A Goal-Oriented Approach for Optimizing Non-functional Requirements in Web Applications

José Alfonso Aguilar, Irene Garrigós, and Jose-Norberto Mazón

Lucentia-DLSI
University of Alicante, E-03080, San Vicente del Raspeig, Alicante, Spain
{ja.aguilar,igarrigos,jnmazon}@dlsi.ua.es

Abstract. Web design methodologies should be able to provide a requirements analysis stage to consider the large and heterogeneous audience of Web applications. In this stage, non-functional requirements (NFRs) should be addressed in order to improve the quality of the Web application perceived by users. To this aim, different configurations of requirements could be implemented depending on the NFRs preferred by the Web user. Furthermore, prioritizing and making tradeoffs between NFRs is crucial for satisfying the audience of Web applications. Therefore, this work presents an algorithm based on the Pareto optimal approach to evaluate and select the optimal configuration of requirements for a Web application. To do this, the NFRs are maximized according to a priority list provided by the audience. Our approach is illustrated with a running example.

Keywords: non-functional requirements, Web engineering, goal-oriented requirements engineering.

1 Introduction

Unlike traditional stand-alone software, the audience of Web applications is both open and large. Therefore, users may have different goals and preferences and stakeholders (in this context, stakeholders are individuals or organizations who affect or are affected directly or indirectly by the development project in a positive or negative form [9]) should be able to cope with these heterogeneous needs by means of an explicit requirements analysis stage in which functional and non-functional requirements are considered [2].

Functional requirements (FRs) describe the system services, behavior or functions, whereas non-functional requirements (NFRs), also known as quality requirements, specify a constraint in the application to build or in the development process [7]. An effective definition of requirements improves the quality of the final product. Unfortunately, in most of the Web engineering approaches, a complete analysis of requirements is performed considering only FRs, thus leaving aside the NFRs until the implementation stage. We totally agree with [3] the argument that NFRs are a very important issue and must be considered from

the very beginning of the development process, to be analyzed in depth, in order to improve the quality of the Web application perceived by users.

Interestingly, the recent inclusion of goal-oriented techniques in Web requirements engineering [4] offers a better analysis in the requirements stage since requirements are explicity specified in goal-oriented models in order to support reasoning about organizational objectives, alternatives and implications, thus having a deep understanding about the domain. This has allowed the stakeholders to choose among the design decisions that can be taken to satisfy the goals and evaluate the implementation of certain requirements in particular (including NFRs). However, this is not enough to ensure that the Web application satisfies the real user needs because they do not offer mechanisms to maximize NFRs, i.e., to find a tradeoff between the FRs and NFRs.

Therefore, not paying attention to eliciting, documenting and tracking NFRs makes harder for the stakeholders to take design choices. Consequently, the quality of the Web application perceived by users will be affected negatively. Thus, NFRs need to be addressed to enable the stakeholder to choose among multiple configurations maximizing the NFRs, helping them to take design choices which positively affects the quality of the Web application, i.e., maximizing the NFRs navigability and accessibility, improves the browsing user experience.

Bearing these considerations in mind, this paper presents an algorithm that allows the stakeholder to evaluate the implementation of certain requirements considering the NFRs maximization. The algorithm is based on the Pareto optimal approach [10] , which is useful when there are multiple competing and conflicting objectives that need to be balanced. The algorithm thus constructs a group of configurations (called Pareto front) optimizing the NFRs. A configuration only can be considered on the Pareto front, if and only if, it maximizes a NFR and the other ones remain the same, or they are maximized too. From these balanced configurations, the stakeholder can select the final configuration taking into account the different NFRs from the beginning of the development process. The proposal presented in this paper is defined upon our Web engineering method A-OOH *(Adaptive Object Oriented Hypermedia method)* [1] although it can be applied to any other Web engineering approach.

The remainder of this paper is structured as follows: Section 2, presents 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 Pareto algorithm for softgoals maximization and its application (described step by step) is presented in Section 4. Finally, the conclusion and future work is presented in Section 5.

2   Related Work

In our previous work [2], a systematic literature review has been conducted for studying requirement engineering techniques in the development of Web applications. Our findings showed that most of the Web engineering approaches focus on

the analysis and design phases and do not give a comprehensive support to the requirements phase. Furthermore, the NFRs are considered in a isolated form, leaving them out of the analysis stage. In addition, we can also conclude that the most used requirement analysis technique is UML use cases and profiles. On the other side, with regard to approaches that consider NFRs from early stages of the development process, in [8] the authors propose a metamodel for representing usability requirements for Web applications. Moreover, in [3] the authors present the state-of-the-art for NFRs in a MDD (Model-Driven Development), as well as an approach for a MDD process (outside the field of Web engineering). Unfortunately, these works overlook how to maximize the NFRs.

To sum up, there have been many attempts to provide techniques and methods to deal with some aspects of the requirements engineering process for Web applications. Nevertheless, there is still a need for solutions that considers NFRs from beginning of the Web application development process, in order to assure that they will be satisfied at the same time that the functional requirements are met, improving the quality of the Web application perceived by users.

3   Specifying Requirements in Web Engineering

This section briefly describes our proposal to specify requirements in the context of a Web modeling method by using $i*$ models [6], [1]. As a goal-oriented analysis technique, the $i*$ framework focuses on the description and evaluation of alternatives and their relationships to the organizational objectives. This proposal supports an automatic derivation of Web conceptual models from a requirements model by means of a set of transformation rules.

Following, we shortly describe an excerpt of the $i*$ framework which is relevant for the present work. For a further explanation, we refer the reader to [11]. The $i*$ framework consists of two models: the strategic dependency (SD) model to describe the dependency relationships (represented as ⊸▷⊸) 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 (represented as ◌) provides a detailed way of modeling internal intentional elements and relationships of each actor (◯). Intentional elements are goals (◯), tasks (◇), resources (▭) and softgoals (◯). Intentional relationships are means-end links (⊸▷) representing alternative ways for fulfilling goals; task-decomposition links (⊸+) representing the necessary elements for a task to be performed; or contribution links ($\xrightarrow{help}_{hurt}$) in order to model how an intentional element contributes to the satisfaction or fulfillment of a softgoal. Possible labels for a contribution link are "Make", "Some+", "Help", "Hurt", "Some-", "Break", "Unknown", indicating the (positive, negative or unknown) strength of the contribution.

To adapt $i*$ framework to the Web engineering domain we use the taxonomy of Web requirements presented in [5]. Next, we have used the extension mechanisms of UML to define a profile for using $i*$ to specific Web domain terminology. Therefore, new stereotypes have been added according to the different kind of Web requirements (NFRs are modeled directly using the $i*$ softgoal element).

A sample application of the i* modeling framework for Web domain is shown in Figure 1, which represents the SR model of our running example for the Conference Management System (CMS), the complete specification of the case study can be found at: `http://users.dsic.upv.es/~west/iwwost01`. The purpose of the system is to support the process of submission, evaluation and selection of papers for a conference. It is important to highlight that each element from Figure 1 corresponds to a requirement type from the taxonomy previously mentioned, i.e., the content requirement (Content) from the taxonomy is displayed with the notation *"Resource"* from i* and the navigational (Navigational) and service (Service) requirements with the symbol *"Task"* from i*, both with their respective associations (*decomposition-links*). The extract of the CMS example is focused on the selection of the review process. Four actors participate in the CMS, but due to space limitations, for this example only the author, reviewer and system actors were considered. In this case, three actors are detected that depend on each other, namely *"Reviewer"*, *"Author"* and *"Conference Management System"*. The reviewer needs to use the CMS to *"Review paper"*. The author depends on the CMS in order to *"Paper be reviewed"*. These dependencies and the CMS actor are modeled by a SD and SR models in Figure 1.

The goal of the CMS actor is *"Process of review of papers be selected"*. To fullfill this goal, the SR model indicates 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 by means of the navigational requirement *"Blind review process"*. 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 (sofgoals hereinafter), i.e., the service requirement *"Download paper without authors' name"* needs the content requirement *"Papers"*, also, 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.

4 Optimizing NFRs in Web Applications

In this section, we present a proposal (see Figure 2) which provides support for the stakeholder in order to evaluate and decide which functional requirements have to be implemented to improve the Web application functionality while NFRs are maximizing. To do this, we extend the goal-oriented requirement approach for Web engineering, described in Section 3, with the Pareto front algorithm, in order to achieve the optimal configuration based on the softgoals maximization. Also, a set of steps in order to fully satisfy the stakeholder goals have been defined. Therefore, the effect of the implementation of a set of FR can be assessed and the best among several implementation options can be selected by prioritizing the softgoals while still satisfying the goals.

The Pareto front algorithm is useful when there are multiple competing and conflicting objectives [10] that need to be balanced. The Pareto front is a notion
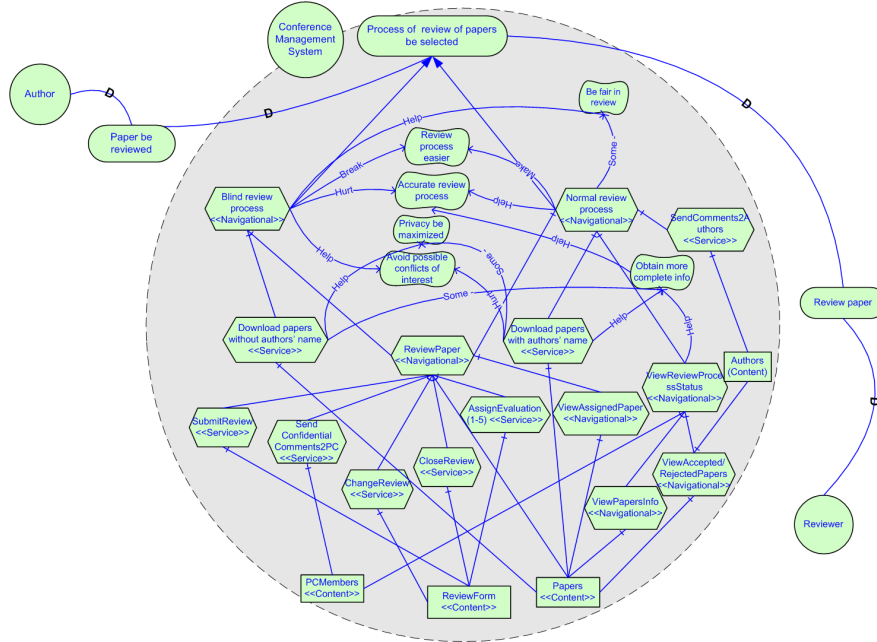
Fig. 1. Part of the Conference Management System requirements expressed in a SR and SD Models

from economics widely applied to engineering, which is described as follows: *"given a set of alternative allocations and a set of individuals, allocation A is an improvement over allocation B only if A can make at least one individual better than B, without making any other worse"*. In this sense, a set of individuals refers to the set of requirements, also, a set of alternative allocations corresponds to the state of the requirement (implemented or not implemented), and make an individual better by means of maximizing softgoals, and the opposite, means weakening softgoals. Therefore, a Pareto front, is one that no other configuration better satisfies a single softgoal, while satisfying the others equally. The set of Pareto configurations can be used to make a well-informed decision about which requirements configuration is the optimal to balance the tradeoff between softgoals.

Finding the set of Pareto optimal configuration can be defined as the problem of finding a (decision) vector of decision variables X (i.e., a valid implemented/not implemented requirements configuration), which maximizes a vector of M objective functions $f_i(X)$ (i.e., the satisfaction of softgoal i in configuration X) where $i = 1..M$ (with M the amount of softgoals). To do so, the concept of domination between vectors is defined as follows: a decision vector X is said to dominate a decision vector Y (also written $X \succ Y$) if and only if their corresponding objective vectors of objective functions $f_i(X)$ and $f_j(X)$ satisfies: $\forall i \in \{1...M\} f_i(X) \geq f_i(Y)$ and $\exists i \in \{1...M\} f_i(X) > f_i(Y)$, it is then said that

all decision vectors that are not dominated by any other decision vectors form the Pareto optimal set, while the corresponding objective vectors are said to form the Pareto front. Our approach as a running example is described next.

4.1   The Pareto Algorithm as a Running Example

Following the Pareto efficiency, we have defined the following steps to determine the Pareto optimal configuration for requirements implementation while the softgoals are balanced and maximized (see Figure 2). At the same time, these steps are applied in our running example presented in Section 3.
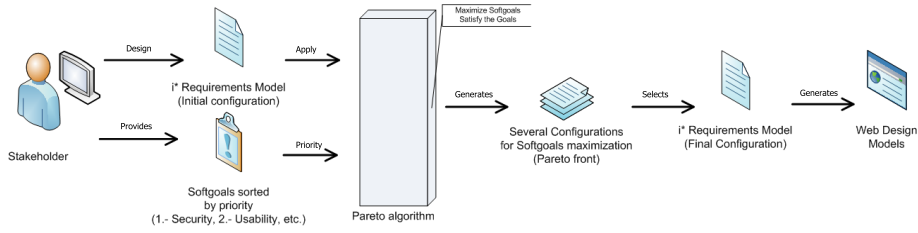

Fig. 2. Overview of the Pareto approach

Step 1. Create the Initial Requirements Model. The stakeholder creates an initial requirements model (IRM) using the $i*$ framework with which specifies the goals, softgoals and functional requirements (*Navigational*, *Service*, *Personalization*, *Layout* and *Content*) that the Web application must satisfy. This IRM specifies the requirements that must be implemented as a first prototype of the final Web application. At the same time, the stakeholder defines a list of softgoals sorted by priority, with which specifies the softgoals that the Web application must accomplish. For this running example, the requirements model is the one described in Section 3.

Step 2. Create a Requirements List. The second step consists of developing a list of the requirements (implemented or not) that contribute to any softgoal in the $i*$ requirements model (see Table 1). This table shows a requirements list 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 *"Privacy be maximized"*, "S5" *"Avoid possible conflicts of interest"* and "S6" it is the *"Obtain more complete info"*.

Step 3. Store Each Possible Requirements Configuration. In this step, each possible implementation (configuration) of N requirements is stored in a decision vector X_v: $\forall v 0 \preceq v < 2^N$, $\forall i \in \{1...N\} X_{v_i} = T_i$, where X_{v_i} is the ith element of X_v, $T_i = I$ if the requirement is implemented and $T_i = N$ if the requirement is not implemented.

Table 1. The requirements contributions to softgoals

Requirements	"S1"	"S2"	"S3"	"S4"	"S5"	"S6"
R1.- "Blind review process"	Help	Break	Hurt	-	Help	-
R2.- "Download papers without authors' name"	-	-	-	Help	-	Some -
R3.- "Normal review process"	Some -	Make	Help	-	-	-
R4.- "Download paper with authors' name"	-	-	-	Some -	Hurt	Help
R5.- "View review process status"	-	-	-	-	-	Help

Step 4. Assign a Weight to Each Contribution from Requirements to Softgoals. The contribution of each requirement (implemented or not) must be quantified. To this aim, the stakeholder creates a matrix by using the following weights to each kind of contribution: w= 0 if the requirement does not contribute to any softgoal, w= +1 if there is a Help contribution link, w= -1 if there is a Hurt contribution, w= +2 if there is a Some + link, w= -2 if the contribution is Some -, w= +4 if there is a Make and w= -4 if there is a Break contribution link.

Therefore, the matrix is defined, so that each entry $W_{i_j}^k$ corresponds to the contribution of the *ith* requirement to the *jth* softgoal on the k status (implemented or not): $\forall i \in \{1...N\}$, $\forall j \in \{1...M\}$, $\forall k \in \{I, N\}$ $W_{i_j}^k = w$, where N is the number of requirements and M is the number of softgoals, and w is defined as previously described.

For computing the objective functions in this running example, the following matrix (1) is defined containing the quantification of each requirement to softgoals, as explained in the previous section. As an example, row 3, (requirement "Normal review process"), column 2 (softgoal "Review process easier"), shows "+4" in the matrix, indicating a "Make" contribution if the requirement is implemented, and on the other side, if the requirement "Blind review process" (row 1) is implemented, column 2 will be indicating "-4" in the matrix, this means a "Break" contribution.

$$M_{i_j}^k = \begin{pmatrix} +1 & -4 & -1 & 0 & +1 & 0 \\ 0 & 0 & 0 & +1 & 0 & -2 \\ -2 & +4 & +1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & -1 & +1 \\ 0 & 0 & 0 & 0 & 0 & +1 \end{pmatrix} \tag{1}$$

Step 5. The Objective Function. For each softgoal j the corresponding objective function F_j with respect to a decision vector X_v is calculated by summing the contributions of all requirements to each softgoal j taking into account the requirements configuration defined in x_v: $\forall j \in \{1...M\}$, $\forall v 0 \preceq v < 2^N$ $F_j(X_v) = \Sigma_{j=1}^M W_{i_j}^k$, where N is the number of requirements, M is the number of softgoals.

Finally, the sum of all objective functions with respect to a decision vector X_v is computed to obtain the overall fitness of the decision vector X_v: $\forall j \in \{1...N\}$, $\forall v 0 \preceq v < 2^N$ $\Sigma_{j=1}^M F_j(X_v)$, where N is the number of requirements and M is the number of softgoals.

Table 2 shows all possible decision vectors (column 2 to 6, all rows), in other words, all possible requirements configurations, where "I" represents the status "Implemented" and "N" represents "Not implemented". The results of the corresponding objective functions are shown in columns 7 to 12, and the overall

Table 2. The posible requirements to implement or not for the softgoal tradeoff

Configuration	R1	R2	R3	R4	R5	F(S1)	F(S2)	F(S3)	F(S4)	F(S5)	F(S6)	Pareto front
X1	I	I	I	I	I	-1	0	0	-1	0	0	No
X2	I	I	I	I	N	-1	0	0	-1	0	-1	No
X3	I	I	I	N	I	-1	0	0	1	1	-1	Yes
X4	I	I	I	N	N	-1	0	0	1	1	-2	No
X5	I	I	N	I	I	1	-4	-1	-1	0	0	Yes
X6	I	I	N	I	N	1	-4	-1	-1	0	-1	No
X7	I	I	N	N	I	1	-4	-1	1	1	-1	Yes
X8	I	I	N	N	N	1	-4	-1	1	1	-2	No
X9	I	N	I	I	I	-1	0	0	-2	0	2	Yes
X10	I	N	I	I	N	-1	0	0	-2	0	1	No
X11	I	N	I	N	I	-1	0	0	0	1	1	Yes
X12	I	N	I	N	N	-1	0	0	0	1	0	No
X13	I	N	N	I	I	1	-4	-1	-2	0	2	Yes
X14	I	N	N	I	N	1	-4	-1	-2	0	1	No
X15	I	N	N	N	I	1	-4	-1	0	1	1	Yes
X16	I	N	N	N	N	1	-4	-1	0	1	0	No
X17	N	I	I	I	I	-2	4	1	-1	-1	0	Yes
X18	N	I	I	I	N	-2	4	1	-1	-1	-1	No
X19	N	I	I	N	I	-2	4	1	1	0	-1	Yes
X20	N	I	I	N	N	-2	4	1	1	0	-2	No
X21	N	I	N	I	I	0	0	0	-1	-1	0	No
X22	N	I	N	I	N	0	0	0	-1	-1	-1	No
X23	N	I	N	N	I	0	0	0	1	0	-1	Yes
X24	N	I	N	N	N	0	0	0	1	0	-2	No
X25	N	N	I	I	I	-2	4	1	-2	-1	2	Yes
X26	N	N	I	I	N	-2	4	1	-2	-1	1	No
X27	N	N	I	N	I	-2	4	1	0	0	1	Yes
X28	N	N	I	N	N	-2	4	1	0	0	0	No
X29	N	N	N	I	I	0	0	0	-2	-1	2	Yes
X30	N	N	N	I	N	0	0	0	-2	-1	1	No
X31	N	N	N	N	I	0	0	0	0	0	1	Yes
X32	N	N	N	N	N	0	0	0	0	0	0	No

fitness for each decision vector is shown in column 13. Finally, in the last column, we indicate if the corresponding decision vector is in the Pareto front. Grey rows are the Pareto front.

Step 6. Maximize the Softgoals and Still Satisfaying the Goals. In this step the stakeholder creates a list of softgoals sorted by priority (the softgoals priority was stablished in the list from *Step 1* by the stakeholder) and a list of goals that the Web application has to achieve. For this case, the softgoals priority list is shown in Table 3.

Table 3. Softgoals priority list for achieve the goal *"Process of review of papers be selected"*

Order	Softgoal
1	*"S4.- Privacy be maximized"*
2	*"S2.- Review process easier"*
3	*"S3.- Accurate review process"*
4	*"S1.- Be fair in review"*
5	*"S5.- Avoid possible conflicts os interest"*
6	*"S6.- Obtain more complete info"*

Step 7. Select the Pareto Optimal Configuration. Finally, according to Table 3, the two most important softgoals to maximize are "S4" and "S2". Therefore, it is necessary to select the final solution according to the priorities established over the softgoals.

First of all, it is necessary to select the configurations that besides being Pareto front satisfy the goal *"Process of review of papers be selected"*, for this running example, the configurations "X3", "X7", "X17" and "X25" are the only ones that satisfy the goal from all the 14 configurations that are Pareto front (see Table 2). Importantly, this step could be done in *Step 5*, i.e., calculating the

objective function for those configurations that satisfy the goals, but not doing it in this form allows us to select between different configurations considering only the softgoals maximization, leaving aside the *goals*, this gives a wider scope to the stakeholder for the final implementation.

The next step consists in selecting from the configurations that are Pareto front and satisfy the goal, the ones that maximize the softgoals according with the list from Table 3. To do this, it is necessary to check all the configurations with the requirements model to select the configurations that allow to achieve the goal (in this case there are two paths, i.e., two means-ends links), these are "X3", "X7", "X17" and "X25" . Then, it is necessary to select the best option according to the softgoals to maximize. For the softgoal "S4", "X3" and "X7" are the configurations which its overall is maximized and, for the softgoal "S2" are "X17" and "X25".

For this running example, the configuration "X3" is the best option, because according with the priority list, "S4" and "S2" are the softgoals to prioritize. The configurations "X17" and "X25" maximize "S2", however the contributions of both to softgoal "S4" (which is the number one from the priority list) are -1 and -2 (see Table 2). Furthermore, besides that the configuration "X3" has an overall fitness of $+1$ for "S4" as same as the configuration "X7", the configuration "X3" has an overall fitness of 0 for "S2" and, "X7" has an overall fitness of -4 for "S2", resulting more affected that the configuration "X3" (see Table 2), with which indicating that optimizing security comes at a high cost with respect to other softgoals (usability). The rest of solutions of the Pareto front are intermediate configurations that lead us to different tradeoffs.

Finally, the final requirements model (FRM) is the configuration "X3" (see Table 2). Therefore, the requirements "R1.- Blind review process", "R2.- Download papers without authors name", "R3.- Normal review process" and "R5.- View review process status" must be implemented in order to maximize the softgoals "S4.- Privacy be maximized" and "S2.- Review process easier". In "X3" only "R4" is not implemented. These requirements enable alternative paths (means-ends links) to satisfy the goal.

5   Conclusion and Future Work

In this work, we have presented an extension to our goal-oriented requirements analysis approach for the development of Web applications. Our approach allows the stakeholder to evaluate and decide which requirements configuration to implement. To do so, we devised an algorithm based on the Pareto optimal approach that is particularly suited to balance and maximize the conflicting softgoals. Furthermore, it facilitates the evaluation of the obtained (Pareto) optimal solutions and the selection of the final solution taking into account the priorities of softgoals. To do this, it includes weighted contributions according to the importance of softgoals, in order to further help the stakeholder to balance and optimize the different softgoals. Future work consists in the integration of

our goal-oriented approach for requirements analysis and the Pareto algorithm in a MDD solution for the development of Web applications, within the A-OOH approach.

References

1. Aguilar, J.A., Garrigós, I., Mazón, J.N., Trujillo, J.: An MDA Approach for Goal-Oriented Requirement Analysis in Web Engineering. J. Univ. Comp. Sc. 16(17), 2475–2494 (2010)
2. Aguilar, J.A., Garrigós, I., Mazón, J.N., Trujillo, J.: Web Engineering Approaches for Requirement Analysis- A Systematic Literature Review. In: 6th Web Information Systems and Technologies (WEBIST), vol. 2, pp. 187–190. SciTePress Digital Library, Valencia (2010)
3. Ameller, D., Gutiérrez, F., Cabot, J.: Dealing with Non-Functional Requirements in Model-Driven Development. In: 18th IEEE International Requirements Engineering Conference (RE), pp. 189–198. IEEE, Los Alamitos (2010)
4. Bolchini, D., Paolini, P.: Goal-Driven Requirements Analysis for Hypermedia-Intensive Web Applications. J. Req. Eng. 9(2), 85–103 (2004)
5. Escalona, M.J., Koch, N.: Requirements Engineering for Web Applications - A Comparative Study. J. Web Eng. 2(3), 193–212 (2004)
6. Garrigós, I., Mazón, J.N., Trujillo, J.: A requirement analysis approach for using i* in web engineering. In: Gaedke, M., Grossniklaus, M., Díaz, O. (eds.) ICWE 2009. LNCS, vol. 5648, pp. 151–165. Springer, Heidelberg (2009)
7. Gupta, C., Singh, Y., Chauhan, D.S.: Dependency Based Process Model for Impact Analysis: A Requirement Engineering Perspective. J. Comp. App. 6(6), 28–30 (2010)
8. Molina, F., Toval, A.: Integrating Usability Requirements that can be Evaluated in Design Time into Model-Driven Engineering of Web Information Systems. J. Adv. Eng. Softw. 40, 1306–1317 (2009)
9. Sommerville, I.: Software Engineering, 6th edn. Addison-Wesley, Reading (2001)
10. Szidarovszky, F., Gershon, M., Duckstein, L.: Techniques for Multiobjective Decision Making in Systems Management. Elsevier, Amsterdam (1986)
11. Yu, E.S.K.: Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering. In: 3rd IEEE International Symposium on Requirements Engineering (RE), p. 226. IEEE, Washington, DC, USA (1997)