement "Download papers without authors' name". It is necessary to know which other requirements are affected by this change. In addition, this action implies that the goal "Process of review of papers be selected" can not be satisfied. Thus, it is necessary to search for alternative paths in the i* requirements model (if there any) in order to fully-satisfy the goal "Process of review papers be selected". To this aim, our algorithm is triggered. The execution of the impact analysis algorithm is detailed next.

The first step to execute our algorithm consists of applying the preconditions. For this running example, the preconditions result true, it means that there is any problem to the algorithm has been executed.

Next, it is necessary to develop a list of the requirements (implemented or not) that contribute to any softgoal in the i* requirements model (see Table 2). Also, if a softgoal contributes to other one, the softgoal must be added to the list too.

Requirements	S1	S2	S3	S4	S5	S6
"Blind review process"	Help	Break	Hurt	Help	-	-
"Download papers without authors' name"	-	-	-	-	Help	Some-
"Normal Review Process"	Some-	Make	Help	-	-	-
"Download paper with authors' name"	-	-	-	Hurt	Some-	Help
"View review process status"	-	-	-	-	-	Help
"Obtain more complete info"	-	-	Help	-	-	-

Table 2. The requirements contributions to the softgoals

Table 2 highlights in bold the requirement to be removed. This table shows a requirements list (FR and NFRs) and their type of contributions to the softgoals where S1 corresponds to softgoal "Be fair in review" from requirements model, S2 to "Review process easier", S3 represents "Accurate review process", S4 conforms to "Avoid possible conflicts of interest", S5 it's the "Privacy be maximized" softgoal and S6 refers to "Obtain more complete info".

The next step is to identify the number of softgoals affected by the requirement to be removed. If necessary, a list of the softgoals that receive a contribution from the requirement to be removed is made. In this example, the requirement to be removed is "Download papers without authors' name", this one affects two softgoals: "Privacy be maximized" and "Obtain more complete info" S5 y S6 respectively (see Table 2).

For each softgoal that receives a contribution from the requirement to be removed, we search for a non-implemented requirement of which contribution compensates the possible elimination of the requirement to be removed. To do this, it is necessary applying the heuristics defined in the section "Impact Analysis Algorithm for Goal-oriented requirements in web engineering".

For example, the softgoal "Privacy be maximized", according to Table 2, receives a strongly-positive contribution (Help) from the requirement to be removed, thus being necessary searching for a non-implemented requirement to contribute to this softgoal. In this case, only the requirement "Download papers with authors' name" contributes (negatively) to this softgoal (weakly-negative, i.e. Some -). Therefore, applying the heuristics described in the section "Impact Analysis Algorithm for Goal-oriented requirements in web engineering", specifically the heuristic number 2 (H2), the requirement "Download papers with authors' name" could be implemented.

Considering the softgoal "Obtain more complete info" according to Table 2, it receives a weakly-negative contribution (Some -) from the requirement to be removed, thus being necessary searching for a non-implemented requirement to contribute to this softgoal. In this case, two requirements (positively) contribute to this softgoal, "Download papers with authors' name" and "View review process status" (strongly-negative, i.e. Help). Therefore, the heuristic H3 applies for this softgoal, thus, these requirements should be implemented.

After analyzing the softgoals contributions, the next step is searching for any softgoal in the requirements list that contributes to another softgoal. In this example, the softgoal "Obtain more complete info" makes a strongly-positive contribution (Help) to the softgoal "Accurate review process", thus, the next step consists of searching for the requirement that makes a contribution to the softgoal and applying the heuristics. Therefore, the requirement that makes a contribution to the softgoal "Accurate review process" is "Normal review process", this contribution is strongly-positive (see Figure 2), hence, according to H3 this requirement must be implemented.

After having performed the algorithm the requirements that could be implemented in order to fully-satisfy the goal "Process of review papers be selected" after having removed the requirement "Download papers without authors' name"; these are (i) "Normal Review Process", (ii) "Download paper with authors' name" and (iii) "View review process status". Next, it is necessary to evaluate the heuristics assigned to each requirement to know what could be implemented. To do this, it is necessary to evaluate the contribution type of each requirement, i.e., the navigational requirement "Download papers with authors' name" negatively contributes (Some -) to the softgoal "Privacy be maximized", thus hurting the softgoal "Avoid possible conflicts of interest" (Hurt) and helping the softgoal "Obtain more complete info" (Help). Therefore, by using the human judgment the navigational requirement "Download papers with authors' name" can be implemented. For the navigational requirement "Normal review process",

it is easier to determine whether it can be implemented because it only contributes to one softgoal, the "Accurate review process", hence its contribution is Help, and therefore this requirement must be implemented. Finally, the navigational requirement "View review process status" positively contributes to the softgoal "Obtain more complete info"; consequently this requirement must be implemented.

The final step is to apply the postcondition from the section "Impact Analysis Algorithm for Goal-oriented requirements in web engineering". In this running example, according to the postcondition, it is necessary to implement the navigational requirements "View papers info" and "View Accepted/Rejected papers" because these requirements are associated with the navigational requirement "View Review Process Status". In addition, the content requirement "Authors" and the service requirement "Send Comments to Authors" must be implemented too in order to implement the alternative path to fully satisfy the goal "Process of review papers be selected". Hence, the content requirement "Authors" is associated with the navigational requirement "View Accepted/Rejected papers" and the service requirement "Send Comments to Authors" is related with the navigational requirement "Normal review process".

After finishing the execution of the algorithm, we obtain the requirements that are directly and indirectly affected by the deletion of the requirement "Download papers without authors' name". Moreover, the algorithm can find out which requirements must be implemented to continue satisfying the goal considering the contributions received from the softgoals. In this running example the requirements to implement are: "Download papers with authors' name", "Normal review process" and "View review process status". Finally, according to the post-condition the requirements "View papers info", "View Accepted/Rejected papers", "Authors" and "Send Comments to Authors" must be implemented too. Figure 5 shows the final requirements model with the alternative path implemented to fully-satisfy the goal "Process of review papers be selected".

OPEN SOURCE IMPLEMENTATION FRAMEWORK

In this section we describe in detail the implementation of the impact analysis algorithm within our approach for goal-oriented requirements analysis in Web engineering. To this aim, we have combined a set of technologies such as Eclipse (Eclipse, 2011), EMF (Eclipse Modeling Framework) (EMF, 2011), GMF (Graphical Modeling Framework) within the GMP (Graphical Modeling Project) (GMP, 2011) and Java (JAVA, 2011).

Eclipse is an open source IDE (Integrated Development Environment) used as a software platform to create integrated development environments; within Eclipse, the EMF (Eclipse Modeling Framework) project is a modeling framework and code generation facility for building tools and other applications based on a structured data model (abstract syntax). Also, the facilities for creating metamodels and models are provided by the metametamodel Ecore. Therefore, by using the facilities offered by EMF, it is possible to create a visual representation of the elements defined within the EMF metamodel by means of GMP (concrete syntax). The Eclipse Graphical Modeling Project (GMP) provides a set of generative components and runtime infrastructures for developing graphical editors based on EMF and GEF (Graphical Editing Framework). Both the Eclipse Modeling Framework (EMF) and the Graphical Modeling

Framework (GMF) are capable of generating editor plug-ins. Next, each one of the steps performed for the implementation framework is described.
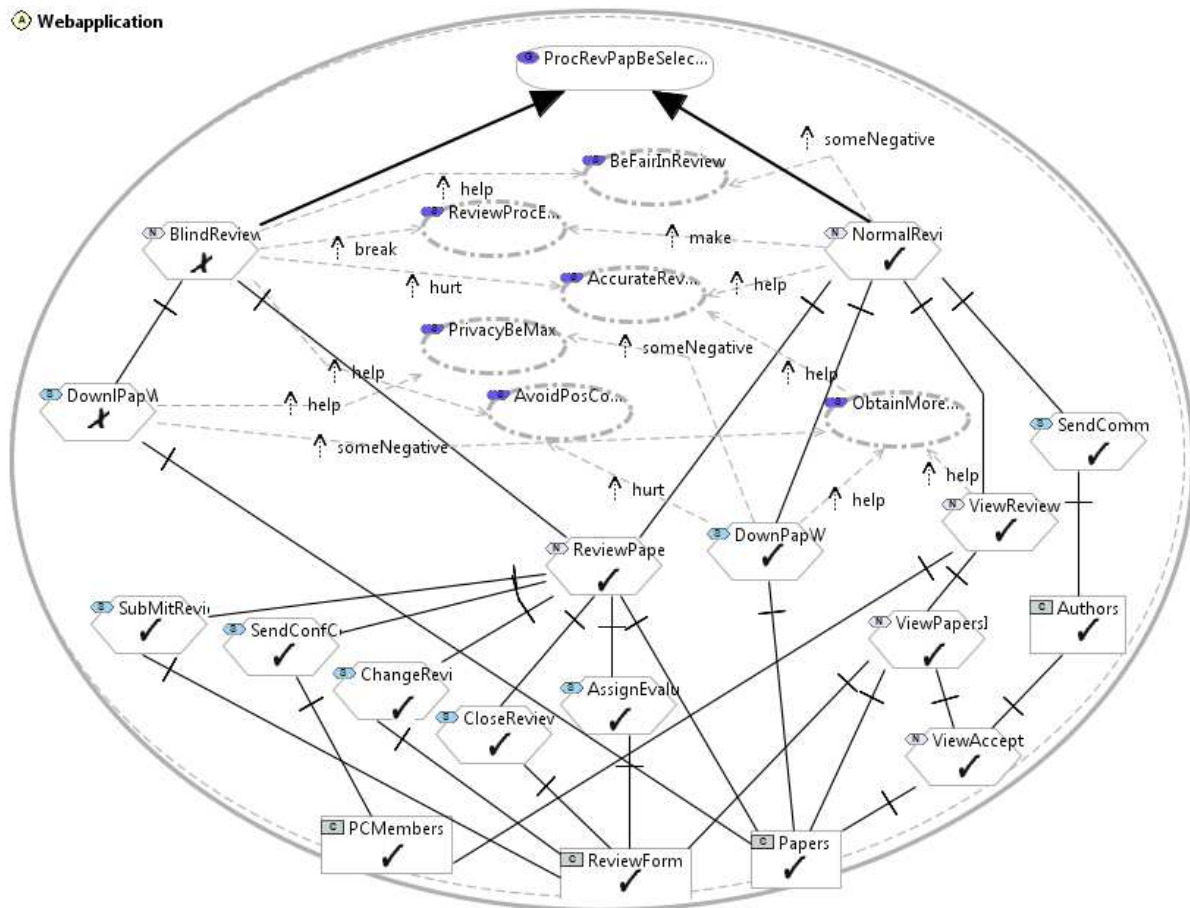

Figure 5. The Conference Management System requirements expressed in a SR and SD Models with the new requirements to implement (the alternative path "Normal Review Process")

The first step is the implementation of the Web requirements metamodel. The requirements metamodel was created using the EMF metamodel to incorporate a number of taxonomic features for the specification of Web requirements. With the implementation of this metamodel has been possible to adapt the i* modeling framework in the Web domain, with which is possible to model the needs and expectations of the stakeholders of the Web application. The clasification of Web requirements presented in (M.J. Escalona & Koch, 2004), and previously defined, have been incorporated as Ecore classes to represent each type of the requirements clasification. Figure 6 shows a screenshot where the reader can see the implementation of this metamodel in Eclipse IDE by means of the EMF.
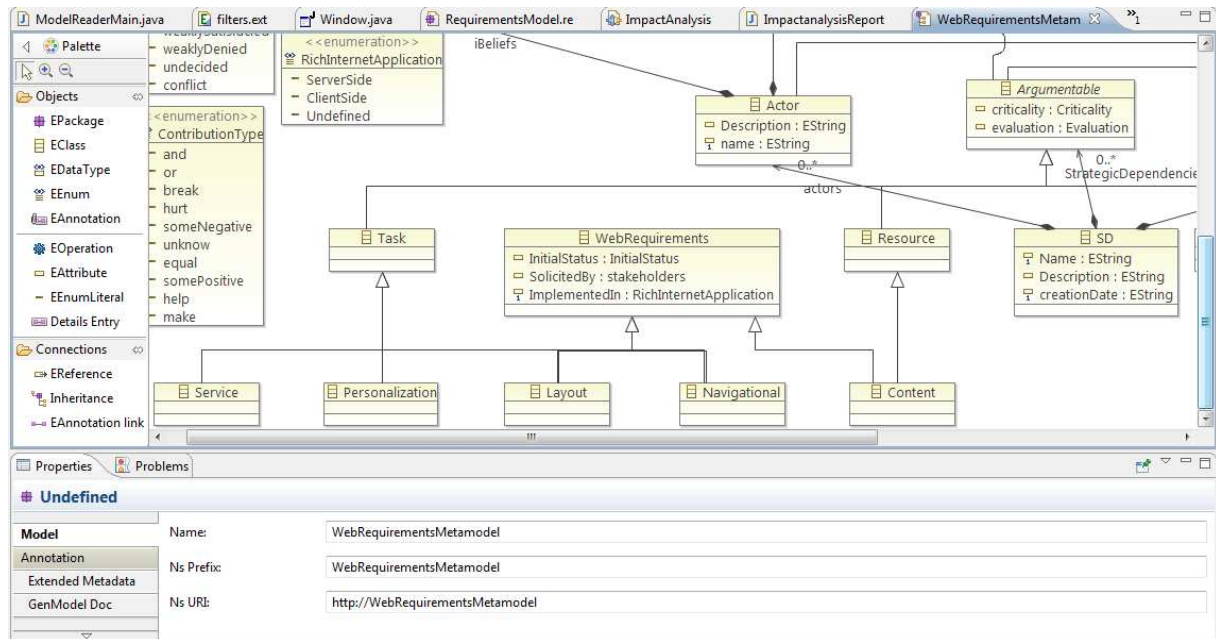
Figure 6. Screenshot of the Web requirements metamodel in Eclipse EMF

Once the metamodel has been implemented it is necessary to provide a graphical tool to assist the designer with the requirements specificaton. To do so, we have implemented a graphical editor using the GMF technology from the Eclipse Graphical Modeling Project.

GMF is a framework composed by three models: (i) Domain Model, (ii) Graphical Definition Model and (iii) Tool Definition Model. The Domain Model (ecore/genmodel files) is the starting point for the development of most of the Eclipse-based applications. The Domain Model represents the abstract syntax of the application domain, with this model the domain objects are represented as EMF classes. The Graphical Definition Model is a list of figures and shapes described in gmfgraph files, which will be used in the diagram to display classes from the domain model in a particular form defined by the designer. The Tool Definition Model (gmftool file) is the visual representation for the tools that will be avaliable in the final editor; basically, it defines what text you want to display on the tool palette and the button's tool tip. These three models are combined in the Mapping Model (*gmfmap* file) with which GMF knows what action take place when the designer selects a tool (Tool Definition Model), which classes need to be created (from the Domain Model), and what figures its necessary to render (Graphical Definition Model) when those classes are added to the diagram editor by the designer. Finally, once these models are combined in the Mapping Model, GMF transforms this model in the Diagram Generator Model. Then, it is generated a *gmfgen* file and the diagram application code.

In Figure 7, it is displayed a screenshot of the graphical editor implemented (called WebREd) (WebREd, 2011) by combining the Web requirements metamodel and the GMF technology. In the center of the figure is described the requirements specification for the the Conference Management System from the case study presented in the section "Goal-oriented requirements analysis in web engineering".
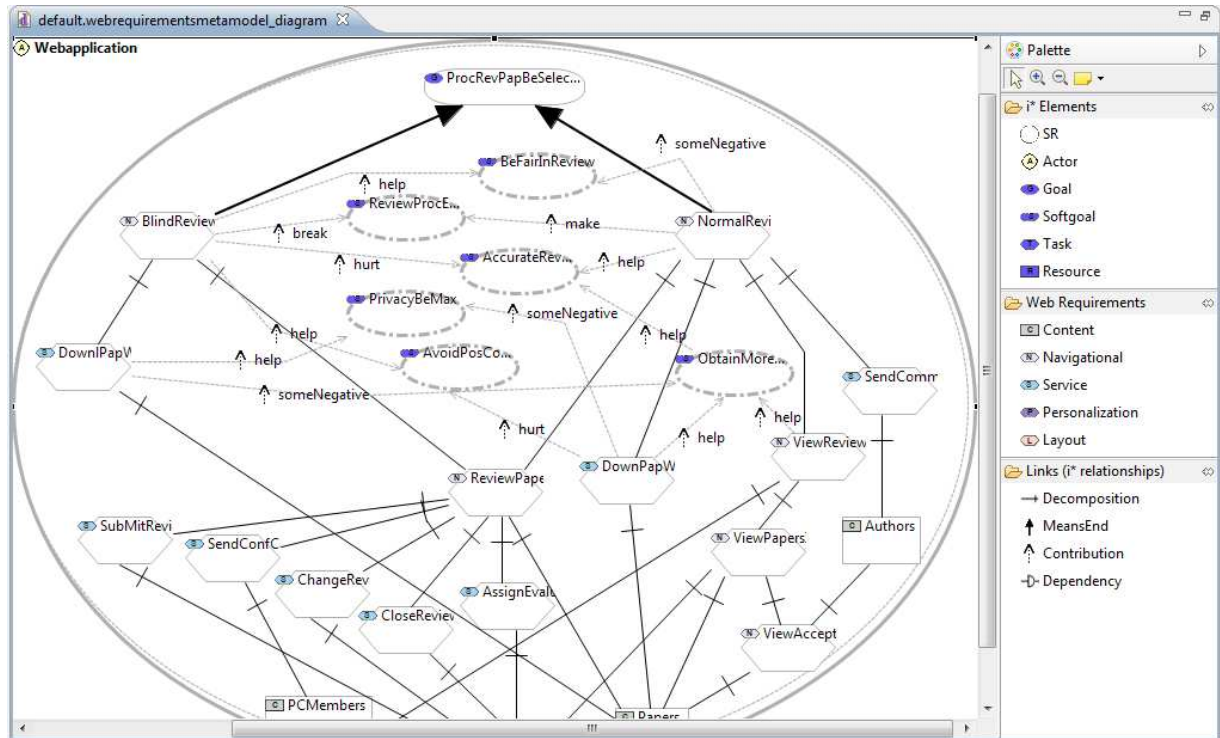
Figure 7. Screenshot of the Web requirements editor (WebREd)

As the reader can see, we have implemented each one of the elements of the Web requirements metamodel in the tool, thus the designer can model each requirement type from the clasification adopted (described in the section "Goal-oriented requirements analysis in Web engineering"). Also, WebREd provides the basic elements to model clasical goal-oriented models such as Task, Resource, Goal and Softgoal. The palette for drawing the different elements of the i* models for requirements specification can be seen on the right side of the Figure 7. At the top of the figure, there are those elements required to specify goal-oriented models according to the i* notation. Also, the requirements clasification adopted to specify requirements in the Web domain are in the right-center of the figure. Finally, the different types of relationships used in the i* modeling framework, with which the elements can be associated, are the right-botton of the figure.

With regard to the impact analysis support, this is performed in an automatic manner. When the designer specifies the requirements of the Web application by using the WebREd editor it is possible to know which requirements are affected due to a change in the Web application conceptual models. The impact of a change in the requirements can be consulted by the designer by means of a screen (window) and by means of a PDF (Portable Document File) report.

In Figure 8, it is shown a screenshoot of the impact analysis support offered by the WebREd editor. At the top of the figure, the main window for the impact analysis option is displayed. At the top we find the name of the element affected by a change originated in any of the Web application conceptual models described as "Element to remove". Also the information about the model currently selected is shown in the main window. In this particular case, this information is about the requirements model, thus including the name, description and creation date of the requirements model. Next, there is a tabbed pane with two options for the designer. The first one

shows a list of the requirements affected if and only if the requirement "Authors" is removed; also, is showed the type of each one of the affected requirements. Moreover, if the designer selects one of the requirements listed by double clicking on it, a new window (message dialog) is showed with a list of the softgoals affected by the requirement selected from the list (Figure 8, down). This softgoal's list shows the strength of the contribution made by the requirement affected to the softgoals. On the other hand, in the second tab, the requirements to be implemented (only when it is necessary) by the designer to still continue satisfying the goal are listed.

To conclude, the main window for the impact analysis option in the WebREd editor allows the designer to print a report in PDF (Portable Document File) format with which the designer can check the affected requirements.
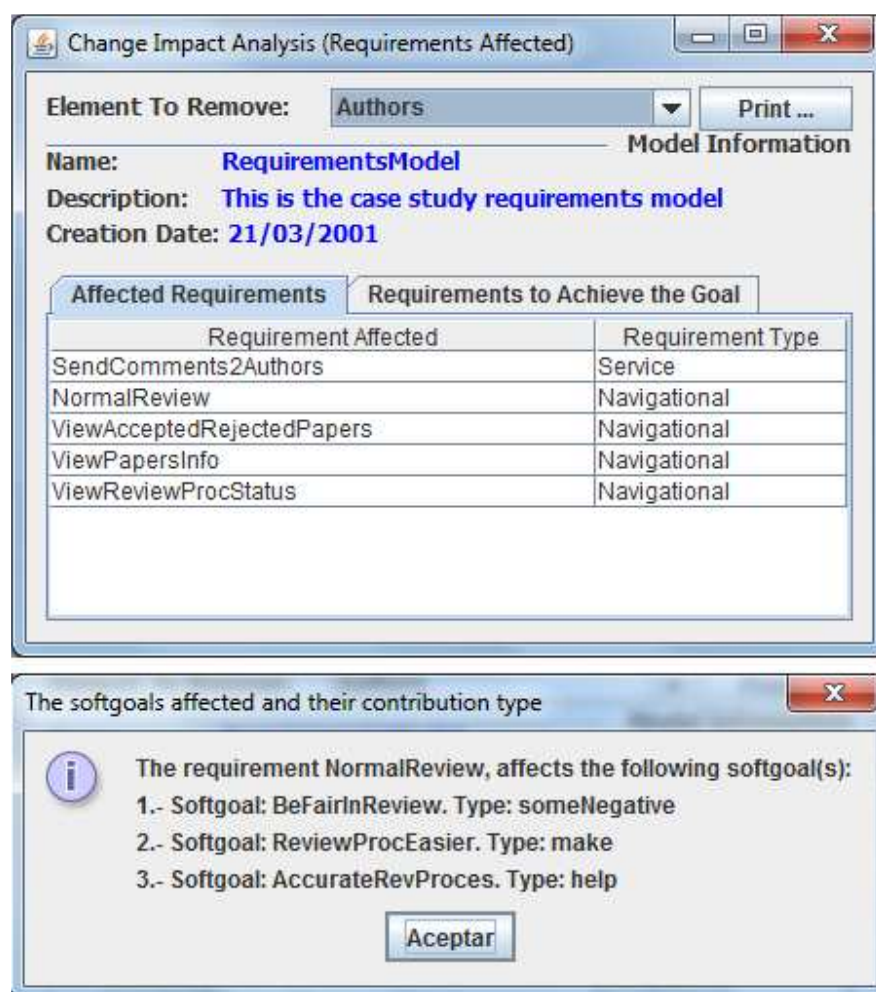

Figure 8. Screenshot of the impact analysis support offered by the Web requirements editor (WebREd)

CONCLUSION

Due to the dynamic idiosyncrasy of the Web and its heterogeneous audience Web applications should consider a requirement analysis phase in order to reflect, from the early stages of the Web application development process, specific needs, goals, interests and preferences of each user or user type, and due to the fast evolution of the Web, possible changes in those requirements should also be managed.

In this work, we have presented a methodology based on the i* modelling framework to specify Web requirements. Moreover, an algorithm to analyze the impact derived from a change done in the requirements model is presented.

Benefits of applying the algorithm described in this work include both the analysis of the impact derived from a change in a conceptual model and the ability to find an alternative path to fully-satisfy the goal by means of the softgoals tradeoff.

Nevertheless, according with the work presented in (Ameller, Franch, & Cabot, 2010), the softgoals are not considered with sufficient importance from the early stages of the development process. In this context, our proposal makes a contribution to the requirements analysis field considering the softgoals; hence it allows the designer to make decisions from the very beginning stages of the development process that would affect the structure of the envision website in order to satisfy users needs. Therefore, the designer can improve the quality of the requirements model analyzing the balance of the softgoals with the stakeholders.

Our short-term future work includes the definition of a metamodel to help to record the relationships among FR and NFRs, thus adapting the tradeoff algorithm presented in this work. Also, the impact analysis report will be completed with the adition of the requirements that the designer will need to implement to satisfy the goal. Besides, it is important to remark that the graphical editor is the basis for a prototype tool for the development of Web applications using the MDA (Model-Driven Architecture) paradigm.

Finally, note that this work has been done in the context of the A-OOH modeling method; however it can be applied to any Web modeling approach.

BIOGRAPHIES

José Alfonso Aguilar is a PhD candidate at the University of Alicante and is subventioned by the CONACYT (Consejo Nacional de Ciencia y Tecnología) and University of Sinaloa from Mexico. He has published papers about model-driven engineering and requirements engineering in national and international workshops and conferences, (such as WEBIST, ICCSA, WISM within ER, JISBD and so on) and in the Journal of Universal Computer Science (J.UCS). His Research interests are: Web engineering (WE), requirements engineering (RE), model-driven Web Engineering (MDWE). Other areas of interest: Content Management Systems (CMS), Geographic Information Systems (GIS), Web 2.0, Agile Software Development Methodologies, Process Improvement & CMMi.

Irene Garrigós is an assistant professor and post-doc researcher at the Department of Software and Computing Systems in the University of Alicante, (Spain), from which she holds a PhD and a Master in Computer Science. She has published several papers in national and international workshops, conferences and journals (such as ICWE, ER, WISE, APWEB, JISBD, information and software technologies, journal of Web engineering, and so on). Dr. Garrigós has served as a Program Committee member of several workshops and conferences such as ER, JISBD, WISM, MDA, FPUML, UWA and has served as assistant referee in several international conferences such as WWW and ICWE. She has done research stays in Belgium (Vrije Universiteit Brussel) and the Netherlands (Technische Universiteit Eindhoven). Her research interests are: Web engineering, personalization, model driven development, requirement engineering, Web and business intelligence, adaptive systems. She was involved in the organization of the 1st and 2nd edition of WeRE workshop (held in the RE and ICWE conferences, respectively) and the 1st and 2nd edition of the BEWEB workshop (held in the EDBT conference).

Jose-Norberto Mazón is an assistant professor at the Department of Software and Computing Systems in the University of Alicante (Spain). He obtained his Ph.D. in Computer Science from the University of Alicante (Spain). He has published several papers about data warehouses and requirement engineering in national and international workshops and conferences, (such as DAWAK, ER, DOLAP, BNCOD, JISBD and so on) and in several journals such as Decision Support Systems (DSS), SIGMOD Record or Data and Knowledge Engineering (DKE). He has also been co-organizer of the International Workshop on Business intelligencE and the WEB (BEWEB 2010 and 2011) and the International Workshop on The Web and Requirements Engineering (WeRE 2010 and 2011). His research interests are: requirement engineering, business intelligence, and model driven development.

REFERENCES

Aguilar, J. A., Garrigós, I., Mazón, J.-N., & Trujillo, J. (2010). *Web Engineering Approaches for Requirement Analysis - A Systematic Literature Review*. Paper presented at the 6th Web Information Systems an Technologies (WEBIST).

Aguilar, J. A., Garrigós, I., Mazón, J.-N. (2011). *Impact Analysis of Goal-Oriented Requirements in Web Engineering*. Paper presented at the 11th Computational Science and Its Applications (ICCSA).

Aguilar, J. A., Garrigós, I., Mazón, J. N., & Trujillo, J. (2010). An MDA Approach for Goal-oriented Requirement Analysis in Web Engineering. *Journal of Universal Computer Science, 16*(17), 2475-2494.

Ameller, D., Franch, X., & Cabot, J. (2010). *Dealing with Non-Functional Requirements in Model-Driven Development*. Paper presented at the Proceedings of the 18th IEEE International Requirements Engineering Conference.

Arnold, R. S., & Bohner, S. A. (1993). *Impact Analysis - Towards a Framework for Comparison*. Paper presented at the Proceedings of the Conference on Software Maintenance.

ATL. (2011). The Atlas Transformation Language. 2011, from http://www.eclipse.org/atl

Boehm, B., & In, H. (1996). Identifying Quality-Requirement Conflicts. *IEEE Softw., 13*(2), 25-35.

Bolchini, D., & Mylopoulos, J. (2003). *From Task-Oriented to Goal-Oriented Web Requirements Analysis*. Paper presented at the Proceedings of the 4th International Conference on Web Information Systems Engineering.

Bolchini, D., & Paolini, P. (2004). Goal-driven requirements analysis for hypermedia-intensive Web applications. *Requir. Eng., 9*(2), 85-103.

Casteleyn, S., Woensel, W. V., & Houben, G.-J. (2007). *A semantics-based aspect-oriented approach to adaptation in web engineering*. Paper presented at the Proceedings of the 18th conference on Hypertext and hypermedia.

Ceri, S., Fraternali, P., & Bongio, A. (2000). Web Modeling Language (WebML): a modeling language for designing Web sites. *Comput. Netw., 33*(1-6), 137-157.

Del Fabro, M., Bézivin, J., & Valduriez, P. (2006). *Weaving Models with the Eclipse AMW plugin.* Paper presented at the Eclipse Modeling Symposium, Eclipse Summit Europe, Esslingen, Germany.

Eclipse. (2011). Eclipse Foundation Open Source Community. 2011, from http://www.eclipse.org/

Elahi, G., & Yu, E. (2009). Modeling and analysis of security trade-offs - A goal oriented approach. *Data Knowl. Eng., 68*(7), 579-598.

EMF. (2011). The Eclipse Modeling Framework Project. 2011, from http://www.eclipse.org/emf

Escalona, M. J., & Aragón, G. (2008). NDT. A Model-Driven Approach for Web Requirements. *IEEE Trans. Softw. Eng., 34*(3), 377-390.

Escalona, M. J., & Koch, N. (2004). Requirements engineering for web applications-a comparative study. *Journal of Web Engineering, 2*, 193-212.

Garrigós, I. (2008). *A-OOH: Extending Web Application Design with Dynamic Personalization.* Unpublished Ph. D, University of Alicante.

Garrigós, I., Mazón, J.-N., & Trujillo, J. (2009). *A Requirement Analysis Approach for Using i* in Web Engineering*. Paper presented at the Proceedings of the 9th International Conference on Web Engineering.

Ginige, A. (2002). *Web engineering: managing the complexity of web systems development*. Paper presented at the Proceedings of the 14th international conference on Software engineering and knowledge engineering.

GMP. (2011). The Graphical Modeling Project. 2011, from http://www.eclipse.org/emf

Gómez, J., Cachero, C., & Pastor, O. (2000). Extending a conceptual modelling approach to web application design. *Lecture Notes in Computer Science, 1789*, 79-93.

Gupta, C., Singh, Y., & Chauhan, D. S. (2010). Dependency based Process Model for Impact Analysis: A Requirement Engineering Perspective. *International Journal of Computer Applications IJCA, 6*(6), 28-33.

Horkoff, J., & Yu, E. (2009a). Evaluating Goal Achievement in Enterprise Modeling–An Interactive Procedure and Experiences. *The Practice of Enterprise Modeling*, 145-160.

Horkoff, J., & Yu, E. (2009b). *A Qualitative, Interactive Evaluation Procedure for Goal-and Agent-Oriented Models.* Paper presented at the 21st International Conference in Advanced Information Systems Engineering (CAiSE), Amsterdam, The Netherlands.

JAVA. (2011). JAVA Standar Edition (SE). 2011, from http://www.oracle.com/technetwork/java/javase/overview/index.html

Jouault, F. (2005). *Loosely coupled traceability for atl.* Paper presented at the European Conference on Model Driven Architecture (ECMDA) workshop on traceability, Nuremberg, Germany.

Koch, N., Knapp, A., Zhang, G., & Baumeister, H. (2008). Uml-Based Web Engineering. In G. Rossi, O. Pastor, D. Schwabe & L. Olsina (Eds.), *Web Engineering: Modelling and Implementing Web Applications* (pp. 157-191): Springer London.

Lindvall, M., & Sandahl, K. (1998). How well do experienced software developers predict software change? *J. Syst. Softw., 43*(1), 19-27.

Melia, S., Cachero, C., & Gomez, J. (2003). *Using MDA in web software architectures.* Paper presented at the Proceedings of the 2nd International Workshop on Generative Techniques in the Context of MDA., Anaheim, California, USA.

Molina, F., Pardillo, J., & Toval, A. (2008). *Modelling Web-Based Systems Requirements Using WRM*. Paper presented at the Proceedings of the 9th international workshops on Web Information Systems Engineering.

Molina, F., & Toval, A. (2009). Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems. *Advances in Engineering Software, 40*(12), 1306-1317.

Nuseibeh, B., & Easterbrook, S. (2000). *Requirements engineering: a roadmap*. Paper presented at the Proceedings of the Conference on The Future of Software Engineering.

Pastor, O., Fons, J., Pelechano, V., & Abrahão, S. (2006). Conceptual Modelling of Web Applications: The OOWS Approach. In E. Mendes & N. Mosley (Eds.), *Web Engineering* (pp. 277-302): Springer Berlin Heidelberg.

Schwabe, D., & Rossi, G. (1995). The object-oriented hypermedia design model. *Communication of the. ACM, 38*(8), 45-46.

Sommerville, I. (2005). *Software Engineering* (7th ed.): Addison Wesley.

Troyer, O. M. F. D., & Leune, C. J. (1998). WSDM: a user centered design method for Web sites. *Comput. Netw. ISDN Syst., 30*(1-7), 85-94.

WebREd. (2011). The Web Requirements Editor. 2011, from http://code.google.com/p/webred/

Yu, E. (1995). *Modelling strategic relationships for process reengineering*. Unpublished phD, University of Toronto Ontario, Canada.

Yu, E. (2002). *Towards modelling and reasoning support for early-phase requirements engineering*. Paper presented at the 10th Requirements Engineering (RE).

Zhang, W., Mei, H., & Zhao, H. (2005). *A Feature-Oriented Approach to Modeling Requirements Dependencies*. Paper presented at the Proceedings of the 13th IEEE International Conference on Requirements Engineering.

Zhao, J. (2002). *Change impact analysis for aspect-oriented software evolution*. Paper presented at the Proceedings of the 7th International Workshop on Principles of Software Evolution.