

Capturing Web Application Requirements through Goal-Oriented Analysis

Davide Bolchini, Paolo Paolini

¹ University of Lugano, Faculty of Communication Sciences, TEC-lab,
via G. Buffi 13 - 6900 Lugano – CH
davide.bolchini@lu.unisi.ch, www.tec-lab.ch

² Politecnico di Milano, Dipartimento di Elettronica e Informazione,
Via Ponzio, 34/5- 20133 Milano
paolo.paolini@polimi.it, <http://hoc.elet.polimi.it>

Abstract. New conceptual tools for effectively supporting the activity of requirements analysis of web applications are needed. Current approaches and model in Requirements Engineering (RE) can be tailored and extended in order to cope with the following distinctive features of interactive and hypermedia-intensive web applications: a) high-level communication and business goals have to be addressed and carefully analyzed as an integral part of requirements management; b) goals and needs of the stakeholders need to be tied up with design in a coherent fashion, bridging the current gap between requirements and hypermedia specifications; c) lightweight, usable and informal models are needed for web designers and analysts with no engineering background. Starting from key achievements in RE this paper introduces a light-weight methodology blending goal-directed RE and scenario-based techniques. The core elements of the methodology will be illustrated through a real project experience.

1 Introduction and Motivation

Requirements Engineering (RE) refers to the activities intended at assuring that a software system fulfills the goals, the needs and the expectations of all the relevant stakeholders [15]. In particular, RE involves a set of intertwined tasks [14] [12]: a) requirements elicitation; b) analysis and modeling; c) negotiation and validation. In particular, the requirements analysis aims at prune the material gathered during elicitation (goals and expectations of the stakeholders, constraints on the system) and operationalizing it into requirements specifications for the system to build.

Being the cornerstone of software development, a number of comprehensive models, methodologies and notations for managing the analysis of software requirements have been developed and assessed in the last decade [10].

However, little attention in RE has been paid to the development of adequate frameworks and methodologies for coping with the requirements analysis of Web applications. Web applications are today first and foremost articulated means of communication between the end-users and the stakeholders who conceived the site.

Poor requirements augment the risk of missing the opportunity of meeting customers' needs and enhancing the user experience [13]. In fact, in order to be successful on the market, a web application has to be stakeholder-centered. On one hand it has to offer content and interaction capabilities which best meet the goals of the users. On the other hand, it has to satisfy as well as possible the business and communication objectives of the site itself. Since web applications are growing more and more in complexity, both in size and in service sophistication, the design and the development of these applications should borrow design approaches, methodologies, practices and techniques from the long-standing tradition of software systems engineering. Having said that, such approaches need to be carefully tailored on the distinctive features of web application requirements analysis, taking into account the potential users of such techniques (web analysts and designers) and the high-degree of hypermedia and interactivity characterizing web sites. The specific needs of requirements management for web application can be summarized as follows.

1. Managing complexity. High-level communication and business goals have to be addressed and carefully analyzed as an integral part of requirements management.
2. Requirements-Design gap. Goals and needs of the stakeholders need to be tied up with design in a coherent fashion, bridging the current gap between requirements and hypermedia specifications [8].
3. Accessibility. Junior and less-experienced designers (often with no engineering background) need informal, lightweight, straightforward methods and best practices for dealing effectively with web site requirements [16] [17].

Besides these web-specific features calling for ad-hoc conceptual tools for web requirements engineering, there are other two general RE issues that should be addressed throughout the requirements management tasks:

- a. Traceability [18]. Web applications frequently changes requirements, both for meeting the volatile market's demand, both for fulfilling changing customers' needs and goals. In order to manage effectively the changing requirements and the evolution of the application, a proper requirements management process should keep traces of the choices taken about requirements by highlighting the reasons for which certain requirements have been defined. In this way, the "Why" behind the decisions about requirements should be clearly documented and explicated. Keeping track of the rationality of requirements definition needs to support both backward traceability and forward traceability. Backward traceability allows seeing which goals a given requirement is a possible fulfillment of. Forward traceability allows following the evolution of a requirements from its rationale to the design artifacts it has impact on.

- b. Suspension of commitment [5]. Web designers often have to cope with ill-structured and vague requirements, which are typical in web site development. In this situation, they risk to imagine and focus on a design solution even before having understood the problem to solve. Requirements analysis should be helped to reflect on the problem in order to understand as good as possible the relevant needs and the issues to solve before ossifying the creativity on a given design artifact. Accordingly, during requirements analysis, one should be able to exploring alternative solutions, before claiming to having envisioned the best one.

Extending and adapting to the web domain some of the basic achievements in RE (namely goal-oriented approaches and scenario-based techniques), this paper introduces a novel model for systematically carrying out the requirements analysis for

web applications. Currently the model – developed within the EU-funded UWA project¹ - offers four elements for supporting the activity of web analysts: a) a light-weight methodology; b) an intuitive UML-based notation; c) a set of heuristic principles; d) a Rational Rose-based analysis tool. This paper will present the core elements of the methodology and of the notation through a real case study.

2 Background

The activity of requirements analysis has been deeply studied in the field of Software Engineering. One of the most promising RE approaches that could be taken into account for web application development is the goal-based analysis. The approach that most systematically considers the goal of the systems under construction is the van Lamsweerde's goal-oriented requirements engineering formal model, known as KAOS [1]. According to this approach, goals represent the targets of achievement for the system. Goals are formally defined and analyzed in such a way that conflicts between goals are pointed out. Goals are then systematically decomposed in subgoals through a refinement process modeled in AND/OR graphs. Finally, goals are eventually operationalized into requirements, which describe the characteristics the system should match. KAOS had the great merit of introducing a goal-oriented attitude in requirements engineering, which turns out to be a crucial support for building systems trying to satisfying the requirements. This formal methodology is effective for an accurate and consistent definition of requirements of safety-critical systems (e.g. software for railway systems, flight traffic control, etc.). Traceability in KAOS is ensured by the documentation of the refinement process. On the other hand, suspension of commitment is allowed as well because goal-directed reasoning force to reflect about the problems to be solved and the target of achievements before defining a design solution.

However, KAOS seems to be not sufficient for analyzing the goals and the expectations of the stakeholders of a web application, mainly for two reasons:

- a) it considers first and foremost the goals the system should achieve. We are instead interested in capturing and documenting explicitly the communication and business goals owned by the stakeholders and only then deriving web site requirements;
- b) the requirements defined by KAOS are very easily mapped onto software components and modules. The level of detail of the requirements is really too fine-grained for web services. Web designers need to express the requirements in a proper level of abstraction (independent from any design or implementation techniques) as to not prematurely commit their selves on design decision.
- c) the formalism adopted seems to be not effective in web site development. Formal methods risk to be hardly accessible and usable by junior web designers or by analysts which have no engineering background.

¹ UWA project – Ubiquitous Web Application (IST-2000-25131) – www.uwaproject.org.

On the basis of KAOS, other goal-oriented frameworks and notations (such a NFR, I*, etc.) have been developed in order to capturing non-functional requirements through a goal-based analysis [10][11].

A different goal-directed approach to requirements analysis is the GBRAM framework (Goal-Based Requirements Analysis Model) [3][19], defined specifically for software-intensive information systems. GBRAM combines goal-based methods (used at a higher level of abstraction) with scenario-based approaches. The activity of goal-refinement and analysis is supported by a set of heuristics in order to help analysts of information systems to their job effectively. Goals, requirements and scenarios played together for capturing most of the complexity characterizing the activity of requirements analysis. A further positive feature of GBRAM is that it describes the goals and the requirements in semi-formal way. Therefore, GBRAM can be considered a methodology that serves as a good starting point for developing a novel requirements analysis framework for web applications. The reasons why GBRAM cannot be applied to the web development so as it is, rely in the fact that this model is focused on the information system domain. It does not address tools for capturing and specifying requirements for highly interactive, navigational and hypermedia-intensive artifacts as web sites are.

As crucial milestone in the field, it is necessary to mention that the requirements engineering community and industry has adopted UML [4] [9] for documenting the entire development process of large software systems. As far as requirements are concerned, UML proposes a set of concepts and a standard notation. However, it seems not to be completely effective for capturing all the features of web application requirements. Some of the reasons are the following:

- a) UML actors identify anything (object or subject) interacting with the system: we need instead to model stakeholders who have goals with respect to the system².
- b) (b) UML use cases are effective for the functional analysis: high-level goal-analysis instead remains poorly addressed.
- c) (c) Hypermedia features (especially navigation requirements) of web applications need a specific requirements definition that UML does not capture because it is general and domain-independent.

A further research branch that can be relevant for the research at issue can be found in Human-Computer Interaction studies. In this field, the activity of requirements analysis has been explored through the notion of scenario [5][20]. As effectively defined by Carroll [5], a scenario is a “story about use”. They describe concretely typical user profiles and their behavior during the interaction of the application. Thus, scenarios are malleable and inexpensive tools that greatly help to envision requirements and explore alternative possible design solutions. Being a narrative of specific individuals using the application in specific circumstances, they represent a powerful technique widely employed for the elicitation of requirements, for the design and evaluation of interactive artifacts [13].

In the next section, we define a requirements analysis framework (the UWA Requirements model) blending goal-oriented techniques with scenarios in order to model the requirements necessary to build effective web applications.

² Some stakeholders identified in the requirements analysis might not to interact with the application.

3 UWA Requirements: a model for the analysis of web application requirements

3.1 An example

In order to present the main components of the model let us take an example of a real application [6]³: the web-based credit-card service of Banca121 (B121), one of the first web-enhanced banks in Italy. The actual design and development of this application allowed to assess and refine the model for requirements analysis (called UWA Requirements) envisioned within the UWA project and briefly presented in this work.

B121 wished to build a web-based service for its credit card center. From the discussion with the managers of the bank, a quite rich picture of the needs and goals for the service arose. In general, the application involves several people, product managers, salesman and different types of end-users. All these persons have particular interests in the success of the application: they are stakeholders. Specifically, these stakes can be seen as goals that each stakeholder owns with respect to the application under analysis.

The bank customer might want to decide what kind of credit card to buy or he has not even decided whether to buy or not. The strategy manager on his part would like to build and maintain the trust of the bank customers; moreover, he has the commitment of increasing the number of customers and keeping the B121 perceived image of being the most technologically-advanced bank. The generic user, who is not yet a customer of the bank, would like to assess the credibility and the quality of service of the bank. Besides, even this kind of user – typically a web surfer - has not yet decided what to buy and would like to be convinced about the convenience of the bank's offers. The picture gets even more complex when other actors come into the scene: there are customers who have already a credit card. These clients might want to upgrade their card, to manage its features or to be persuaded of buying some other product of interest.

3.2 From Goals to requirements

How to respond to all these diverse – often conflicting - and correlated needs? How can the application be designed in order to allow these stakeholders to accomplish their goals? Using the UWA Requirements model goals have been pruned and decomposed into subgoals. For example, in order to decide whether to buy or not (goal), the bank customer might want to consider the bank's offers (subgoal) and the special suggestions that the bank has customized for his profile (subgoal). Subgoals

³ The requirements analysis activity for large web communication projects is very complex and articulated. Moreover, the documentation produced is huge and often difficult to view synoptically. In this paper, a short and simplified excerpt of a real case study can illustrate the main elements of the model.

are detailed and eventually refined into functional requirements, i.e. the main features of the service under construction⁴. In this case, examples of requirements are: Access Customized Card Catalogue, Relate a Card to the Offer, Show Offer Description). While goals express the long-term needs, expectations of the stakeholders, requirements express in a sufficiently high-level how the application will try to fulfill those goals. The concepts explained above can be expressed synoptically with the graphical representation employed in Fig. 1.

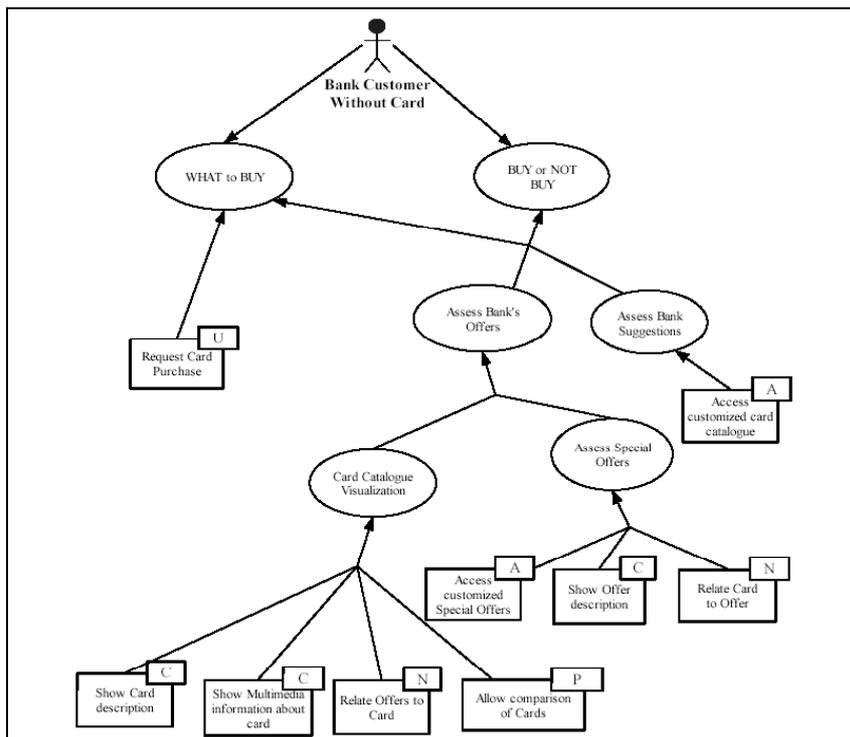


Fig. 1. Goal-analysis for the user Bank Customer without Card.

Decomposing goals into sub-goals and then into requirements is not an easy task . Once the high-level goals have been pruned, selected and defined, the analysts had to decide, with the help of proper elicitation techniques, what are the sub-goals that the stakeholders might want to achieve in order to satisfy their long-term objectives. Sub-goals identified are just a set of possible sub-goals fulfilling the upper goal. In other words, sub-goals and requirements are sufficient – but not necessary – solutions to allow the stakeholders accomplish their goals. The analysis model does not claim to highlight all the requirements and all the goals of the applications but only those which are held as crucial and essential to be analyzed carefully. Moreover, we should note that there are factors influencing the requirement decision that are not explicated

⁴ Goals are bound to the relative requirements through AND relations. This means that all requirements deriving from a goal must be satisfied in order to fulfill the upper goal.

in the analysis. For example, goals are refined taking into account the resources available (such as budget and time constraints). These factors, together with other assumptions⁵ and constraints about requirements and goals, have not been highlighted but surely influenced the actual pruning and decomposition of the goals. In fact, there is not a strict rule determining what factors influencing the analysis should be explicated or should remain assumptions. This is strongly dependent on the shared knowledge between the analysts and the domain expert and their skills and experience in developing web applications.

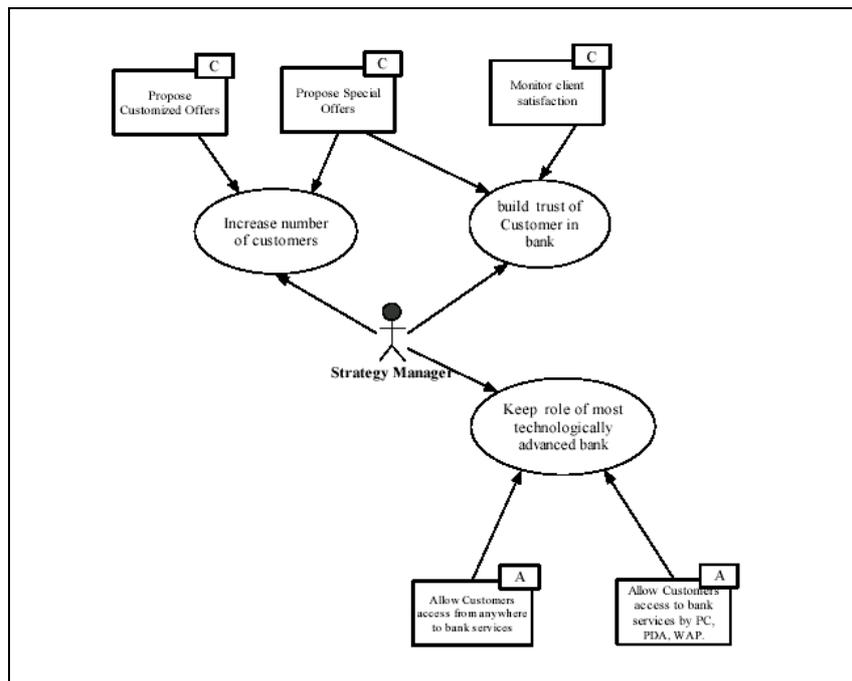


Fig. 2. Goal-analysis for the stakeholder Strategy Manager.

During the B121 requirements analysis it has been important to highlight business and communication goals that could not be given as assumptions because they represent strategic and distinctive success factors of the B121 services. In fact, the Strategy Manager is an important stakeholder whose goals have to be carefully analyzed as well as the goals of the users. The strategy manager does not have just the goal of enhancing the user experience per se; his relevant long-term objectives are related to his business strategy: increasing the customer base, building trust towards

⁵ In the literature of goal-based approaches [10], the process of analyzing goals and identifying sub-goals is often referred to as “refinement” or “derivation”. These terms could well describe processes where goals are formally expressed [1] and sub-goals can be calculated from them. However, for web applications, goals are often ill-defined and always expressed in natural language. Identifying sub-goals from goals is a human decision and not a formal derivation.

the bank and keeping the perceived image of B121 as being the most technology-enhanced bank in Italy. Given these goals, during the brainstorming sessions between analysts and design experts, requirements about content and accessibility aspects have been defined (see Figure 2).

3.3 A taxonomy for requirements

Figure 1 show the analysis graph for the stakeholder “Bank Customer without Card”. The goals (circles) owned by this user profile are decomposed and requirements (rectangles) are then defined. As shown in the figure, requirements are categorized with a label which indicates the design dimensions they have implications on. Seven categories of requirements have been so far identified: content (labeled with C), structure of content (S), access paths to content (A), navigation (N), user operation (U), system operation (O), presentation (P).

1. Content. It is the core value of any hypermedia applications. It refers to that set of ideas and messages that the site communicates to its users. Ideas and messages are mainly specified in terms of the core information objects available. Designing such information objects means designing “what” the user will actually consume and interpret along the session. In the requirements specifications of our example, a requirement concerning this dimension is “Provide abstract for each work”. In the proposed notation, requirements belonging to this dimension are labeled with “C” (see Figure 2).

2. Structure of content. Requirements can also give coarse-grain insight about how the information objects identified are structured. By “structure” we mean the organization of content within the same information object. Giving indications about the structure of content may mean basically two things: a) defining pieces of content that are part of given information object; b) highlighting pieces of content as opposed to others within the same information object. For example, in case that a work had been structured into several information components, a requirement falling into this category could be: “highlight title and abstract”. In the proposed notation, requirements belonging to this dimension are labeled with “S”.

3. Access paths to content. This dimension refers to the navigational paths available to the user in order to reach the needed content. The user should be allowed to access the needed information or be guided in the exploration of the offered content following the navigational access paths best corresponding to their expectations and goals. This design dimension captures the hypermedia artifacts exploited by the user to start the navigation, to locate and reach the interested content. Examples include: “access published works by author, by paper title, and by theme”. In the proposed notation, requirements belonging to this dimension are labeled with “A” (see Figure 2).

4. Navigation. Requirements can suggest connecting different information objects allowing the user to navigate from one piece of content to another. Semantic relationships among pieces of information can be relevant for navigation i.e., they can be exploited by the user to traverse the path connecting one object to one or more others in order to complete their cognitive or operational task. This design dimension

captures the hypermedia artifacts exploited by the user to navigate, after having accessed a given information object, from that object to one or more other semantically related artifacts. For example: “relate a work to all its references,” “relate a work to the details of its authors”, “allow navigating from an author to all the works written by that author”. In the proposed notation, requirements belonging to this dimension are labeled with “N” (see Figure 2).

5. Presentation. Requirements can also give guidelines for defining the visual communication strategies for presenting content, navigational capabilities, and operation to the user. Presentation implies two main aspects: graphics and layout. Graphics are concerned with the visual elements composing the user interface (buttons, icons, images, font proportions, titles, etc.). Layout is concerned with the physical positioning of these objects on the page. In the proposed notation, requirements belonging to this dimension are labeled with “P”.

6. User operation. User operations are those operations that are visible to users. They are the only operations the users must be aware of. Roughly, these operations are all the operations that users can enable by interacting with the application. For example: “download paper,” “post review,” “submit comment,” “post personal data,” etc. In the proposed notation, requirements belonging to this dimension are labeled with “U”.

7. System operation. System operations are those operations that are not visible to users, but are performed for system’s purposes or become mandatory to “build” user operations. For example: “force authentication”, “Induce user preferences from recurrent navigational paths”. In the proposed notation, requirements belonging to this dimension are labeled with “O”.

A requirement belongs to exactly one dimension. Actually, this restriction can also be seen as a necessary (although certainly not sufficient) condition for a requirement to be considered such: if a requirement cannot be easily and clearly assigned to exactly one dimension, then it is too general to be called a requirement (and is therefore still a goal). Again, the number and nature of dimensions is not fixed a priori, but new ones can be added at will and at any time.

Categorizing requirements is a common activity in requirements engineering. The main reason to define requirement taxonomy is that, since requirements specifications are the input for the design activity, requirements can be in this way better organized and passed over to the designer in a coherent fashion. This taxonomy allow to classify requirements according to the application dimension that they will likely influence during design. However, it is important to remember that functional requirements are not design decision. They should not introduce premature design solutions, but they should rather clearly express a design problem that will be solved by the designers⁶.

During the analysis, non-functional requirements surfaced. The hypermedia taxonomy for requirements allows also dealing with non-functional requirements more systematically. Even non-functional requirements can be categorized under the same classification. In the example, non-functional requirements such as

⁶ Taking the set of requirements as input, designers can then employ any design methodology (e.g. HDM, OOHD, WebML, RMM, WSDM and other structured models.) or any informal design method to create and shape design solutions. Surely, if designers can master a structured and systematic design method they can more easily be guided by the requirements specification.

effectiveness, orientation, accessibility, self-evidence, predictability are salient for the navigational (N) aspects of the web site of the bank. Requirements such as completeness, authority and accuracy are taken into account for the content (C). Clearness, Animation control and Consistency have been instead considered for the presentation (P).

In this way, designers are facilitated in their work: they receive functional requirements organized according to a useful taxonomy; for each requirements category, they can assess the quality of the design solutions (deriving from the functional goal-analysis) by respecting the non-functional requirements.

3.4 Goals and scenarios as mutually supportive

In order to improve the analysis, user scenarios have also been defined throughout the whole refinement process. Even if they were often not written but just envisioned with the customers, scenarios helped to uncover hidden goals or to remove some others [19]. The power of scenarios is their capability of representing goals more concretely and easy to understand. Scenarios have been negotiated and defined by analysts together with the representatives of the stakeholders in order to: 1) define or re-define stakeholders; 2) define, re-define and refine goals; 3) assess and complete the derivation graph.

An example of user scenario for the B121 application is the following:

Mr. Rossi is a university student who has been B121 customer for 1 year. He does know quite well the B121 current web site because he uses it on a regular weekly basis for checking his account. Mr. Rossi has not yet got a B121 credit card and he has never got a credit card before, but is now willing to ask for one. He would like to visit the B121 web site in order to choose the credit card most suitable to his needs and usage habit. In fact, Mr. Rossi usually spends not more than 500 euros per month and would like to have an automatic notified charging on his account. Therefore, he looks for the credit card offers for students, hoping to find easily the product best corresponding to his profile.

Such a scenario can be actually employed in two directions:

- a) it can help to refine the high-level goals into sub-goals and then into requirements, giving a concrete example of intentions of use. From the scenarios, the goal graph can be defined by abstraction.
- b) It can help to assess and validate the goal graph, by describing a concrete walkthrough in the goal-refinement process.

In the case study (see Figure 1), the particular scenario described above played the role b). In fact, once the goal-graph had already been defined in a preliminary phase, the scenario allowed to envision a detailed and plastic picture of stakeholder “Customer without card“ who owns the goals: “What to buy”, then refined into “Assess bank suggestions”.

This example shows an effective interplay between goal-oriented analysis and scenario-based techniques. In fact, as well described by Sutcliffe [12], the use of scenarios complements goal modeling because, whereas goals focus on abstractions that describe users’ intentions, scenarios make abstract intentions clearer, by giving concrete examples of how the application might fulfill users’ goals.

4 Closing Remarks

So far, the basic elements of UWA Requirements model for web application requirements analysis have been illustrated through a case study. To sum up, they are: stakeholder, goal, sub-goal, requirement, requirement dimension, refinement process.

We should not think that the model proposed captures most of the outcome of the requirements analysis rather than the process. In fact, the process of requirements analysis for B121 has been an on-going negotiation between the analysts, the designers and the customers. This process is a complex interplay between decisions about goals, requirements definition and scenarios. The UWA Requirements model turned out to be expressive enough to document the output of this process and to communicate it effectively to the design team and to the customers. It should be noted that the model presented in this work can be complemented by other techniques (such as UML use cases, hypermedia design models, and interface design methodologies) for the lower-level functional specifications and interaction design.

There are several claimed advantages of the UWA Requirements model with respect to other approaches. First of all, it introduces a fundamental requirements modeling primitive - the hypermedia taxonomy for requirements - , which has a twofold benefit: a) it allows capturing hypermedia and web high-level specification already during requirements; b) it helps to smoothly indicate design suggestions rather than giving *the* solution. Moreover, the proposed model is a light-weight, informal and stakeholder-understandable one: the traceability from requirements to goals and reasons behind requirements can be *easily* read in the derivation graph. The UWA Requirements model is thus a set of conceptual tools for facilitating the communication and the shaping of the common ground of reasoning between analysts and stakeholders. Actually, in our project experience, the notation and the concepts have been judged as easy to use and intuitive by the analysts and by the stakeholders. In general, we have experienced that such a model allows better requirements definition in terms of completeness and accuracy of requirements specification. Even if we have not the space for showing the design schemas and final B121 application, we noticed that the web application implemented provides all the needed functionality and content needed by the stakeholders. Using the UWA requirements model allowed facilitating the work of the designers and being able to trace the reasons of most of the design artifacts back to the goals of the stakeholders.

5 Future Work

Next actions prospected for assessing and completing the definition of the UWA Requirements model are the following:

- i) adding and empirically validating new features to the model: a) assigning a priority to each stakeholder; b) assigning a value for each relation stakeholder-goal. That could give the analyst an indication of the degree of relevance of a goal for the stakeholder.

ii) defining an heuristics of refinement principles (i.e. practical suggestions) addressed to analysts and designers for improving the quality of the outcome of the analysis. A preliminary set of heuristics have already been defined.

A CASE tool for UWA Requirements model has been implemented as an add-in of Rational Rose® editor. As far as now, the tool supports the basic features of the model. Parallel to the refinement of the methodology, current work is focusing on the empirical usability evaluation of the tool with designers and analysts.

References

1. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-directed Requirements Acquisition. *Science of Computer Programming*, 20:3–50, 1993.
2. Van Der Geest, T., *Web Site Design is Communication Design*, Benjamins, Amsterdam, 2001.
3. Antòn A. I., Potts, C., The use of goals to surface requirements for evolving systems, in *Proceedings of the 20th international conference on Software engineering*, 1998, Kyoto, Japan.
4. Conallen, J., *Building Web Applications with UML*, Addison-Wesley, 2000.
5. Carroll, J.M., *Making Use. Scenario-based design of human-computer interactions*, MIT Press, 2000.
6. UWA consortium, *Requirements and design specification for Banca121 Pilot application*, UWA project deliverable D11, available at www.uwaproject.org, 2001.
7. Garzotto, F., Baresi, L., Paolini, P., *From Web Sites To Web Applications: New Issues For Conceptual Modelling*, *Proceedings of the World Wide Web and Conceptual Modeling'00 Workshop, ER'00 Conference*, Salt Lake City, 2000, Springer.
8. Nüell, N., Schwabe, D., Vilain, P., *Modeling Interactions and Navigation in Web Applications*, *Proceedings of the World Wide Web and Conceptual Modeling'00 Workshop, ER'00 Conference*, Springer, Salt Lake City, 2000.
9. Jacobson, I., Booch, G., Rumbaugh, J. *The Unified Software Development Process*. Addison-Wesley, 1999.
10. A. van Lamsweerde. *Requirements Engineering in the Year 00: A Research Perspective*. In *Proceedings of ICSE'2000 – 22nd International Conference on Software Engineering*, Limerick, 2000. ACM Press. Invited Paper.
11. Mylopoulos, J., Chung, L., Yu, E., “From object-oriented to goal-oriented requirements analysis”, *Communications of ACM*, Vol. 42 No. 1, January 1999, 31-37.
12. Sutcliffe, A., *User-Centred Requirements Engineering*, Springer, 2002.
13. Brinck, T., Gergle, D., Wood, S.D., *Usability for the Web: Designing Web Sites that Work*, Morgan-Kaufmann, 2002.
14. Sommerville, I., Sawyer, P., *Requirements Engineering – A good practice guide*, Wiley, 1997.
15. Gause, D.C., Weinberg, G.M., *Exploring requirements: quality before design*, Dorset House, 1989.
16. UWA consortium, *Evaluation of UWA design methodology*, UWA project deliverable D13, available at www.uwaproject.org, 2001.
17. Cato, J., *User-Centred Web Design*, Addison Wesley, 2001.
18. Ramesh, B., *Factors Influencing Requirements Traceability Practice*, *Communications of the ACM*, Vol. 41, No. 12 (dec 1998), pp. 37-44.

28 WER 2002

19. Antòn, A., Goal Identification and Refinement in the specification of Software-based Information Systems, Ph.D Dissertation, Georgia Institute of Technology, Atlanta, GA, 1997.
20. Gotel, O., Finkelstein, A., An Analysis of the Requirements Traceability Problem, in Proc. of International Conference on Requirements Engineering (ICRE), IEEE CS Press.
21. Carroll, J., Scenarios and Design Cognition, IEEE RE 02 International Conference on Requirements Engineering, Essen, 2002. Potts, C., Bruns, G., Recording the reasons for design decisions, Proceedings of the 10th international conference on Software engineering, 1988, Singapore.
22. Antòn A. I., Potts, C., The use of goals to surface requirements for evolving systems, in Proceedings of the 20th international conference on Software engineering, 1998, Kyoto, Japan.
23. Potts C., Using schematic scenarios to understand user needs, Conference proceedings on Designing interactive systems: processes, practices, methods, & techniques, August 1995.
24. Cooper Interactive Design, Perfecting your personas, www.cooper.com/newsletters/2001_07/perfecting_your_personas.htm, 2002.