Requirements Interdependencies
- Moulding the State of Research into a Research Agenda

Åsa G. Dahlstedt
Department of Computer Science
University of Skövde,
Box 408, SE-541 28 Skövde, SWEDEN
asa.dahlstedt@ida.his.se

Anne Persson
Department of Computer Science
University of Skövde,
Box 408, SE-541 28 Skövde, SWEDEN
anne.persson@ida.his.se

Abstract

This paper focuses on requirements interdependencies, i.e. how requirements may relate to and affect each other. Due to these interdependencies, requirements cannot be treaded as stand alone artifacts, which affect various activities and decisions during development and maintenance.

The aim of this paper is to provide an overview of current state of research, as well as defining a research agenda for the area of requirements interdependencies. The paper also includes a first step towards a neutral classification of requirements interdependencies.

In order to provide a sound foundation for research within the area of requirements interdependencies, the first step should be to further investigate the different types of interdependencies found. The classification presented in this paper needs further elaboration and it also needs to be validated on several different requirements sets. The research agenda continuous with research issues concerning identifying, documenting and using the requirements interdependencies.

1. Introduction

This paper focuses on a phenomenon called requirements interdependencies. Most requirements cannot be treated independently, since they are related to and affect each other in complex manners [1, 2]. Actions performed based on one requirement may affect other requirements in ways not intended or not even anticipated. Interdependencies may also affect various decisions and activities during development, e.g. requirements change management [3,4], release planning [2,5], requirements management [6], and requirements implementation [7]. Despite this, little is known about the nature of requirements interdependencies, and further research is needed in order to understand this phenomenon better [5,8,9].

The amount of literature addressing requirements interdependencies is fairly small and it approaches the area from fairly different perspectives. Pohl [4] as well as Ramesh and Jarke [6] discuss requirements interdependencies as part of requirements traceability, focusing on requirements management as well as change management. The effect requirements interdependencies have on requirements selection or release planning is discussed by Karlsson et al [5], Carlshamre and Regnell [8] and Carlshamre et al [2]. Robinson et al [7] reports on requirements interaction management, which deals with identifying how requirements may affect each other's achievement.

The aim of this paper is to provide an overview of the current state of requirement interdependency research, and to compile the different views found. The aim is also to take the first step towards developing an integrated classification of interdependencies, which will support further research into this area and also help identifying fundamental problems when dealing with interdependencies in the systems development process.

2. Traceability: a Basis for Understanding Requirements Interdependencies

Requirements traceability has been acknowledged as an important part of software and information systems development [4, 10, 11] supporting various activities during the life of an information system. The topic is well explored. Judging by the large amount of literature describing both theoretical and empirical studies (see e.g. 4, 12, 10, 13[1], 14, 15, 16, 17]. Ramesh and Jarke [6] present an extended overview of the current state of research within the area, based on several years of research.

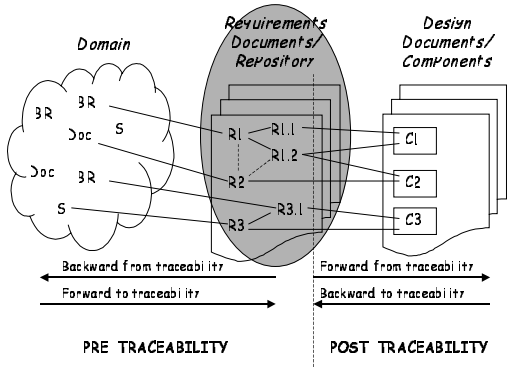

Figure 1: Different types of traceability

There are several definitions of the term traceability (see e.g. 6, 18, 19, 4, 20]. In this paper, we have chosen to define traceability as the *ability to describe and follow the life of a requirement, in both forward and backward direction, ideally through the whole system life cycle* [21, pp. 32, based on 13]. The definition indicates that requirements traceability can be divided into two main types: pre traceability and post traceability (Figure 1). Pre traceability refers to those aspects of a requirement s life before it is included in the requirements specification [13] and is focused on enabling a better understanding of the requirement. Post traceability, on the other hand refers to those aspects of a requirement s life from the point in time when it has been included in the requirements specification [13] and is focused on enabling a better understanding and acceptance of the current system/software.

[1] This paper has also been published as a book chapter in Arnold, R. and Bohner, S. Software Change Impact Analysis, IEEE Computer Society Press, 1996

Requirements pre traceability is hence concerned with *requirements production* and focuses on the *domain* with which we interact when the requirements are developed and in which the systems is to be installed. Requirements post traceability is concerned with *requirements deployment* and is focused on the *software* that is developed based on the requirements. Pre and post traceability may also be divided into four traceability types, which are presented in [22].

According to [6] traceability information provides important support within requirements engineering, design, systems evolution, and test procedures. Traceability can be used to ensure that each requirement has been satisfied, as well as to demonstrate that all system components can be related to some requirement.

The various types of traceability links presented in Figure 1 support different situations and activities during the development and maintenance of the software or information system. None of these will alone give full traceability support (see [3]). Different stakeholders are also usually interested in different types of traceability information. Despite this, current literature and standards provide few guidelines regarding which type of information should be captured and used in what context [6].

Figure 2 describes the associations between the terms related to requirements traceability. Traceability is defined as the ability to describe and follow the life of a requirement. This is ensured by recording traceability links, which shows the various relationships that exists between the information objects in the development information. Traceability can therefore be seen as a property of the development information.

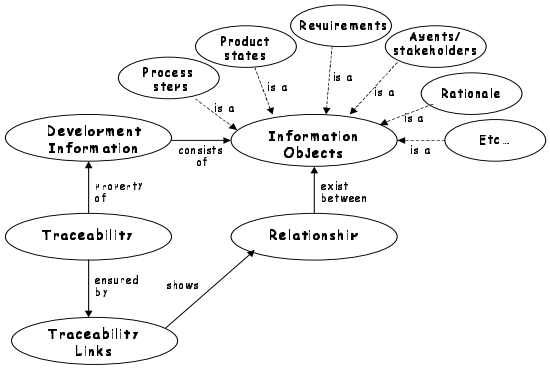

Figure 2: The terms traceability, relationship, and dependency.

Relationships can be defined as connections, associations, or dependencies between information objects, for example precondition, refines, is based on, and describes. The term dependency is used in fairly different manners by different authors. Pohl [4] has a broad view of the term and has defined 18 different dependency types (see Figure 3). Ramesh and Jarke [6], on the other hand, use the term in a more specific sense, distinguishing between dependencies and other types of relationships. This implies that the term dependency can either be seen as a synonym for the term relationship, or as a stronger connection between two objects, where the objects affect each other in some way, e.g. in case of changes. In this paper, we will not distinguish between dependency and relationship. We are interested in exploring the different manner by which requirements can relate to each other, which may mean that they affect each other as well. We have also chosen to use the term *interdependency* to emphasise that the relationships that we focus on are those that exist between information objects of the same type – namely requirements.

3. Requirements Interdependencies – Current State of Research

This section aims at provide an overview of the current state of requirements interdependencies by outline the findings from the literature as well as the findings from the interview studies.

3.1. Requirements Interdependencies a Literature Review

Requirements interdependencies is a fairly unexplored area judging by the relatively small amount of literature discussing it. However, there are some milestones within requirements interdependency research.

Within the early days of traceability research, Pohl [4] developed a traceability framework, which included a dependency model defining 18 different types of possible dependency links (see Figure 3). Pohl's model describes dependency types that can exist between any types of information object. We focus requirements interdependencies, but there are most certain some correlations between these general dependencies and requirements interdependencies, which motivate why this dependency model is relevant for our investigation.

However, Pohl's dependency model must somewhat be adapted and specialised towards *requirements interdependencies* to be useful in this context. There are some dependency types included in Pohl's model that clearly cannot exist between requirements (see [23] for description of the categories and dependency types in the model). These are the category "Documents" and the dependency type "Compare". These are therefore excluded from further discussion regarding this dependency model. In the other cases, the term trace object in the description of the dependency types may be replaced by requirement and we will use this interpretation during the forthcoming discussion.

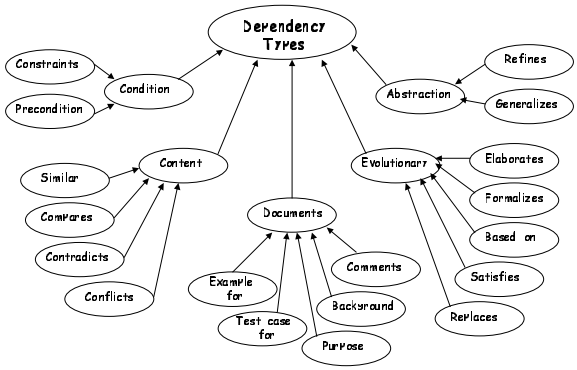

Figure 3: The dependency model [4]

Even though Pohl's model is a valuable starting point for this investigation, the categories and dependency types presented in Pohl's model are sometime difficult to clearly distinguish form each other. There are also additional requirements interdependency types found in subsequent literature.

Pohl also mentions that knowledge about how the requirements have evolved, and hence relate to each other, is considered to be important when dealing with changes and change integration. Kotonya and Sommerville [3] agree with this view and states that requirements interdependencies are one of the most important types of traceability, from a change management perspective. These dependency types are a considerable part of Pohl's model (both abstraction and evolutionary). Pohl also identify requirements interdependencies as an enabler of identifying reusable software component. If similar requirements are detected when the stated requirements are compared with requirements of the existing requirements, this indicates a reusable

component. Similar is one of the dependency type included in the model.

Karlsson et al [5] have developed an approach for requirements selection, through pairwise comparison. They state that requirements prioritisation approaches must include means for managing requirements interdependencies in order to fully support developers. Due to these interdependencies, requirements cannot be treated as stand alone artefacts. For example, if you chose to implement a high priority, low cost requirements, you may have to also implement a low priority, high cost requirement. Requirements can hence not be selected based on priority solely. Karlsson et al [5] concludes that there is a lack of support for requirements interdependencies, where the impact of including or excluding requirements can be observed. They have identified an initial set of interdependency types, which they considered as relevant in the context of requirements selection (see [23]).

Carlshamre and Regnell [8] agree with [5] and conclude that release planning is a very complex task, due to requirements interdependencies. Management of requirements interdependencies are considered as especially important when the requirements are fostered asynchronously in a life cycle model , since it connect the requirements fragments. Future research is requested concerning the different types of interdependencies that exist between requirements. Carlshamre and Regnell [8] describe some types of interdependencies (see [23])

Carlshamre et al [2] have continued the work of [5] and [8], and conducted an industrial survey on requirements interdependencies within release planning. Six different types of interdependencies were defined (see [23]), partially based on the types presented in [5], and analysed in relation to 20 high priority requirements within five different companies. The findings from this survey include that there are few single requirements, i.e. requirements with no relationship to other requirements. It was sometimes fairly difficult for the respondents to choose interdependency type for a relationship between two requirements, because more than one interdependency type could be used. There was hence a need to prioritise the interdependency types. It was also concluded that requirements interdependencies are rarely identified explicitly. There are several reasons for this. The large amount of interdependencies makes them difficult to identify and manage. Requirements interdependencies are also fairly fuzzy, meaning that the relationship they describe can be more or less critical. If R1 increase the implementation cost of R2, it could be a large increase or an insignificant. This problem is also discussed by [6], who states that it is fairly difficult to identify the strength of an interdependency link. Even though pairwise analysis of requirements also supports identification of other problems with the requirements, it requires much time. It is important to find ways of reducing the assessment time and Carlshamre et al discuss some approaches.

Ramesh and Jarke [6] have taken the first steps towards reference models for requirements traceability. They do not focus on requirements interdependencies, but, as we stated above, requirements interdependencies is a traceability problem. Documenting traceability links between requirements in order to model requirements traceability is done even by the companies with a very simple traceability practice. Most of the interdependency types discussed within Ramesh and Jarke [6] are related to requirements management and requirements evolution (see [23]). They state that the decomposition of high level requirements into more detailed requirements, are important to keep track on, e.g. in order to manage the explosion in number of requirements as well as facilitating understanding the requirements by mapping them back to their sources.

Ramesh and Jarke [6] also emphasise that it is neither feasible nor desirable to maintain linkage between all related requirements and output produced during the development process, due to the overheads involved in maintaining traceability links. Instead, it is more feasible to identify the critical requirements and concentrate on storing the relevant traceability information for those.

Robinson et al [7] report on an area called requirements interaction management. This area focuses on managing relationships between requirements, which may interfere with each other s achievements. The idea is to identify requirements that cannot be satisfied simultaneously. Robinson et al, has hence taken an implementation or realisation oriented view on requirements interdependencies. The main aim is to manage conflicts between requirements, and identify the problems with satisfying requirements at requirements definition time. Robinson et al have defined a number of different requirements interdependency types (see [23]).

3.2. Findings from the Interview Survey

This section presents some preliminary results regarding requirements interdependencies from an ongoing interview survey. The survey focuses on current practice and challenges concerning requirements engineering in Swedish software industry, and one part of the survey is focused on requirements interdependencies. For more information about the survey we refer to [24,25].

Generally, most of the respondents acknowledge that requirements do relate to and affect each other. However, managing requirements interdependencies is not considered as an important problem more as an overhead. Not many of the participating companies were documenting requirements interdependencies explicitly. Instead, the requirements were clustered, usually with respect to which requirement that should be implemented together. This could depend on e.g. if they were dealing with the same part of the system, if it would be cost efficient to implement the requirements together at the same time, or if they should be implemented by the same person.

The interdependency types mentioned by the respondents were conflict and cost of implementation. Conflicting requirements affect each other s achievements, and the main work is to make trade offs regarding how to implement the different requirements. Cost of implementation is concerned with identifying requirements that can/should be implemented at the same time, since it decreases the implementation cost. Duplicates and similar requirements were also mentioned.

Requirements interdependencies that are easy to discovere are also considered as easy to manage without documenting them. These interdependencies are handled ad hoc through an experienced and knowledgeable staff. Instead, it is those requirements interdependencies that are difficult to identify as week as difficult to identify they are affecting each other that is the problem. Usually these interdependencies exist between non functional requirements.

4. Requirements Interdependencies – Taking One Step Further

In trying to penetrate the ideas behind the different contributions in the literature it has become clear to us that the perspective that the authors take on the area results in slightly different classifications. In essence, these classifications seem to be influenced by what some stakeholder wants to do with the requirements as part of the development process, e.g. requirements selection or release planning. Also, the various classifications overlap and the meaning of certain terms, which denote dependency types, are not clear in the area as a whole. E.g. the term temporal dependency is given different meanings by different authors. The complete list of interdependency types on which we base the analysis can be found on [23].

A first step forward from the current state could be to classify the interdependencies found in the literature from a more or less neutral perspective and then perhaps go further by adding classifications that are focused on particular areas of interest, e.g. requirements selection or release planning. Taking this stance we have identified two categories of interdependencies that could be considered to be more or less neutral. We tentatively call them *STRUCTURAL* and *COST/VALUE* interdependencies.

4.1. Structural Interdependencies

Structural interdependencies are concerned with the fact that given a specific set of requirements, they can be organised in a structure where relationships are of a hierarchical nature as well as of a cross structure nature. Often high level business requirements are gradually decomposed into more detailed software requirements. Also, requirements from different parts of a hierarchy may influence each other across the overall hierarchy. We find that the following interdependency types fall into this category:

Requires
The fulfilment of one or more higher level requirements depends on the fulfilment of one or more detailed requirements [2, 8, 5, 7].

Explains
A general requirement is explained by a number of more specific requirements [6, 4].

Similar to
One stated requirement is more or less similar to one or more other requirements [4, 7, 2].

Conflicts with
A requirement is in conflict with another requirement if they cannot exist at the same time or if increasing the satisfaction of a requirement decreases the satisfaction of another requirement [7, 4, 5].

4.2. Cost/value Interdependencies

Cost/value interdependencies are concerned with the costs involved in implementing a requirement in relation to the value that the fulfilment of that requirement will provide to the perceived customer/user. The following interdependency types fall into this category:

Increases/Decreases cost of
If one requirement is chosen for implementation then the cost of implementing another requirement increases or decreases [5, 2, 8].

Increases/Decreases value of
If one requirement is chosen for implementation then the value to the customer of another requirement increases or decreases [5, 2, 4]

4.3. A Research Agenda

During the analysis of the current state of research, several issues for future investigation can be identified. Before we can enter deeply in how to manage requirements interdependencies, we first need to identify what types of interdependencies that may exist between requirements. Judging by the discrepancies concerning the requirements interdependency types presented in literature, there are still some miles to go. Our classification is a first step toward developing an overall, neutral model of requirements interdependencies. This classification, however, needs to be further elaborated, and most of all, validated on several different sets of requirements. Since an experienced problem is to choose between different types of interdependency to describe a relationship between two requirements, we find it important to identify a few, fundamental interdependency types. These may later be adjusted to different activities or decisions during the software development process.

A part from developing a reference model on requirements interdependencies, we can identify three major issues for requirements interdependency research:

- *How can we identify requirements interdependencies?* The major problem with requirements interdependencies are not just to record and maintain links between related requirements. These relations must also be identifies somehow. Some interdependencies may be easy to discover when analysing the requirements set, but there are interdependencies, which are difficult to identify. In addition, it can also be difficult to identify how the requirements affect each other, especially regarding non functional requirements. We need to investigate how to identify requirements interdependencies as well as explore how requirements affect each other. Pohl [4] has proposed a method for automatic recording traceability links. Carlshamre et al [2] describe how to use pairwise analysis of the requirements to discover interdependencies, and discuss several alternatives on how to decrease the time required to perform this analysis.
- *How can we describe requirements interdependencies?* When the different relationships between requirements have been identified we must also provide support for storing and managing these interdependencies. A common problem within today s traceability tool is that those provide means to store a relationship between the requirements but not the semantic with such interdependency [6]. There is also a need for mechanisms concerning how to identify the most critical interdependencies between requirements, because it is not feasible to link every related requirements. It must hence be possible to show the strength of the interdependencies [6, 2].

Requirements traceability research include several alternative approaches for recording and managing traceability links. One important research issue is to investigate which of those that are suitable for recording and managing requirements interdependencies. Carlshamre et al [2] presents one approach for describing requirements interdependencies. This approach is built at visualisation, which is considered as an important feature for this issue.

How can we deal with requirements interdependencies? According to Ramesh and Jarke [6] literature and standards within requirements traceability provides few guidelines regarding what type of information that must be captured and used in what context. An important research issue within requirements interdependencies is to investigate what it mean in the different contexts when we state that

there is an interdependency. As indicated by literature, different types of interdependencies are important in different development activities or as basis for various decisions. Another important research issue is to explore which types of interdependencies that are important in the different situations. The first step toward this is to investigate what types of activities that are affected by requirements interdependencies. The following activities are mentioned in the literature.

Requirements Management is concerned with managing the large amount of requirements and information elicited during requirements engineering [9]. Capturing requirements interdependencies may be useful in this phase since it provides an overview of how the high level requirements are decomposed into more detailed requirements [6]. Keeping track of the derived requirements is also a way of managing the fast increasing number of requirements.

Change management. One of the major challenges in software and information systems development is the constant evolution and change of requirements [6]. Requirements interdependencies are shown to be useful in this context since it shows the evolution of requirements, and provide an historical record of the requirements. Requirements interdependencies also allow us to view the major assumptions behind a requirement, by relating it to the originating requirement. However, one of the most important benefits of requirements interdependencies is that they show how requirements relate to and affect each other, and, hence, facilitate impact analysis of change proposals [3, 4].

Release planning is an activity concerned with selecting an optimal collection of requirements for implementation in the next version of a system. The selection is usually based on requirements priority. However, due to the fact that requirements are related to and affect each other, priorities cannot be the only basis [8]. Knowledge about how requirements interact and restrict each other is also an important basis for these decisions.

Reuse of components. If similarity between requirements is documented, these interdependencies can be used to identify reusable components by comparing the stated requirements with the requirements of the existing system [4].

Implementation. Software design is to a large extent concerned with decision making. Many trade offs are made e.g. to decide the scope and functionality of the system as well as between implementation cost and other resources [6]. Requirements interdependencies may support these types of trade offs and decisions, e.g. by revealing interaction between requirements which may interfere with their achievement [7].

Maintenance. Few software and information systems are stable once they are implemented in the organisation. Most systems must continuous evolve due to changes in organisation or users needs, or due to errors made during the development [4]. Requirements interdependencies are useful in this context, since it shows how changing requirements affect other requirements already implemented in the software.

5. Concluding Remarks and Future Work

Requirements interdependencies are essential in supporting several situations and activities within the system development process. However, there is little known about the nature of requirements interdependencies, which is shown by the relatively small amount of literature discussing the phenomenon.

This paper explores the current state of research and compiles the different views of requirements interdependencies found in literature. A first step towards a neutral classification of requirements interdependencies are presented and discussed. Requirements interdependencies can be divided into two main categories, structural and cost/value interdependencies.

A research agenda for requirements interdependencies is outlined. The first step is to further elaborate and validate the classification framework presented in this paper. Other issues is related to identification, documentation and dealing with requirements interdependencies. There are several activities, which may be affected by requirements interdependencies, namely requirements management, change management, release planning, reuse of components, implementation, and maintenance.

References

[1] Regnell, B., Paech, B., Aurum, A., Wohlin, C., Dutoit, A. and Natt och Dag, J. (2001). Requirements Mean Decisions! - Research issues for understanding

and supporting decision making in Requirements Engineering. First Swedish Conference on Software Engineering Research and Practise (SERP 01), October 25 26, Ronneby, Sweden

[2] Carlshamre, P., Sandahl, K., Lindvall, M., Regnell, B. and Natt och Dag, J. (2001) An Industrial Survey of Requirements Interdependencies in Software Product Release Planning. Fifth International Symposium on Requirements Engineering, 27 31 August, Toronto, Canada.

[3] Kotonya and Sommerville (1998) *Requirements Engineering Processes and Techniques*. John Wiley & Sons.

[4] Pohl, K (1996) *Process Centered Requirements Engineering*. John Wiley & Sons Inc.

[5] Karlsson, J., Olsson, S. and Ryan, K. (1997) Improved Practical Support for Large scale Requirements Prioritisation. *Requirements Engineering Journal*, 2(1), p. 51 60

[6] Ramesh, B. and Jarke, M. (2001) Toward Reference Models for Requirements Traceability. *IEEE Transactions on Software Engineering*, Vol.27, no1., p. 58 93

[7] Robinson, W.N., Pawlowski, S.D. and Volkov, V. (1999) Requirements Interaction Management. GSU CIS Working Paper 99 7, Department of Computer Information Systems, Georgia State of University, Atlanta.
(Printed 04l201 from http://cis.gsu.edu/ wrobinso)

[8] Carlshamre, P. and Regnell, B. (2000) Requirements Lifecycle Management and Release Planning in Market Driven Requirements Engineering Processes. Second International Workshop on the Requirements Engineering Process. Grenwich, London.

[9] Grehag, Å. (2001) Requirements Management in a Life Cycle Perspective A Position Paper . In Ben Achour Salinesi, C., Opdahl, A.L., Pohl, K. and Rossi, M. (Eds) Proceedings of the Seventh International Workshop on Requirements Engineering: Foundation for Software Quality, REFSQ 01, Interlaken, Switzerland. Essenere Informatik Beiträge, pp. 183 188.

[10] Gotel, O. (1995) *Contribution Structures for Requirements Traceability*. PhD Thesis, Department of Computing Imperial Collage of Science, Technology and Medicine, University of London.

[11] Maciaszek, L.A. (2001) *Requirements Analysis and System Design Developing Information Systems with UML*. Addison Wesley.

[12] Jarke, M., Rolland, C., Sutcliffe, A. And Dömges, R. (1999) *The NATURE of Requirements Engineering*, Shaker Verlag, Aachen.

[13] Gotel, O. and Finkelstein, A. (1994) An Analysis of the Requirements Traceability Problem. In *Proc. of the 1st international Conference on Requirements Engineering*, Colorado Springs, Colorado, USA. p. 94 102

[14] Ramesh, B. (1993) A Model of Requirements Traceability for Systems Development. Technical report. Naval Postgraduate School. Monterey, CA. USA. September.

[15] Ramesh, B., Powers, T., Stubbs, C. and Edwards, M. (1995) Implementing Requirements Traceability: A Case Study. In *Proc. of the 2nd International Symposium on Requirements Engineering*, York, England, p. 89 93.

[16] Kaindl, H. (1993) The Missing Link in Requirements Engineering, *ACM SIGSOFT Software Engineering Notes*, 18(2), pp. 30 39.

[17] Gotel, O. and Finkelstein, A. (1997) Extended Requirements Traceability: Results of an Industrial Case Study. In *Proc. 3rd International Symposium on Requirements Engineering* (RE97), IEEE Computer Society Press, p. 169 178.

[18] IEEE 830 (1994) *Guide to Software Requirements Specification*. ANSI/IEEE Std. 830, Institute of Electrical and Electronics Engineers, New York

[19] Johnson, W.L., Feather, M.S. and Harris, D.R. (1991) Integrating Domain Knowledge, Requirements, and Specifications, *Journal of Systems Integration*, 1, p. 283 320.

[20] Robertson, S. and Robertson, J. (1999) *Mastering the Requirements Process*, Addison Wesley

[21] Jarke, M. (1998) Requirements Tracing, *Communication of the ACM*, December 41(12).

[22] Davis, A.M. (1990) The Analysis and Specification of Systems and Software Requirements, In *Systems And Software Requirements Engineering*, IEEE Computer Society Press, p. 119 144

[23] Dahlstedt, Å. and Persson, A. (2003) An Overview of Requirements Interdependency Types . http://www.ida.his.se/ida/ asa/ReqInterdependencies.pdf

[24] Karlsson, L., Dahlstedt, Å., Natt och Dag, J., Regnell, B. and Persson, A. (2002) Challenges in Market Driven Requirements Engineering an Industrial Interview Study, Eighth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ), September, Essen Germany.

[25] Dahlstedt, Å. G., Karlsson, L., Persson, A., Natt och Dag, J. and Regnell, B. (2003) Market Driven Requirements Engineering Processes for Software Products a Report on Current Practices. Submitted to Development of Product Software, DoPS 03, 20 and 21 June 2003, Velden, Austria.