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.

Requirements relate to and affect each other, i.e. they are interdependent. This paper provides an overview of the current state of research on requirements interdependencies and formulates a research agenda for the area. The research agenda, which is based on a new classification drawn from the literature and intermediary results from an ongoing interview study, addresses a number of unresolved issues concerning the identification, documentation and use of requirements interdependencies in the software development process.

1. Introduction

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. Dependencies between requirements may also affect various decisions and activities during development, e.g. requirements change management [3, 4], release planning [2, 5], requirements management [6], requirements reuse [7] and requirements implementation [8]. This implies that there is a need to take interdependencies into consideration in order to make sound decisions during the development process (for examples, see Section 3.1). Despite this, little is known about the nature of requirements interdependencies, and further research is needed in order to understand the phenomenon better [5, 9,10].

The overall aim of our research is to identify which types of requirements interdependencies that are critical to take into consideration in specific development situations, such as e.g. release planning or requirements management. Also, we aim to propose approaches for managing dependencies according to the needs in each specific situation. This paper provides a first step towards this research goal, by providing an overview of the current state of requirements interdependency research, by developing an integrated classification of fundamental interdependency types discussed in the literature, and formulating a research agenda for further research.

The amount of literature addressing requirements interdependencies is fairly small and it approaches the area from different perspectives. Pohl [4] as well as Ramesh and Jarke [6] discuss the topic 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 [9] and Carlshamre et al [2]. Robinson et al [8] reports on requirements interaction management, which deals with identifying how requirements may affect each other's achievement.

The aim of this paper is, as stated above, to provide an overview of the current state of requirements interdependency research. Our first step was to explore which types of interdependencies that are currently known. This was done by compiling the different views found in the literature, by identifying common patterns among described types to discover fundamental ones. The result is a classification of known interdependency types presented in Section 4. The classification is neutral with respect to development situations. It needs to be further elaborated with respect to the specific needs within the different development situations where requirements interdependencies affect the work. We have also formulated a research agenda, where fundamental problems when dealing with interdependencies have been identified as well as identified some initial development situations where it is considered important to take requirements interdependencies into account.

The paper is organised as follows. Section 2 places requirements interdependency into its context – requirements traceability. We also provide a brief overview of the area an a discussion about the term interdependency. An overview of the literature found addressing requirements interdependencies are provided in Section 3 together with some preliminary findings from an ongoing interview survey. This is then compiled into a neutral classification of fundamental interdependency types presented in Section 4. This section also includes the research agenda developed for requirements interdependency research. The paper ends with some concluding remarks in Section 5.

2. Traceability: a Basis for Understanding Requirements Interdependencies

Requirements traceability has been acknowledged as an important part of software and information systems development [4, 11, 12] supporting various activities during the life of a software system. We view the area as a basis for addressing requirements interdependencies. The topic is well-explored, judging by the large amount of literature describing both theoretical and empirical studies (see e.g. 4, 13, 11, 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.

There are several definitions of the term traceability [see e.g. 6, 18, 19, 4]. 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*" [20, pp. 32, based on 14]. 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 [14] 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 [14] and is focused on enabling a better understanding and acceptance of the current system/software.

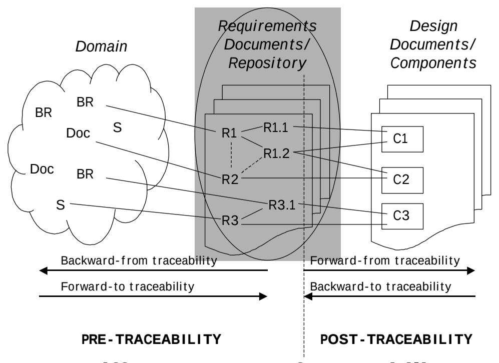

Figure 1: Different types of traceability

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 [21]. According to [6] traceability information provides important support within requirements engineering, design, systems evolution, and test procedures.

The various types of traceability links presented in Figure 1 support different situations and activities during the development and maintenance of the software 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].

Traceability is concerned with tracing relationships between trace objects of various types, e.g. requirements, rational, document, process stages etc. In this paper, we focus on relationships between a specific type of trace object – namely explicitly stated requirements (showed by the shaded area in Figure 1). 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 2). 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 manners 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 trace objects of the same type.

3. Requirements Interdependencies – Current State of Research

This section aims at providing an overview of the current state of research on requirements interdependencies by outlining findings from the literature concerning requirements interdependency types and affected development situations as well as findings from an ongoing interview survey. The complete set of requirements interdependency types found in the literature are presented in [22]. These are discussed and compiled into a neutral classification of fundamental requirements interdependencies presented in Section 4. We have delimited our survey to literature explicitly discussing interdependencies between requirements.

3.1. Requirements Interdependencies – a Literature Review

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

In 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 (Figure 2). Pohl's model describes dependency types that can exist between any type of trace object used in the requirements engineering process. We focus on 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.

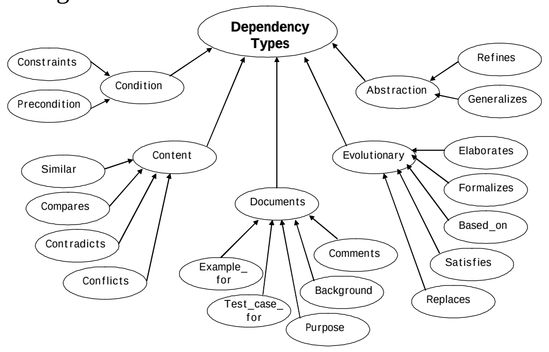

Figure 2: The dependency model [4]

However, Pohl's dependency model must be somewhat adapted and specialised towards requirements interdependencies to be useful in our research. There are some dependency types included in Pohl's model that clearly cannot exist between requirements (see [22] for a description of the categories and dependency types in the model). These are the category "Documents" and the dependency type "Compare", which 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 in the forthcoming discussion.

Even though Pohl's model is a valuable starting point for our research, the categories and dependency types presented in Pohl's model are sometimes difficult to clearly distinguish from each other. There are also additional requirements interdependency types found in subsequent literature. There is hence a need to adapt and revise this model in order to develop a model focusing specifically on requirements interdependencies and also to incorporate recent research.

Pohl 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 the notion of requirements interdependency is one of the most important aspects of traceability, from a change management perspective. These dependency types are a considerable part of Pohl's model (both abstraction and evolutionary). Pohl also identifies requirements interdependencies as an enabler of identifying reusable software components. If similar requirements are detected when the stated requirements are compared with existing requirements, this indicates a reusable component. The dependency type "Similar" is included in the model.

Karlsson et al [5] have developed an approach for requirements selection, through pair-wise 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 choose to implement a high priority, low cost requirement, you may also have to implement a low priority, high cost requirement. Requirements can hence not be selected based solely on priority. Karlsson et al [5] concludes that there is a lack of support for requirements interdependencies, one particular 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 [22]).

Carlshamre and Regnell 98] agree with [5] and conclude that release planning is a very complex task, due to requirements interdependencies. Management of requirements interdependencies are considered to be especially important when the requirements are "fostered asynchronously in a life cycle model", since they connect the requirements fragments. Future research is claimed to be needed concerning the different types of interdependencies that exist between requirements. Carlshamre and Regnell [9] describe some types of interdependencies (see [22]).

Carlshamre et al [2] have continued the work of [5] and [9], and conducted an industrial survey on requirements interdependencies within release planning. Six different types of interdependencies were identified (see [22]), 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 are that there are few single requirements, i.e. requirements with no relationship to other requirements. It was sometimes fairly difficult for the respondents in the study 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 results in difficulties to identify and manage dependencies. Requirements interdependencies are also fairly fuzzy, meaning that the relationship they describe can be more or less critical. If R1 increases 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 pair-wise 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 to this end.

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. According to [6] companies with a simplistic traceability practice also document traceability links between requirements in order to model requirements traceability. Most of the interdependency types discussed are related to requirements management and requirements evolution (see [22]). Ramesha and Jarke [6] also state that the decomposition of high level requirements into more detailed requirements, is important to keep track on, e.g. in order to manage the explosion in the number of requirements as well as facilitating understanding of the requirements by mapping them back to their sources.

Ramesh and Jarke [6] also emphasise that it is neither feasible nor desirable to maintain links 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 to concentrate on storing the relevant traceability information for those.

Robinson et al [8] 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 also defined a number of different requirements interdependency types (see [22]).

An approach for systematic recycling of requirements between requirements documents referring to product variants is presented by von Knethen [4], who also considers it important to ensure that all related requirements to a copied requirement are transferred to the recycled document. There are some interdependency types presented and used within this approach (see [22]).

We can hence conclude that several different types of interdependencies are presented in the literature and that different activities or development situations are in focus (see Section 4).

3.2. Some findings from an Ongoing Interview Study

This section presents some preliminary results regarding requirements interdependencies from an ongoing interview study. The study focuses on current practice and challenges concerning requirements engineering in Swedish software industry, and one part of the study is more specifically focused on requirements interdependencies. For more information about the study, see [23, 24].

Generally, most of the respondents in the study acknowledge that requirements do relate to and affect each other. However, not many of the participating companies documented requirements interdependencies explicitly. Instead, the requirements were clustered, usually with respect to which requirement that should be implemented together. This could e.g. depend on whether the requirements concerned the same part of the system, if it would be cost efficient to implement the requirements at the same time, or if they should be implemented by the same person.

The interdependency types mentioned by the respondents were mainly 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 this decreases the implementation cost. Duplicates and similar requirements were also mentioned.

Requirements interdependencies that are easy to discover are also considered easy to manage without documenting them. These interdependencies are handled ad-hoc through experienced and knowledgeable personnel. Instead, it is those requirements interdependencies that are difficult to identify that are problematic to deal with. Also, it is sometimes possible to identify that there is an interdependency, but the consequences of the dependency is difficult to comprehend. Usually these interdependencies exist between non-functional requirements.

4. Towards a Model of Fundamental Interdependency Types

Before we can enter deeply into addressing how to manage requirements interdependencies in different situations, we first need to compile the different views expressed in the literature into an integrated model, which is neutral with regard to development situation. One identified problem is to choose between different types of interdependencies and Pohl's dependency model alone comprises 18 types. Also, judging by the discrepancies between requirements interdependency types presented in the literature, there is still some work to be done.

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 the 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 our analysis can be found in [22].

Based on the literature and also on some intermediary results from an ongoing interview study, we have developed a classification (Figure 4), which could be considered to be a first step towards developing an overall, neutral model of fundamental requirements interdependencies.

This classification will most likely need to be further elaborated and most of all validated, e.g. using a number of different sets of requirements. Since we have focused on identifying a few types that we so far consider to be fundamental, these may later be adjusted or extended to suit different needs in the software development process, e.g. in requirements selection or release planning.

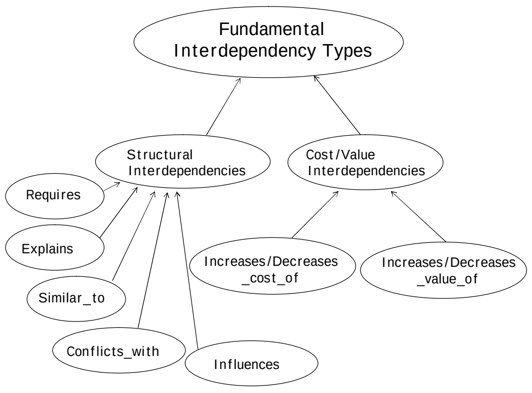
Fundamental Interdependency Types
Structural Interdependencies
Cost/Value Interdependencies
Requires
Explains
Similar_to
Conflicts_with
Influences
Increases/Decreases _cost_of
Increases/Decreases _value_of

Figure 4: The new classification

Taking this stance we have identified two categories of interdependencies that could be considered to be fundamental and 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 requirement depends on the fulfilment of another requirement. This type can be used to describe a hierarchical relation between two requirements, but also relations across hierarchical structures.

This dependency type is derived from the interdependency types "requires" [2], "and" [2], "logical" [9] and "must-exist"[5]. This relationship can also be viewed in the opposite direction i.e. instead of R1 requires R2, R2 is a prerequisite for R1 [2]. The interdependency type *Requires* then also covers "precondition" mentioned by Pohl [4]. Carlshamre et al [2] concludes in their investigation that the temporal dependency type [9, 2, 8] is seldom interesting. It may either be viewed as a *Requires* dependency or an *Increase/decrease_cost_of* dependency (see 4.2). We have chosen to agree with this view, since a temporal interdependency also is useful from the perspective of which activity that should be performed and is hence not neutral.

The "or" dependency [2] is difficult to categorise, since it can be related to different interdependency types. The "or" dependency relates alternative solutions to each other, which e.g. may be required by another requirement i.e. R1 requires some of the following requirements {R2, R3, or R4}. Clearly, this dependency type requires more research in order to be fully understood.

"Satisfy" [4] and "positive correlation" [8] can be viewed as a weaker dependency of the type require. They both concern linking requirements, which support the fulfilment of another requirement. In this context, the require dependency type is used when R1 *must* be implemented in order to fulfil R2, while "satisfy" and "positive correlation" is weaker and describes a situation where the fulfilment of R1 have a positive effect on the fulfilment of R2.

Explains

A general requirement is explained by a number of more specific requirements. This dependency type is used to describe hierarchical structures of a weaker nature than *Requires* and relates more detailed requirements to their source requirements. If a detailed requirement is derived from a high level requirements, but it is not a prerequisite for this requirement, the relation is of the dependency type explain.

This dependency type covers "elaborate", "part_of", "is_a" and "derive" from Ramesh and Jarke [6], and "elaborate", "formalise", "replaces", "generalises", "refines" and "based_on" by Pohl [4] as well as "refinement" from von Knethen et al [7]. As stated above, we seek to identify basic interdependency types, and these are fairly similar and may be difficult to distinguish. We have therefore chosen to summarise them into one overall dependency type.

Similar_to

One stated requirement is more or less similar to one or more other requirements.

This interdependency type corresponds with "similar" [4] and "structure" [8]. This interdependency type is also mentioned within the interview study. Natt och Dag [25] presents an evaluation of the feasibility to use natural language processing techniques to identify duplicates within a requirements set.

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.

This interdependency type includes both situations were it is impossible to implement both requirements, and situations were these have a negative influence on each other's achievements and a trade-off between the resolution of the requirements must be made. It hence covers "constraint" [4], "negative correlation" [8], "conflict" [4] and "cannot_exist" [5]. Conflict is also one of the most frequently mentioned interdependency types in the interview survey. Robinson et al [8] has a strong focus on conflict dependencies, and present some relations, which can be interpreted as reasons for the conflict e.g. "resource", "tasks" and "causality".

Influences

A requirement has an influence on another requirement.

It is indicated in the literature that a requirement may affect or influence another requirement in other ways than requires, explains and conflicts. Both [6] and [7] has a fairly general interdependency type, termed "depend_on" and "dependency". Our hypothesis is that more dependencies can be identified, especially when this classification is further elaborated with respect to different development activities or situations. However, at this stage we choose to include a general interdependency type which can be used if a relationship between two dependent requirements is not of the type "requires", "explains" or "conflicts_with".

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.

This interdependency type includes "icost" [2], "positive cost" and "negative cost" [5] as well as "value-related" [9].

Increases/Decreases_value_of

If one requirement is chosen for implementation then the value to the customer of another requirement increases or decreases.

This interdependency type covers "cvalue" [2] as well as "positive value" and "negative value" [5].

4.3. A Research Agenda

Apart from developing a reference model of fundamental requirements interdependencies and extending this to cater for specific needs in the software development process, we can identify three major issues for research in the area of requirements interdependencies:

- *How can we identify requirements interdependencies?* The problems within requirements interdependencies are not only concerned with how to record and maintain links between related requirements. These relationships must also be identified somehow. Some interdependencies may be easy to discover when analysing the requirements set, but there are interdependencies, which are more 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 to explore how requirements affect each other. Pohl [4] has proposed a method for automatically recording traceability links. Carlshamre et al [2] describe how to use pair-wise analysis of the requirements to discover interdependencies, and they also discuss several alternatives regarding how to decrease the time required performing this analysis. Both these approaches assume that the developers know how the requirements affect each other. There is, however, a need for approaches focusing on how the explore the consequences of an interdependency, i.e. how the requirements affect each other.

- *How can we describe requirements interdependencies?* When the different relationships between requirements have been identified we must also provide support for storing and managing them. A common problem in current traceability tools is that they provide means to store a relationship between requirements but they provide very little guidance regarding the semantic and inherent meaning of the relationship [6]. There is also a need for mechanisms identifying the most critical interdependencies, because it is not feasible to link every related requirement. It must hence be possible to show the strength of the interdependencies [6, 2].

Requirements traceability research includes several alternative approaches for recording and managing traceability links. One important research issue is to investigate which of those are suitable for recording and managing requirements interdependencies. Also, Carlshamre et al [2] presents one approach for describing requirements interdependencies. This approach is built on visualisation, which is considered as an important feature for this issue. It could also be relevant in this context to look at different techniques for goal modelling (see e.g. [26]) as a means to model and describe interdependencies, since requirements could be considered to be low-level goals. The F^3 Enterprise Modelling language [27], more specifically a sub-model denoted the Information Systems Requirements Model, also includes means of describing requirements interdependencies, based on this notion.

- *How do we use requirements interdependencies in the software development process?* According to Ramesh and Jarke [6], literature and standards within requirements traceability provide few guidelines regarding what type of information that must be captured and used in what context. An important research issue is, therefore, to investigate what it means in different contexts when we state that there is an interdependency. As indicated by the 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 are critical to consider in different situations. The first step towards this is to investigate what types of activities are affected by requirements interdependencies. As a starting point the following activities, mentioned in the literature, can be used.

Requirements Management is concerned with managing the large amount of requirements and information elicited during requirements engineering [10]. 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 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. 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, which, hence, facilitates 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 [9]. Knowledge about how requirements interact and restrict each other is, therefore, 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].

Reuse of requirements. When requirements are reused in requirements documents describing various variants of a product, it is considered as relevant to ensure that all requirements related to a copied requirement is transferred to the recycled document [7].

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 [8].

Testing. A potentially interesting area where requirements interdependencies may be a relevant aspect to take into consideration is software testing. During this activity, test cases are developed based on the requirements which fulfilment is supposed to be ensured. Since requirements relate to and affect each other, their knowledge about requirements interdependencies may affect the ability to create purposeful and complete test cases.

Maintenance. Few software and information systems are stable once they are implemented in the organisation. Most systems continuously 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

Keeping track of requirements interdependencies is essential in order to support 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 compiles the different views of requirements interdependencies found in the literature. It also takes a first step towards what we call a neutral classification of fundamental requirements interdependencies. In this classification, interdependencies are grouped in two main categories; structural and cost/value interdependencies.

A research agenda for requirements interdependencies has also been outlined. The first step is to further elaborate and validate the classification framework presented in this paper in relation to development activities or situations affected by requirements interdependencies. Other research issues are related to the identification, documentation and use of requirements interdependencies.

We have mainly addressed the area of requirements interdependencies from a theoretical point of view in this paper. The main concern for the future, however, is to focus on empirical research that gives a useful contribution to solving pressing problems in the field of software development practice.

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 Engingeering – 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] von Knethen, A., Peach, B., Kiedaisch, F. and Houdek, F. (2002) Systematic Requirements Recycling through Abstraction and Traceability. Proc. of IEEE Joint International Conference on Requirements Engineering, 9-13 Septermber, Essen, Germanny, pp. 273-281.

[8] 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 041201 from http://cis.gsu.edu/~wrobinso)

[9] 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.

[10] 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.

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

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

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

[14] 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

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

[16] 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.

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

[21] 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

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

[23] 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.

[24] 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.

[25] Natt och Dag, J., Regnell, B., Carlshamre, P., Andersson, M. and Karlsson, J. (2002) "A feasibility study of automated natural language requirements analysis in market-driven development. " *Requirements Engineering, 7*, pp. 20-33.

[26] Bubenko J.A. jr, Persson A., Stirna J. (2001) *User Guide of the Knowledge Management Approach Using Enterprise Knowledge Patterns*, deliverable D3, IST Programme project HyperKnowledge -- Hypermedia and Pattern Based Knowledge Management for Smart Organisations, project no. IST-2000-28401, Dept. of Computer and Systems Sciences, Royal Institute of Technology, Stockholm, Sweden,
available on http://www.dsv.su.se/~js/ekd_user_guide.html

[27] Bubenko jr., J. A., *"Extending the Scope of Information Modelling"*, Fourth International Workshop on the Deductive Approach to Information Systems and Databases, Lloret, Costa Brava (Catalonia), Sept. 20-22, 1993. Department de Llenguatges i Sistemes Informatics, Universitat Politecnica de Catalunya, Report de Recerca LSI/93-25, Barcelona.