CREWS Report 99-16

Improving Reviews of Conceptual Models by Extended Traceability to Captured System Usage

by

Peter Haumer*, Matthias Jarke*, Klaus Pohl**, Klaus Weidenhaupt*

**) Information Systems (I5), RWTH Aachen*
Ahornstraße 55, 52056 Aachen, Germany
Tel.: +49 (0)241 / 80 21 501
Fax: +49 (0)241 / 8888 321
{haumer,jarke,weidenhaupt}@informatik.rwth-aachen.de

***) Software Systems Engineering, University of Essen*
Altendorferstr. 57-101, 45117 Essen, Germany
Tel.: +49 (0)201 / 81003 - 00
pohl@informatik.uni-essen.de

Improving Reviews of Conceptual Models by Extended Traceability to Captured System Usage

Abstract

When specifying change for an existing system, the history and functionality of the system to be replaced has to be considered. This avoids neglecting important system functionality and repeating errors. The properties and the rationale behind the existing system can be elicited by analysing concrete system-usage scenarios. The results of the analysis of the existing system are then typically represented using conceptual models. To establish conceptual models of high quality reviewing the models is common practice. The problem faced with when reviewing conceptual models, is that the reviewer cannot assess and therefore understand the basis (concrete system usage) on which the conceptual models were built.

In this paper, we present an approach to overcome this problem. We establish Extended Traceability, by recording concrete system-usage scenarios using rich media (e.g. video, speech, graphic) and interrelating the recorded observations with the conceptual models. We discuss the main improvements for review processes and illustrate the advantages with excerpts from a case study performed in a mechanical engineering company.

Keywords: scenario-based requirements engineering, requirements management, requirements traceability, formal reviews, goal modelling, rich media, CASE environments

1. Introduction

The role of Requirements Engineering (RE) is to establish a complete, consistent and unambiguous requirements specification of the desired system which defines the requirements at an abstract conceptual level. When specifying the requirements for the new system, the history and functionality of an existing system to be replaced has to be considered (cf. [McMenamin and Palmer, 1984] or [Gause and Weinberg, 1989]). An important reason for this is that in many cases, a system is not designed from scratch, but the vision for the new system is developed by the wish to change an existing system. Therefore, the new system has to provide to a large degree the functionality of old systems and has to obey its context of constraints. Another notable reason is that the analysis team can learn a lot from the success stories and pitfalls of the design and usage of the existing system. Considering the history and functionality of the existing system avoids repeating failures or forgetting important functionality. Consequently, many RE approaches (e.g., [McMenamin and Palmer, 1984]; [Jackson, 1995]) consider two categories of conceptual models:

– *Current-state* models, which express essential properties and functionalities of critical aspects of the **existing** system;
– *Desired-state* models, which defines the requirements for the **future** system.

Requirements Engineering is also a cooperative learning process. Stakeholders and requirements analysts have to communicate with each other for eliciting and understanding requirements, as well as for detecting gaps and inconsistencies while validating requirements. The quality of the conceptual models created during RE heavily depends on successful stakeholder involvement. This applies especially to the quality of current-state models, because only the stakeholders dealing with the existing system in use (e.g. domain experts, system users, maintenance people etc.) have the fundamental knowledge about specific properties, behaviours, problems etc.

To ensure correctness and appropriateness of current-state models, they must be validated. Reviews are a common validation technique widely used in industrial practice [Freedman and Weinberg, 1982]. In a review typically one or more stakeholders check the validity of a model.

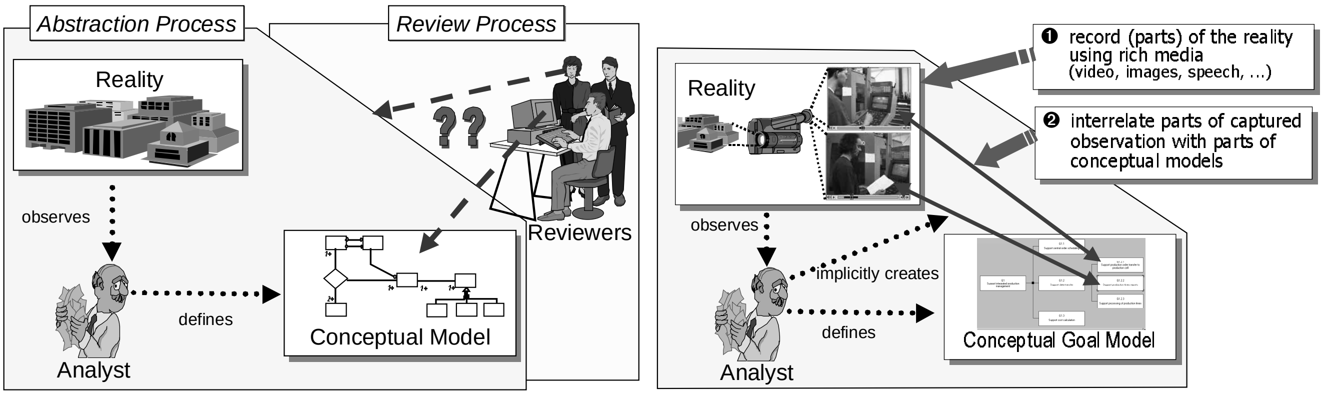

Figure 1(left). Problem: Abstractions are not understandable
Figure 1(right). Solution: Extended Traceability by interrelating parts of observations with parts of models

However, reviewing is burdened with a major problem: Conceptual models are hard to understand for somebody not involved in the model definition. Generally, the abstractions made by the analyst who created the model are not traceable for others, i.e. the reviewer cannot assess and therefore understand the basis on which an abstraction was made. This is illustrated in Figure 1 (left). The analyst depicted at the lower left corner has produced the conceptual model on the lower right based on his observations and his perception of current reality. The abstraction process has taken place in his mind and is by no means traceable. Thus, the group of people depicted in the upper right corner of the figure can hardly assess the quality and correctness of the conceptual model during a review for several reasons:

– It is important that stakeholders which have a fundamental knowledge about the modelled domain like users, customers, domain experts etc., but who are normally not directly involved into analysis processes, participate in the review of conceptual current-state models. They might provide invaluable insights and can decide if system aspects have been understood by the analyst in a correct way. However, these persons are normally not trained in the modelling language and therefore, a different way of communication concerning the model has to be found.

– Even when reviewers are familiar with the modelling technique, they might not understand what a concept is about. For instance, a concept could be too abstract to unambiguously understand it (*conceptual distance*), the reviewer might not be familiar with the concept's domain, or he does not know the aspects modelled.
– Because most conceptual models are semi-formal, reviewers might have different interpretations of the concept's semantics. Abstracting from real world situations different analysts would create different models based on their personal perception and interpretation of the observed situations and modelling capabilities. Moreover, a stakeholder could interpret an abstract concept in a different way based on his personal knowledge and experiences. Therefore, different perceptions and opinions on current system reality might lead to misunderstandings and distorted review results.

Consequently, reviewers need not only more information about the products, but also more information about the analyst's abstraction process and the rationale which led to the concepts' creation. This tracing back to the origins of the abstractions has been defined by Gotel and Finkelstein in [Gotel and Finkelstein, 1994] as *pre-specification traceability* (in short: *pre-traceability*; in contrast to *post-traceability* which traces the realisation of requirements, e.g. in design and implementation; cf. [Pohl, 1996b] for an overview on requirements traceability). Without pre-specification traceability current-state models are hard to understand and justify.

To overcome the conceptual distance and interpretations problems, the use of rich media to record and discuss current-system usage is described in several publications (e.g., video-supported Participatory Design techniques [Brun-Cottan and Wall, 1995]; [Suchman and Trigg, 1991]; the Scenario-Based Engineering Process (SEP) [McGraw and Harbison, 1997]; Contextual Inquiry [Beyer and Holtzblatt, 1993]; ethnography approaches [Blomberg, 1993]; [Hughes et al., 1995]; and workplace culture approaches [Bødker and Pedersen, 1991]). All these approaches support the performance of site visits and the recording of current system usage; e.g., how to prepare and handle observed personnel, which observations material to use for what purpose, etc. As several applications have shown, the use of rich media to record scenarios about current-system usage, and the use of the recorded system usage to discuss and elicit requirements with diverse stakeholders, improve the quality of the resulting specifications.

These techniques primarily have been developed to include stakeholders like users and domain experts into analysis processes and improve the creation process of abstractions based on improved communication about the problem domain. The use of persistent rich media and its relation to conceptual current-state models is most beneficial when the observation of a certain task or system aspect is expensive or hard to repeat. Moreover, rich media tend to be closer to reality, in particular when the observations are hard to describe. Although, even video-taped scenes may reflect the bias of the analyst performing the video-taping and editing, they are still more objective than relying on written minutes or even the personal memory of the analyst. However, they do not provide a tight integration with existing and common RE methods and artefacts by linking their rich media sources, i.e. the origins of the abstractions made, with the resulting model components. Because of this missing link, abstractions based on usage situations captured with rich media, are still **not** traceable.

In this paper, we present an approach for interrelating captured observations and conceptual models which overcomes this shortcoming and thereby improves the review processes. The underlying idea for a solution is depicted in Figure 1 (right), which illustrates what we call *Extended Traceability*. We assume that, as in the techniques mentioned above, current system usage is recorded using rich media (step 1 of Figure 1 (right)). Whenever a captured observation is used in the abstraction process (i.e. either for the elicitation or validation of concepts) of the conceptual models, we support the persistent interrelation of the used recorded observation with the concepts (step 2 of Figure 1 (right)). More precisely, the parts of a conceptual model is related to exactly those parts of the recordings, that have influenced its definition. The interrelations provide traceability of concepts through direct access of relevant parts of real world observations to make abstractions of these observations more transparent, even on a very fine-grained level.

The established interrelations result in a special form of pre-traceability that we call *Extended Traceability*. This type of pre-traceability extends existing traceability approaches (e.g., [Pinheiro and Goguen, 1996]; [Ramesh et al., 1997]; [Watkins and Neal, 1994]; [Weiser and Morrison, 1998]) in the way that a) it provides traceability back to concrete instance examples from the real world instead of just tracing between different representations of abstractions and b) it allows to establish traceability in a fine-grained manner by interrelating arbitrary parts of conceptual models with arbitrary parts of real world observations and not just interrelationships on a document level.

The application of our technique focuses on observable aspects of current-system usage, i.e. interaction of human users with the system. Moreover, it is important that the system aspects and usage under consideration can actually be captured by rich media. Thus, our approach would probably be less suitable for, e.g., embedded systems, or when the current system covers the functionality of the future system only to a very small degree.

In [Haumer et al., 1998] we already provided a broad overview of our technique and the tool environment and described several loosely coupled applications of the approach as well as the lessons learned during a trial application we performed. In this paper, we go into more detail about a more coherent and more specific application of the approach: the support of reviews.

The remainder of the paper is structured as follows. In Section 2 we describe how to establish Extended Traceability between real world scenes and conceptual goal models. In Section 3, we outline the principle types of support provided for the review processes and illustrate their advantages by an embedded example drawn from a case study. In Section 4, we summarise our results and provide an outlook on future work.

2. Principles of Extended Traceability

2.1 Establishing Extended Traceability while Using it

Figure 2 depicts our overall approach for establishing Extended Traceability (see [Haumer et al., 1998] for details). Current-system usage is recorded on-site using rich media (Figure 2, step 1) in what we call *Real World Scenes (RWS)*. The material gathered during an observation may contain information about many system usages. We therefore edit and pre-structure this material into what we call a *Real World Example (RWE)* (Figure 2, step 2). An RWE is a collection of materials that represents *one* coherent and complete observed system usage. The material belonging to an RWE should be arranged in a suitable manner, for example, if the observation was recorded using video, the video should be edited in a way that it shows the temporal sequence of a sample system usage.

Note, that we did not develop any support on how to perform site visits and record RWEs, because this was not the focus of our approach. Therefore, for our own case studies we reused existing guidelines for capturing (e.g. [Brun-Cottan and Wall, 1995]; [McGraw and Harbison, 1997]) and pre-structuring real world scenes into real world examples.

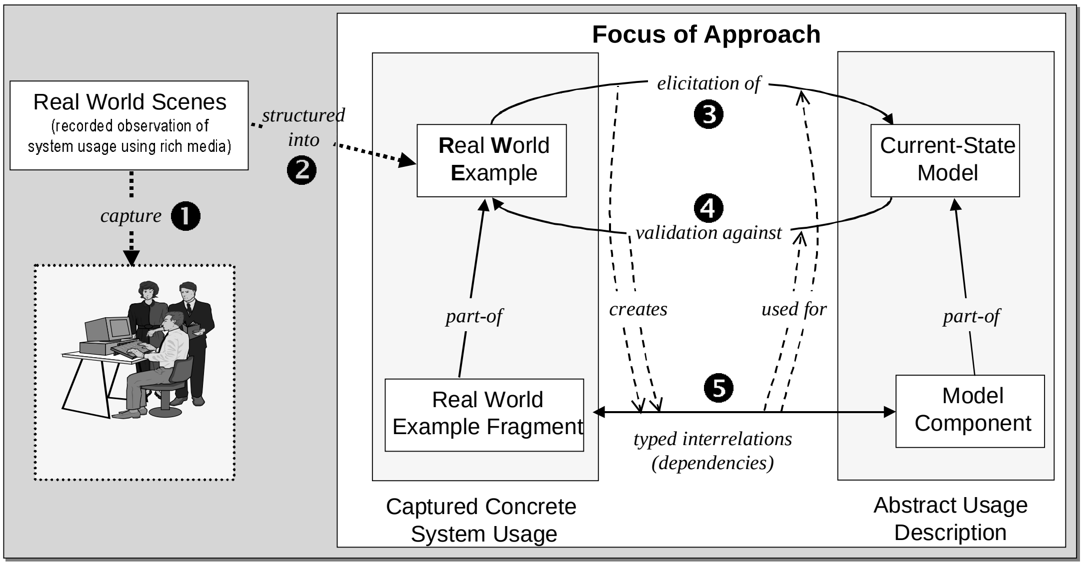

Figure 2. Schematic overview of Extended Traceability.

RWEs are used for two main purposes. On the one hand, new concepts can be elicited from RWEs (Figure 2, step 3). On the other hand, current-state model concepts which either have been elicited from RWEs or have been elicited from other sources can be validated against various RWEs (Figure 2, step 4), i.e. several other but similar usage situation. In both cases, we interrelate corresponding fragments of the RWEs (*Real World Example Fragments (RWEF)*) with the concepts of the current-state model (Figure 2, step 5), which have been elicited or validated against these scenes. In other words, fine-grained interrelations and therefore *Extended Traceability* means that we relate parts of the media (e.g. cut-out video clips, one picture as extreme) to parts of the conceptual model and not whole hours of video to a large conceptual model. This enables fast and selective access to the relevant parts of the video and avoids time-consuming watching of irrelevant information.

The advantage of this technique is that the intertwined process of elicitation and validation, i.e. creation and usage of interrelations, gradually creates further interrelation structures. Thus, initially unstructured scenes become more and more structured while using them and conceptual models become gradually documented in respect to their relation to real world observations during the analysis process. As an important side effect traceability of scenarios (i.e. scenes) usage is established as well. Consequently, the established interrelation structure imposes on the one hand access paths upon the real world examples; on the other hand, the conceptual model is annotated by a set of real world evidences that are close to the perception of the involved stakeholders.

For persistent recording of the extended trace information we adapted the trace repository developed for PRIME[1] (see [Pohl, 1996a] and [Pohl, 1996b] for details). Our extension called CREWS-PRIME manages all products and traceability information in a (logically) central repository. Features of the tools built on top of this repository will be described throughout Section 3. A detailed description of the process-integrated tool environment CREWS-PRIME can also be found in [Haumer et al., 1998].

2.2 Extended Traceability of Conceptual Goal Models Abstracting from Current-System Usage

Note, that we do not propose a new modelling technique, but argue to embed an appropriate existing technique in our overall approach.

However, in early phases of analysing the existing system, it is important to understand and agree about the *why* behind certain properties of the system (i.e., “Why does the system support this activity?”), before dealing with details about *what* and *how* [Anton, 1996]; [Dardenne et al., 1993]; [Yu, 1994]; [Yu, 97], (e.g., the data the system deals with) system function and/or system behaviour. Consequently, more and more RE frameworks

---

[1] PRIME: PRocess Integrated Modelling Environment.

suggest the explicit definition of goal models prior to the definition of the more common conceptual data, behaviour and functional models (cf. [Yu and Mylopoulos, 1998] for an overview of goal modelling in RE).

In addition, examples of system usage can represent different incarnations of one task fulfilling one specific goal. This makes it hard to compare several examples at a low level of abstraction, as on the system interaction model respectively data model level. Concentrating on goals facilitates the detection of commonalities between different observations. If more detailed knowledge about the achievement of a goal is required (*what* and *how*), a behavioural, functional, data model can be created. Moreover, establishing a goal model in the first place and defining detailed conceptual current-state models only when required reduces effort and time. This is supported by the rule of common RE practice that a current-state model should represent essential abstractions of interesting aspects for the change definition towards the desired system. In general, one can say that the creation of a current-state model should be driven by the change vision of the new system. A complete and detailed reengineering activity is often considered unnecessary and too costly.

Thus, we have chosen goals as the central concept for defining the current-state model. Existing goal modelling approaches differ in the concepts provided for structuring and interrelating goals. In this paper, we use the most common structuring constructs, namely the organisation of goals in hierarchical AND/OR reduction/refinement graphs[2]. The typed interrelations between the goals and the RWEFs can facilitate detailing or remodelling.

The fine-grained interrelation between RWEFs and goal models is realised using typed dependency interrelations. The type expresses information about the analyst's[3] interpretation on the relationship between RWEF and the goal concept. The two most important types for interrelating goals and RWEFs are the link types `Attains` and `Fails`. `Attains` expresses that the analyst interprets the RWEF as an example of how the related goal is fulfilled. Likewise, we support for goal models the link type `Fails`, which expresses the direct opposite to

[2] The choice of another goal model would not affect the approach presented in this paper. Our basic approach of interrelating the goals defined in the goal model with parts of the captured observations of current system usage could be adapted to every type of conceptual (goal) model.
[3] i.e., the person who creates the current-state goal-model as well as the interrelations between RWEFs and goals.

`Attains`, namely that the RWEF shows how a goal is failed. Supplementary to `Attains` and `Fails`, RWEFs jutting out of the set of collected examples for a concept as special reference examples can be marked by the amplifying link types `Positive` (for `Attains`) and `Negative` (for `Fails`). (cf. [Haumer et al., 1998] for a more detailed discussion on link types).

3. Improvements for the Review Processes

As Freedman and Weinberg state in [Freedman and Weinberg, 1982], practice-proven reasons why technical reviews are necessary are because a) to err is human and b) that although some people are good in catching their own errors during technical design, large classes of errors still escape the originator more easily than anyone else. Especially conceptual current-state models have to be validated by reviewers knowing the modelled domain to ensure their correctness and appropriateness. Thus, the review also has to fulfil an educational and communicative purpose, integrating users and domain experts into the analysis process. However, in Section 1 we described major problems of current review processes including stakeholders from different domains. In this section we outline how we utilise Extended Traceability to tackle these problems by

– providing guidance for the review process through the visualisation of recorded trace- and review activity-information allowing to set criteria for review priorities;
– providing means for selectively retrieving rich media background information for reviewed goal concepts, i.e. supporting access to pre-traceability for concepts;
– providing means for recording and visualising review results.

Each feature described will be illustrated by narrative examples drawn from review sessions of a trial application. The trial application was performed to test our approach at a manufacturing company located in Aachen named ADITEC. This trial application was a first experimental case study of using our approach whose results, also due to the participation of the tool developers, should not be over-generalised.

ADITEC is a machine manufacturing company specialised in the production of gears for different types of industrial devices. To support production they use a semi-integrated production management system consisting of a central scheduling system and machine terminals attached to each production machine. In our trial application, the ADITEC management wanted the information flow to be analysed and improved, because of recently increasing problems with incorrect and missing report data.

In this section, we use our tool environment CREWS-PRIME to demonstrate how our technique improves the review process of conceptual models[4]. For simplification, we consider only two stakeholders in the review described below: Peter, the system analyst and Franz, a domain expert being the reviewer.

Before the review took place in our trial application, the RE-analyst Peter has recorded and created a set of RWEs during a site visit at the ADITEC Company. Using the captured RWEs Peter elicited and validated a number a set of goals using the iterative elicitation and validation technique sketched in Section 2.1. As a result, a comprehensive reference base was established providing access to example fragments (RWEFs) for each goal and vice versa. During the review Peter and Franz make use of this reference base to check the goal hierarchy.

3.1 Guiding Review with Relevance and Success Rates for Goals

To review all goals and related real world fragments systematically can be a lengthy and tedious task. To cope with the high number and complexity of the goals presented to the reviewer he needs criteria to prioritise the goals he wants to review.

Using Extended Traceability, we provide means for a rapid aggregated evaluation and visualisation of related RWEFs in form of computed goal model annotations. For the performance of an example-based review, it is important to know how much real world evidence is actually related to a concept. In addition, the reviewer wants to know how relevant the concept actually is within the collected set of real world examples, i.e. how

---

[4] For presenting the parts of the example we have to narrate contents of the real world observation material recorded on video, which is often difficult to understand for non-verbal activities observed, whereas it would be instantly clear watching it on video.

often it has been observed in the examples considered, and especially for goal models how successful a goal was attained in the observed cases of system usage.

The basis for the computed annotation is the number of coherent real world examples (i.e. complete usage situations) considered during elicitation and validation. Relative to this absolute number, we enumerate the amount of examples in which a goal has been tackled, i.e. the existence of interrelations between goal and examples, resulting into the so called *Relevance Index* for each goal. Therefore, the Relevance Index expresses the relevance of the goal in respect to the examples used for elicitation and validation and not the relevance of the goal for the overall system. For instance, a goal for which examples of attainment or failure have been observed in five of the six collected examples will have Relevance Index $^5\!/_6$.

Further, to evaluate how successfully a given goal has been tackled in the observed reality, we provide the *Success Index*. This index relates the number of attainment observations (linked by the `Attains` link type) to the absolute number of observed attainments and failures. For instance, a goal for which two attainments and two failures have been related has a Success Index of ½.

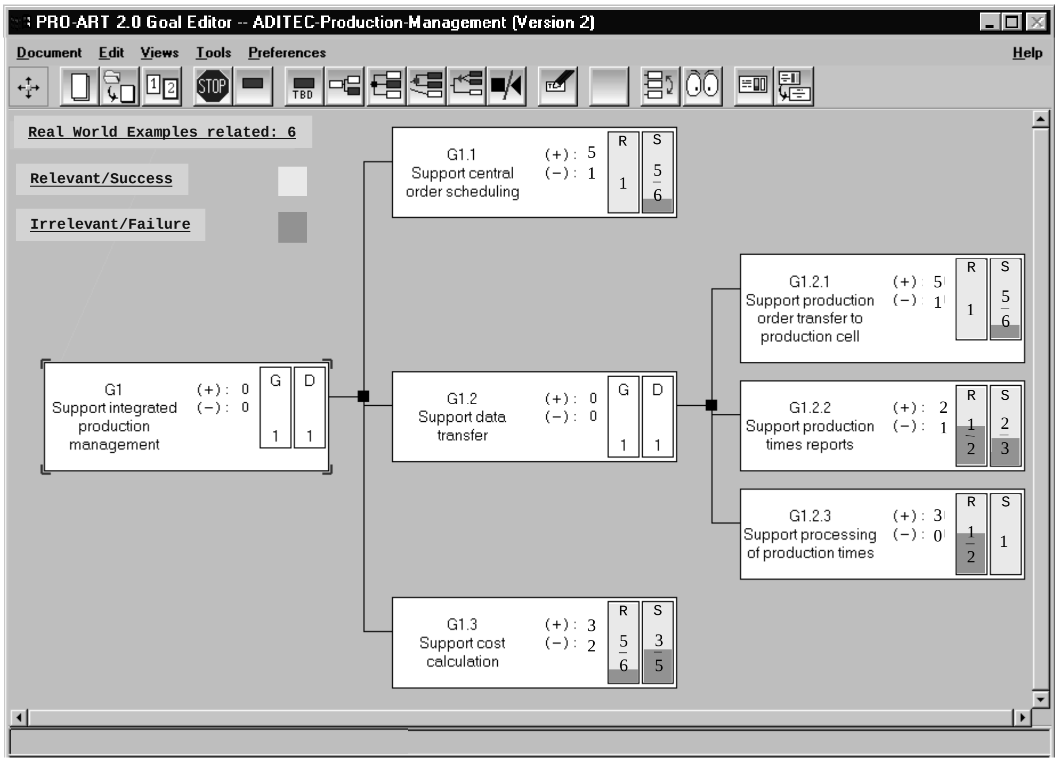

Figure 3. Annotated goal model displaying relevance and success of goals in respect to interrelations.

The two indexes can be used to guide a review as indicators for setting review priorities for resolving possible problems and open issues concerning the current-state model. They can be interpreted in the following ways:

– *High Relevance Index:* By the frequency of which it has been tackled, it is possible that the goal constitutes an important requirement being part of a typical usage situation. The reviewer should check whether he agrees on the appropriateness of the fragments to be called examples for the goal.

– *Low Relevance Index:* Either the goal is an unimportant feature or requirement (i.e. it is not imperative to fulfil it) or it is hard to observe in the captured usage situations. If the reviewer still decides to check on these goals, he should investigate whether he finds further yet unrelated examples for this goal.

– *High Success Index:* Successful attainment of this goal does not seem to be a problem in the existing system. When this goal is also highly relevant, it indicates that an important requirement is already implemented successfully. It is less important for the reviewer to check on these goals, but if he does, he should examine whether he agrees with the interrelation types.

– *Low Success Index:* A goal that is seldom attained but having a super-goal still being achieved might be a hint of an unnecessary goal; especially when combined with a low relevance. The reviewer should check whether he agrees on failures, i.e. the types of the created interrelations.

Note, that the reviewer has to be aware that the indexes only depend on the personal interpretation of the analyst who created the interrelations and assigned their types as well as his thoroughness of really validating all examples against all goals.

***Example*:** Before Franz starts his review, he wants to obtain a quick overview concerning the relevance and success of the goals to be reviewed in respect to the interrelated real world examples. Figure 3 depicts a screen dump of the goal editor displaying the two indexes. In the upper left corner of the window, Franz sees that altogether six examples have been considered during Peter's elicitation and validation session. Each goal is annotated with two bars displaying the Relevance (R-bar) and Success Index (S-bar), respectively, as numbers as well as equally proportioned two-coloured bars (in Figure 3 only displayed as different grey-scales). In addition, for convenience the absolute number of related attainment (+) and failure (-) examples are displayed to keep the reviewer from recalculating these values with the absolute number of examples.

Franz can now quickly decide if and to what degree examples for a goal have been captured during observations and how these observations have been assessed by the analyst or other reviewers before him. Based on this overview he can easily select goals for closer inspection. Franz is surprised that the report of production times (goal *"G1.2.2: Support production times reports"*) has only been observed in three examples (Relevance Index: $^3\!/_6 = {}^1\!/_2$) and among these examples it was unsuccessful in one case (Success Index: $\frac{2}{2+1} = {}^2\!/_3$) as well. Because Franz knows that the worker's report of production times is an integral constituent of the machine terminal protocol and based on the information he just retrieved, he decides to examine this goal in more detail and to start his review here.

3.2 Accessing Recorded System Usage

As outlined in Section 1, many researchers have advocated capturing current-system usage using rich media. Many of the benefits of rich media are also valid for our technique. Persistently captured system usage can be recalled at any time and place. This allows aspects of the current system to be discovered late, for example after several observations of the same recording. Thus, late reflections and their confirmations on the captured recordings are possible. Especially, when rich media such as video have been used, the recordings can provide a better understanding of the usage domain to analysts and stakeholders who did not attend the site visit. Edited video recordings can comprise a focused presentation of temporally and spatially distributed aspects of the considered system. Things which might take hours to observe at several different places can now be surveyed within minutes in front of the computer screen. Edited video can also be used to give a coherent and story-like overview of the system functionality at the beginning of a review session.

However, current techniques of Participatory Design and Ethnography fail to provide a quick and direct access back to the rich media used for the creation of conceptual models. Extended Traceability provides the reviewer with direct access paths into the otherwise unstructured media recordings used for elicitation. The reviewer does not have to view hours of video recordings he might know already or which are not related at all. Instead, he makes use of the interrelations to promptly get the scenes he wants to see.

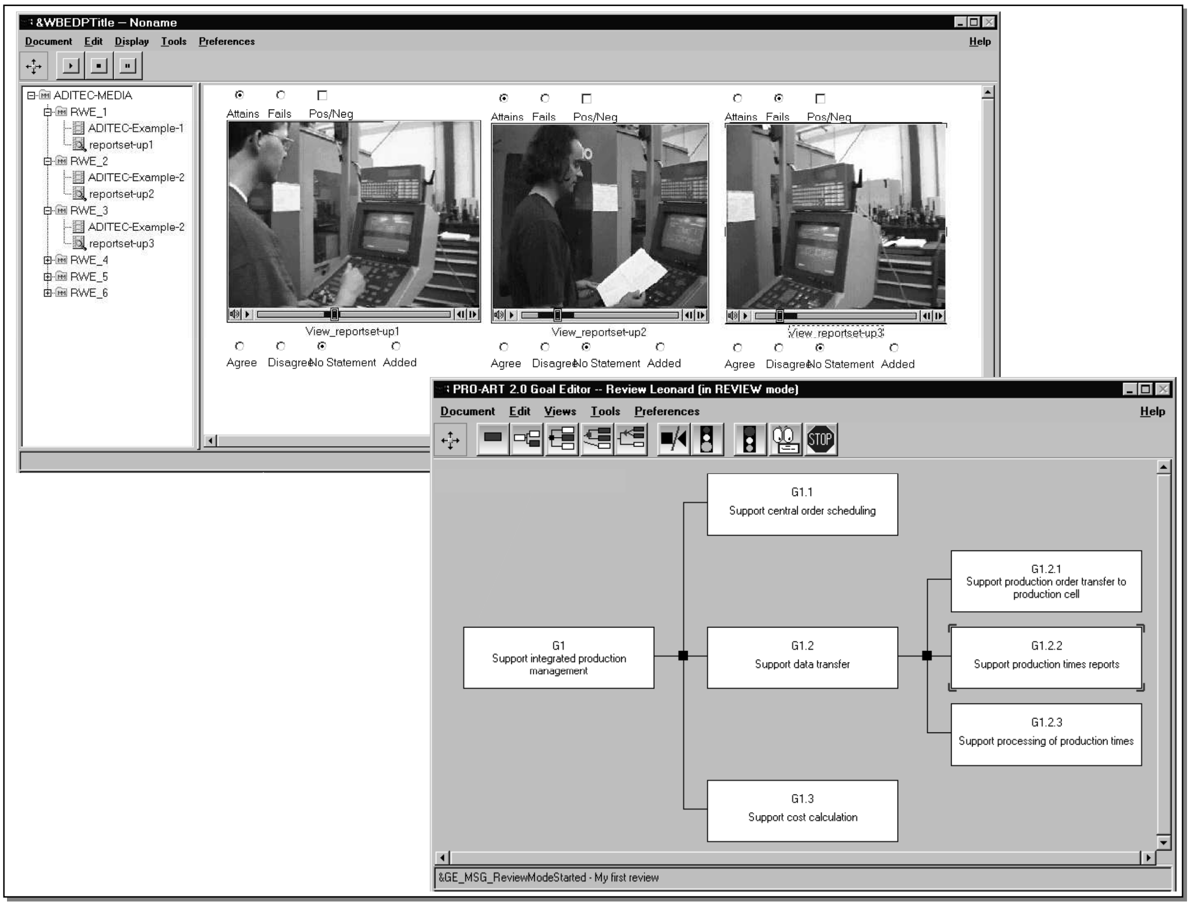

Figure 4. Retrieved RWEFs for goal review.

Example: Figure 4 depicts a snapshot of Franz' review session. He selects goal *"G1.2.2: Support production times reports"* in the Goal Editor (recognisable by the extra edges surrounding goal G1.2.2) for review. The environment retrieves all RWEFs (real world example fragments) related to the selected goal and displays them in the Whiteboard Editor on the left of Figure 4. In this case, the retrieved RWEFs are three video clips showing workers reporting production times. They are annotated with the interrelation types (indicated by the gadgets on top) assigned by Peter during his initial elicitation and validation phase. He found two attaining and one failing example fragments for the report of production times. Thus, Franz can now play-back the three concrete example fragments, but also has the possibility to take a look at what happens before or after theses scenes by using the fast-forward/backward slider of the video gadgets.

3.3 Explanation of Conceptual Models and Real World Examples

Section 1 listed several reasons why reviewers need more information than just the concept itself to perform a review. A major application for Extended Traceability is to provide support for explanation of current-state models to the reviewers. The access structure established between concepts and RWEFs can be applied to retrieve specific examples for a concept which can be used to improve the reviewer's understanding of the concept and his judgement of the correctness and appropriateness about the abstraction made. Using the link types, from Section 2.2, makes it possible to specifically retrieve examples of attaining and failing, respectively, a given goal. In particular, the use of `Positive` and `Negative` links allows illustrating reference examples of goal attainment and failure. These features support new team members and stakeholders, who need not even be familiar with the goal-modelling notation, to be easily and rapidly included into the project by providing answers to their basic questions like "What does this goal mean?", "How is it attained?", or "Why does it fail?"

In contrast, recorded system usage can be quite specific and detailed that it might be useful to have the goals behind the observed actions at hand to quickly assess the purpose of the activities. It is therefore also practical to use the typed interrelations in the other direction to retrieve explanation for the real world examples. For instance, review sessions that start by watching the story-like video RWEs can be continued by retrieving the goals behind certain scenes using our tool environment. Again, for these goals other examples from different usage situations showing attainments and failures could be retrieved in a navigational manner.

Example: After using the typed interrelations to access specific example fragments for the goal *" G1.2.2: Support production times reports"* Franz plays the presented video clips for explanation purposes and further information about the goal. He uses the related RWEFs on the one hand to support his interpretation of the abstract goal concepts with examples for the report of production time. On the other hand, seeing these concrete usage situations from his well-known working environment will easily trigger new thoughts on further insights from Franz about forgotten aspects and problems, which can lead to the expression and elicitation of additional and refined concepts.

3.4 Annotating Concepts and Interrelation Types

Performing the review will lead to review protocols concerning the components of the current-state model themselves and the interrelations to the RWEFs. The reviews are persistently recorded by our tool environment for visualisation and later reference. (e.g. comparison with other review results)

The following types of review statements are recorded:

1. The reviewer can express statements concerning the appropriateness of RWEFs in combination with the interrelation types used to link them to the goal. This statement consists of assigning either *Agree* or the *Disagree* values to the links. In addition, the reviewer can attach a textual comment to each RWEF[5].
2. The reviewer can create additional RWEFs. While reviewing the already related fragments, he also has the possibility to watch the originating RWEs. When he discovers that important captured scenes might be missing for the reviewed goal, he can create new RWEFs, which are linked to the goal using link types selected by the reviewer and which are specially marked as interrelations added by the reviewer.
3. The reviewer can express review statements concerning the goal itself. He can either agree to the necessity and correctness of this goal or disagree, saying that the reviewed goal is not needed or wrong.
4. The review of the goal and its related fragments can lead to the elicitation of new abstractions, which results into modifications of the goal model itself. The goal can be redefined, refined, or proposed for deletion. In addition, the position of the goal and the refinement type (and/or refinement) can be changed. After the goal has been changed, the linked RWEFs can be adapted as well (e.g. by distributing it to refinements of the goal, changing link types etc.).

Statements of the four kinds will be recorded by our CREWS-PRIME environment for each individual review session (several distinct reviews can be performed on the same model) and are later traceable back to this session. Technically, this is achieved by maintaining a version control system for the goal models, interrelations, review statements, and annotations. When several reviews are performed on the same model a so-called *Merg-*

[5] One can attach these textual comments to every fragment and goal; they are therefore not mentioned for the following cases anymore.

ing Session will finally present all review results and allows the creation of a newly negotiated version of the goal model.

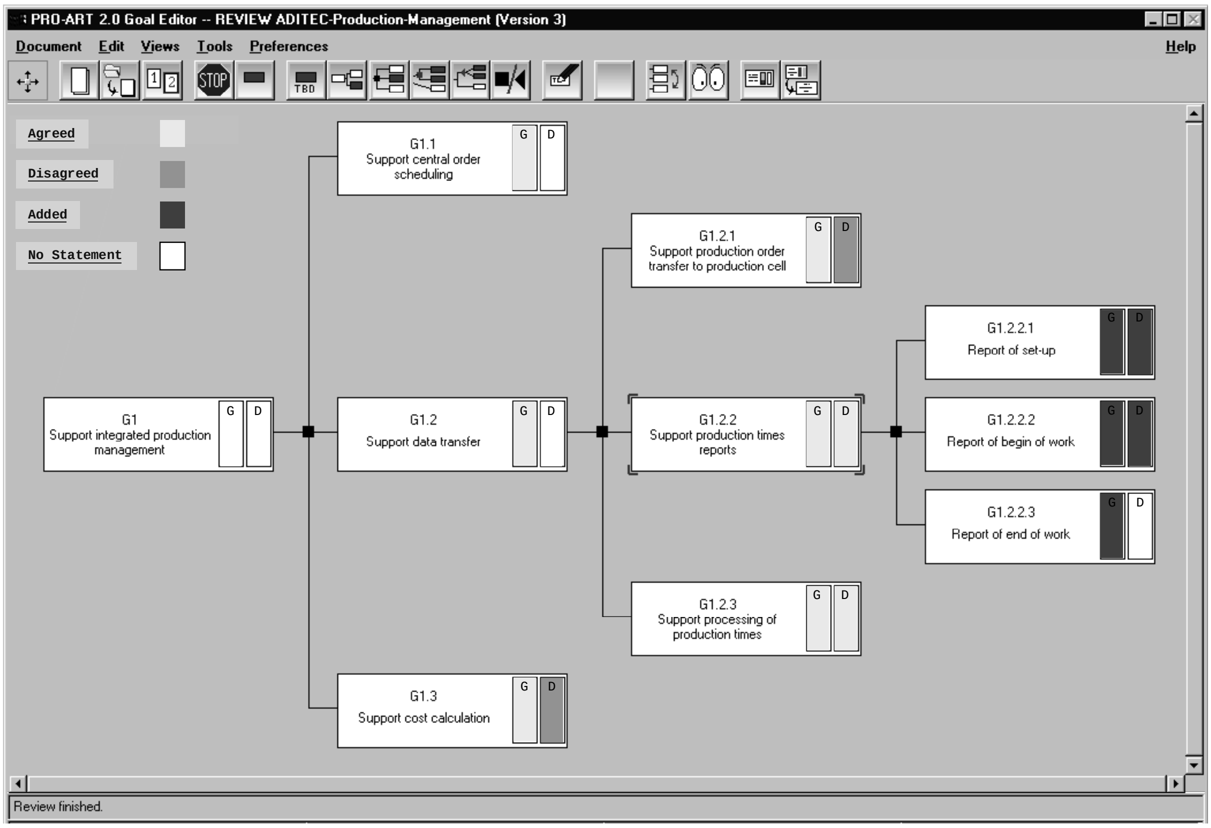

Figure 5: Visualisation of review results.

Example: For the RWEFs retrieved in Section 3.2 and goal G1.2.2 itself, CREWS-PRIME offers Franz to record one or more of the possible review statements (see gadgets under the video clips in Figure 4). First, he decides that the related RWEFs and interrelation types were appropriate, which he confirms by selecting the *Agree* statements for each clip. Further, he decides that goal G1.2.2 is correct, but it is not refined in enough detail because the system has to differentiate several types of production times. Thus, he introduces goals G1.2.2.1 to G1.2.2.3 as sub-goals of G1.2.2 (see Figure 5) and relates the respective parts of the RWEFs to each newly added goal.

3.5 Visualisation of Conflicts between Reviewer and Model Designer

The results of a finalised review session are visualised by another form of annotation for the goal model. Colour codes provide a quick overview of the results to other reviewers or the original model designer.

The annotation consists of a coloured ***D***-bar for review statements of cases (1) or (2), which were related to RWEFs and the type of the ***D***ependency interrelation and a second coloured ***G***-bar related to choices of case (3) and (4) concerning the ***G***oals themselves. The colouring scheme used for these bars is quite simple and provides a quick overlook of the results (again, we can only provide different grey-scales for each colour in the figures due to publication limitations):

– *Green* colour denotes that the reviewer agreed on the reviewed products;
– *Red* colour indicates that some problems occurred during review in the way that he disagreed on the product (one of the displayed dependency or the goal) and further actions have to be taken to resolve this problem before or during the merging session;
– *Blue* colour displays that something has been added, either further RWEFs or new refinements of goals;
– *White* shows that this product has not been reviewed at all or that the reviewer did not want to make a statement.

With a simple look, a stakeholder who did not participate in the review is now able to get a quick impression of the reviewer’s statements and find the goals that need further consideration and discussion with the model designer.

Example: Franz continues his review on the goal model. After finishing, the results are displayed using the visualisation feature as depicted in Figure 5. All goals are now annotated with D- and G-bars displaying the results of Franz' review. For example, the bars of goal G1.2.2 have been coloured green because of Franz' agreement described in Section 3.4. The newly introduced sub-goals are marked blue to reflect that they have been added together with new RWEFs. Franz presents this display to Peter and they start further discussion on the production times report.

4. Conclusions

In this paper, we presented a novel technique to support the review of conceptual models utilising what we call Extended Traceability. We described how the review process is improved by making abstract concepts better understandable and assessable for the reviewer through the fine-grained interrelation with persistently recorded usage situations. We sketched the tool support provided and demonstrated its application in reviews. We illustrated examples for the use of our environment in a review process drawn from a case study.

In [Haumer et al., 1999], we describe experimental research undertaken to further validate our approach. We tested review situations in which people joining an on-going project had to aquatint themselves with goal models with and without Extended Traceability. As the results indicate, Extended Traceability leads to an increase in performance (in respect to completeness and correctness) of specific RE-tasks the test persons had to perform; for instance, identification of possible influence factors in the existing system for a specific goal.

Future work will investigate further applications of our technique, such as the interrelation to captured visionary explorations of future systems. Approaches like the one introduced by Nicola Millard in [Millard et al., 1998] in which she captures role-play and storyboards of stakeholders exploring future telecommunication systems can be extended with our technique. Moreover, we will adapt our technique to elicit models for future systems from the captured role-play and storyboards as well as their review and negotiation using our tool environment.

Finally, educational training introducing stakeholders either into the basic functionality of the existing or desired system or for teaching the modelling language in an example based manner.

Acknowledgements. This work is funded by the European Community under ESPRIT Reactive Long Term Research 21.903 CREWS (Cooperative Requirements Engineering with Scenarios, http://Sunsite.Informatik.RWTH-Aachen.DE/CREWS/) and by Deutsche Forschungsgemeinschaft through the PRIME project (445/5-1). The authors are grateful to their students Sebastian Brandt, Sven Ewald, Markus Hoofe, Michael Mattern, and Marco Pesarese for implementing major parts of the tool environment. We also would like to thank the ADITEC team, especially Franz-Josef Stepprath and Silvia Schlaphof.

5. References

[Anton, 1996]
A.I. Antón, "Goal-based Requirements Analysis", Proc. Int’l Conf. Requirements Eng. (ICRE'96), IEEE Computer Soc. Press, Colorado Springs, Colorado, USA, pp. 136-144.

[Beyer and Holtzblatt, 1993]
H. Beyer and K. Holtzblatt, *Contextual Design: Defining Customer-Centered Systems.* Morgan Kaufmann Publishers, 1997.

[Blomberg, 1993]
J. Blomberg, J. Giacomi, A. Mosher, P. Swenton-Wall, “Ethnographic Field Methods and their Relation to Design”, in *Participatory Design: Principles and Practices.* D. Schuler, A. Namioka (eds.), Lawrence Earlbaum Associates Inc., Hillsdale, New Jersey, 1993, pp. 123-155.

[Bødker and Pedersen, 1991]
K. Bødker and J.S. Pedersen, “Workplace Cultures: Looking at Artifacts, symbols and Practices”, in *Design at Work: Cooperative Design of Computer Systems.* J. Greenbaum, M. Kyng (eds.), Lawrence Earlbaum Associates Inc., Hillsdale, New Jersey, 1991, pp. 121-136.

[Brun-Cottan and Wall, 1995]
F. Brun-Cottan and P. Wall, “Using Video to Re-present the User”, *Communications of the ACM*, Vol. 38, No. 5, Mai 1995, pp. 61-71.

[Dardenne et al., 1993]
A. Dardenne, A. van Lamsweerde, S. Fickas, "Goal-directed Requirements Acquisition", *Science of Computer Programming* Vol. 20, No. 1-2, 1993, pp 3-50.

[Freedman and Weinberg, 1982]
D.F. Freedman and G.M. Weinberg, Handbook of Walkthroughs, Inspections, and Technical Reviews: Evaluating Programs, Projects, and Products, Dorset House Publishing, New York, 1982.

[Gause and Weinberg, 1989]
Donald C. Gause and Gerald M. Weinberg. *Exploring Requirements: Quality before Design.* Dorset House Publishing, New York, 1989.

[Gotel and Finkelstein, 1994]
O. Gotel and A. Finkelstein, “An Analysis of the Requirements Traceability Problem”, *Proc. 1st Int’l. Conf. on Requirements Eng*., Colorado Springs, USA, April 1994, pp. 94-102.

[Haumer et al., 1998]
P. Haumer, K. Pohl, and K. Weidenhaupt. “Requirements Elicitation and Validation with Real World Scenes”, in *IEEE Transactions on Software Engineering*, Vol. 24, No. 12, Special Issue on Scenario Management, 1998, pp. 1036-1054.

[Haumer et al., 1999]
P. Haumer, J. Rack, K. Pohl, "Interrelating Goal Models and Multimedia Scenes: An Empirical Investigation", *Proc. of 5th Int’l. Workshop on Requirements Eng.: Foundation for Software Quality (RESFQ99)*, Heidelberg, Germany, June 14-15, 1999, pp. 83-98.

[Hughes et al., 1995]
J. Hughes, J. O'Brien, T. Rodden, M. Rouncefield, I. Sommerville. “Presenting Ethnography in the Requirements Process.”, *Proc. 2nd IEEE Int’l. Symp. on Requirements Eng.,* York, England, IEEE Computer Society Press, 1995, pp. 27-34.

[Jackson, 1995]
M.A. Jackson, *Software Requirements & Specifications — a lexicon of practice, principles and prejudices.* Addison Wesley Press, 1995.

[McGraw and Harbison, 1997]
K. McGraw and K. Harbison, *User-Centered Requirements: The Scenario-Based Engineering Process.* Lawrence Erlbaum Associates, Inc., Publishers, Mahwah, New Jersey, 1997.

[McMenamin and Palmer, 1984]
S.M. McMenamin and J.F. Palmer, *Essential System Analysis.* Prentice Hall, 1984.

[Millard et al., 1998]
N. Millard, P. Lynch, and K. Tracey. “Child's Play: Using Techniques Developed to Elicit Requirements from Children with Adults”, *Proc. of 3rd Int’l. Conf. on Requirements Eng. (ICRE '98),* Colorado Springs, USA, April 6-10, 1998.

[Pinheiro and Goguen, 1996]
F.A.C. Pinheiro and J.A.Goguen, "An Object—Oriented Tool for Tracing Requirements", *IEEE Software*, Mar. 1996, pp. 52-64.

[Pohl, 1996a]
K. Pohl. "PRO-ART: Enabling Requirements Pre-Traceability", *Proc. of the 2nd Int'l. Conf. on Requirements Eng.*, Colorado-Springs, Colorado, USA, April 1996.

[Pohl, 1996b]
K. Pohl, *Process Centered Requirements Engineering.* RSP marketed by J. Wiley & Sons Ltd., England, 1996.

[Ramesh et al., 1997]
B. Ramesh, C. Stubbs, T. Powers, M. Edwards, "Implementing Requirements Traceability: A Case Study", *Annals of Software Engineering*, Vol.3, 1997, pp 397-415.

[Suchman and Trigg, 1991]
L.A. Suchman and R.H. Trigg, "Understanding Practice: Video as a Medium for Reflection and Design", *Design at Work: Cooperative Design of Computer Systems*, J. Greenbaum and M. Kyng, eds., Lawrence Erlbaum Associates, Inc. Publishers, New Jersey, 1991, pp 65-89.

[Watkins and Neal, 1994]
R. Watkins and M. Neal, "Why and How of Requirements Tracing", *IEEE Software*, Jul. 1994, pp. 104-106.

[Weidenhaupt et al., 1998]
K. Weidenhaupt, K. Pohl, M. Jarke, and P. Haumer. "Scenario Usage in System Development: A Report on Current Practice", *IEEE Software*, Mar., 1998, pp. 34-45.

[Weiser and Morrison, 1998]
M. Weiser, J. Morrison, "Project Memory: Information Management for Project Teams", *Journal of Management Information Systems*, Vol 14., No. 4 , 1998, pp. 140-160.

[Yu, 1994]
E. Yu, *Modelling Strategic Relationships for Process Reengineering.* PhD thesis, Technical Reports on Research in Data and Knowledge Engineering, DKBS-TR-94-6, University of Toronto, Department of Computer Science, 1994.

[Yu, 97]
E. Yu, "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering", *Proc. IEEE Int'l. Symp. Requirements Eng.*, Annapolis, Maryland, Jan., 1997, pp. 226-235.

[Yu and Mylopoulos, 1998]
E. Yu and J. Mylopoulos, "Why Goal-Oriented Requirements Engineering", *4th Int'l. Workshop on Requirements Eng.: Foundation for Software Quality (RESFQ'98)*, Pisa, Italy, Jun., 1998, pp. 15-22.