CREWS Report 98-17

Scenario-Based System Development (A mini-track within the Emerging Technologies Track of the Thirty-second Annual Hawaii International Conference on Systems Sciences (HICSS-32),Maui, HI, January 5-8, 1999.

Improving Reviews by Extended Traceability

by

Peter Haumer, Klaus Pohl, Klaus Weidenhaupt, Matthias Jarke

Lehrstuhl Informatik V, RWTH Aachen
Ahornstraße 55, 52056 Aachen, Germany

+49 (0)241 80 21 501
{haumer,pohl,weidenhaupt,jarke}@informatik.rwth-aachen.de

Improving Reviews by Extended Traceability

Peter Haumer, Klaus Pohl, Klaus Weidenhaupt, Matthias Jarke

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

Abstract

When defining a new system, the history and functionality of the system to be replaced should be considered. This avoids repeating errors and neglecting important system functionality.

The properties and the rationale behind the existing system are typically elicited by analysing concrete system usage-scenarios. The results of the analysis of the existing system are 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 can not assess and therefore understand the basis (concrete system usage) on which the conceptual models where build.

In this paper, we present an approach to overcome this problem. We argue to 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 resulting from the extended traceability and illustrate the advantages with excerpts from a case study performed in a mechanical engineering company.

1. Introduction

Traditionally, 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 the existing system to be replaced has to be considered (cf. McMenamin and Palmer [13] or Gause and Weinberg [7]). There are two main reasons for this: a) the new system has to provide to a large degree the functionality of the old system and has to obey the same context of constraints; b) one can learn a lot from the success stories and pitfalls of the existing system. Considering the history and functionality of the existing system avoids repeating failures or forgetting an important functionality. Consequently, many RE approaches (e.g., [13], [11]) consider two categories of conceptual models:

- *Current-state* models, which (partially) express essential properties and the functionality 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 for the quality of current-state models because only the stakeholders in place (e.g. domain experts, system users, maintenance people etc.) have the fundamental knowledge about the system in use.

To ensure correctness and appropriateness of current-state models, they must be validated. A common technique for validation in industry is reviewing. Typically, the defined model is passed to one or more stakeholders to check the validity of the model.

Reviewing is, however, burdened with a major problem: Conceptual models are hard to understand for somebody not involved in the model definition. The abstractions made by the “designer” of the model are not traceable, i.e. the reviewer can not assess and therefore understand the basis on which an abstraction was made. For example, the definition of a model component could have resulted from an observation of current system usage the

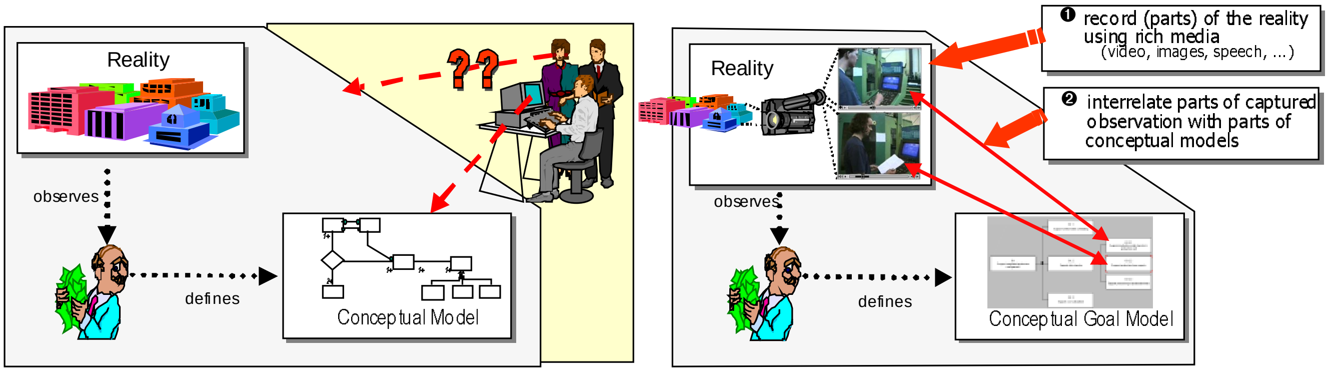

Figure 1 (left). Problem: Abstractions are not understandable
Figure 1(right). Solution: Extended traceability by interrelating parts of observations with parts of models

reviewer is not aware of. Moreover, the same system usage could lead to different conceptual models caused by the personal perception of the people defining the model. Consequently, without traceability back to the origins of the abstractions, current-state models are hard to justify. This is illustrated in Figure 1 (left). The analyst depicted at the lower left corner has produced the conceptual model on the lower right corner 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. Therefore, the group of people depicted in the upper right corner of the figure can hardly assess quality and correctness of the conceptual model during a review.

To overcome this problem, the use of rich media to record and discuss current-system usage is proposed in several publications (e.g., video-supported Participatory Design techniques [5], [20]; the Scenario-Based Engineering Process (SEP) [12]; Contextual Inquiry [2]; ethnography approaches [3], [10]; and workplace culture approaches [4]). All these 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. However, these techniques neglect the integration of the observed system usage with the more traditional and common RE methods. In other words, the recorded observations are not interrelated with the abstract defined requirements, i.e. conceptual current-state models. Consequently, the abstractions made by the stakeholder are not traceable although the basis for the abstraction has been recorded using rich media.

In this paper, we present an approach to overcome this shortcoming and thereby improve the review processes. The underlying idea for a solution is depicted in Figure 1 (right) which illustrates what we call *extended traceability*. We aim to interrelate (step 2 of Figure 1 (right)) the recorded real world observations gained through step 1 with the abstractions made based on those observations. More precisely the parts of a conceptual model, e.g. a single goal of a hierarchical goal model, should be related to exactly those parts of the observations, e.g. a cut-out fragment of a video clip, that have influenced its definition. The interrelations allow direct access of the relevant parts of the observations to make abstractions of these observations more transparent, even on a very fine-granular level.

The established interrelations result in a special form of requirements pre-traceability[1] that we call *extended traceability*. This type of requirements pre-traceability extends existing traceability approaches (e.g., [16], [18], [21], [22]) in the way that we a) trace back to concrete instance example from the real world and b) trace on a very fine-grained manner allowing interrelations of arbitrary parts of conceptual models with arbitrary parts of real world observations and not just interrelate on a document level.

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

[1] Requirements pre-traceability: tracing requirements back to their origins in contrast to requirements post-traceability which traces the realisation of requirements, e.g. in design and implementation. (cf. [8] for an overview requirements traceability)

2. Establishing Extended Traceability

In Figure 2, we elaborate our approach for establishing extended traceability in more detail than in Figure 1 (right) [9].

Current system usage is recorded on-site using rich media (step 1 of Figure 2) in what we call *Real World Scenes (RWS)*. The material gathered during an observation may contain information about many system usages. We therefore pre-structure this material into what we call a *Real World Example (RWE)*. An RWE is a collection of material that represents *one* coherent and complete observed system usage. The material belonging to an RWE should be arranged in a suitable manner, e.g. if the observation was recorded using video, the video should be cut in a way that it shows the temporal sequence of a sample system usage

RWEs are used for two main purposes. On the one hand, new concepts can be elicited from RWEs. On the other hand, existing current-state model concepts can be validated against them. In both cases, we interrelate in a fine-grained manner corresponding *Real World Example Fragments (RWEF*) of the RWEs with the concepts of the current-state model (step 2 of Figure 2), 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 a whole one hour 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 if e.g. somebody wants to achieve an explanation for the model.

Note, that we do not propose a new modelling technique, but argue to embed an appropriate existing technique in our overall approach. Similarly, we reuse existing guidelines for capturing (e.g. [12]) and pre-structuring real world scenes into real world examples.

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 (e.g. "Why does the system support this activity?"), before dealing with details about how and what [1], [6], [24], e.g. the data the system deals with, system function and/or system behaviour. Consequently, more and more RE frameworks suggest the explicit definition of goal models prior to the definition of the more common conceptual data, behaviour and functional models (cf. [25] 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, and/or object-oriented model can be created.

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

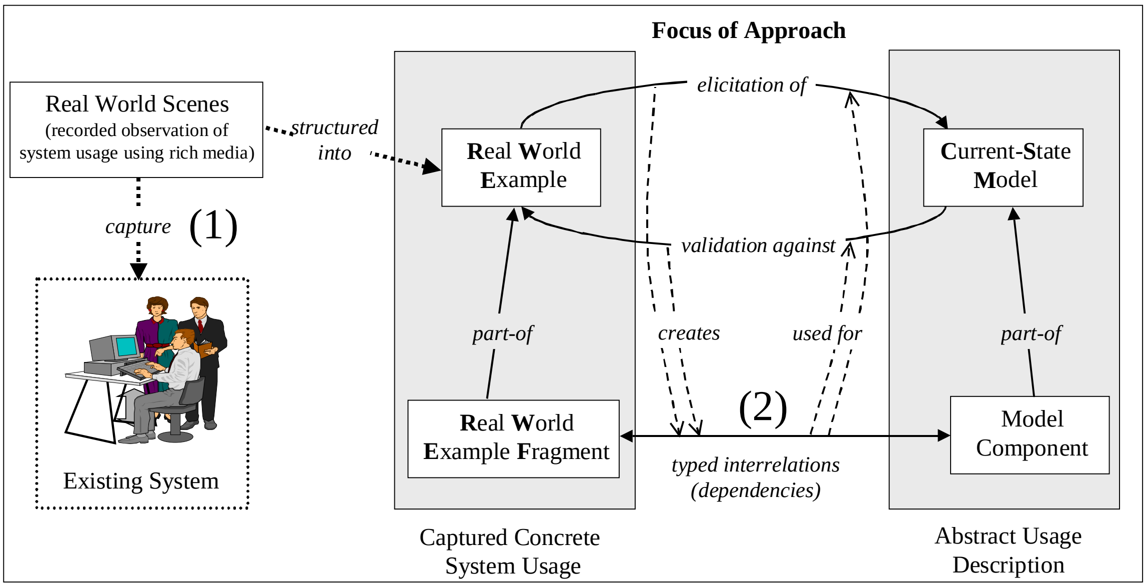

Figure 2. Schematic overview of extended traceability

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 the detailing or re-modelling. Moreover, establishing a goal model in the first place and defining detailed conceptual current-state models whenever required, reduces effort and time to be spent and is thus more cost effective.

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 of the relationship the RWEF and the goal concept have. The two most important types for interrelating goals and RWEFs are the link types `Attains` and `Fails`. `Attains` expresses that the analyst interprets the fragment 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 `Attains`, namely that the example shows how a goal is failed. Supplementary to `Attains` and `Fails,` fragments 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. [9] for a more detailed discussion of link types). Consequently, the established interrelation structure imposes on the one hand access paths upon the real world examples; i.e. a set of real world example fragments typed by the incident links is gained from initially totally unstructured multimedia material. On the other hand, the goal model is annotated by a set of real world evidences that are close to the perception of the involved stakeholders. In the following Sect. 3, we describe how this structure can be used to improve the reviewing process.

For persistent recording of the extended trace information we adapted the trace repository developed for PRIME[4] (see [15] and [16] for details). Our extension called CREWS-PRIME manages all products and traceability information in a (logically) central repository. The tools built on top of this repository are shown in the screen dumps illustrating the exemplary review session described throughout Sect. 3. A detailed description of the process-integrated tool environment CREWS-PRIME can also be found in [9].

[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] The person who creates the current-state goal-model as well as the interrelations between RWEFs and goals.

[4] PRIME: PRocess Integrated Modelling Environment.

3. Improvements for the Review Processes

In the following, we outline the basic principles of the main features of our approach for improving the reviewing process.

Each feature will be illustrated by applications in an accompanying exemplary review session. This will be narrated in a scenario-based manner describing parts of a small trial application we performed (find details in [9]) at a manufacturing company located in Aachen named ADITEC[5]. Using this example, we introduce our tool environment CREWS-PRIME demonstrating how our technique and CREWS-PRIME are applied and which benefits they provide.

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 scenario, the ADITEC management wants the information flow to be analysed and improved, because of recently increasing problems with incorrect and missing report data.

During this section, we will step into the example after the RE-analyst James has collected and recorded a set of examples during a recent site visit at the ADITEC Company. Using the captured real world examples James elicited a number of goals respectively validated these goals against other examples of the set. As a result, a comprehensive reference base is established providing access to example fragments (RWEF) for each goal and vice versa. In the example, James meets with Leonard, the software administrator of ADITEC who uses this reference base to review James findings about understanding and modelling the current system as a 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. Using extended traceability we provide means for a rapid aggregated evaluation and visualisation of related example fragments in form of computed goal model annotations. Thus, 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 col-

[5] 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. As you recall, exactly this fact is actually one of the reasons, why we proposed the use of video and other rich recordings instead of textual representations for our technique.

lected set of real world examples 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. 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. E.g. a goal which has been observed in five of the six collected examples will have Relevance Index $\frac{5}{6}$.

Further, to evaluate how successful a given goal is 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. E.g. a goal for which two attainments and two failures have been related has a Success Index of ½. The example described in the next section will show how both indexes will be visualised providing a quick overview of the calculated values for a goal model (see Figure 3 for a screen dump).

The two visualised 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:

- *Low Relevance Index:* Either the goal is an unimportant requirement (i.e. it is not imperative to fulfil it) or it is hard to observe in the captured usage situations. The reviewer should check if he finds further yet unrelated examples for this goal.
- *High Relevance Index:* By the frequency of which it has been tackled, it is possible that the goal constitutes an important requirement of which achievement is part of a typical usage situation. The reviewer should check if he agrees on the appropriateness of the fragments to be called examples for the goal.
- *Low Success Index:* When the goal is seldomly attained but a super-goal still achieved than this might be a hint of an unnecessary goal, especially when combined with a low relevance. The reviewer should check if he agrees on failures, i.e. the types of the created interrelations.
- *High Success Index:* Successful attainment of this goal does not seam to be a problem in the exiting system. When this goal is also highly relevant it indicates that an important requirement is already implemented successfully. The reviewer should check if he

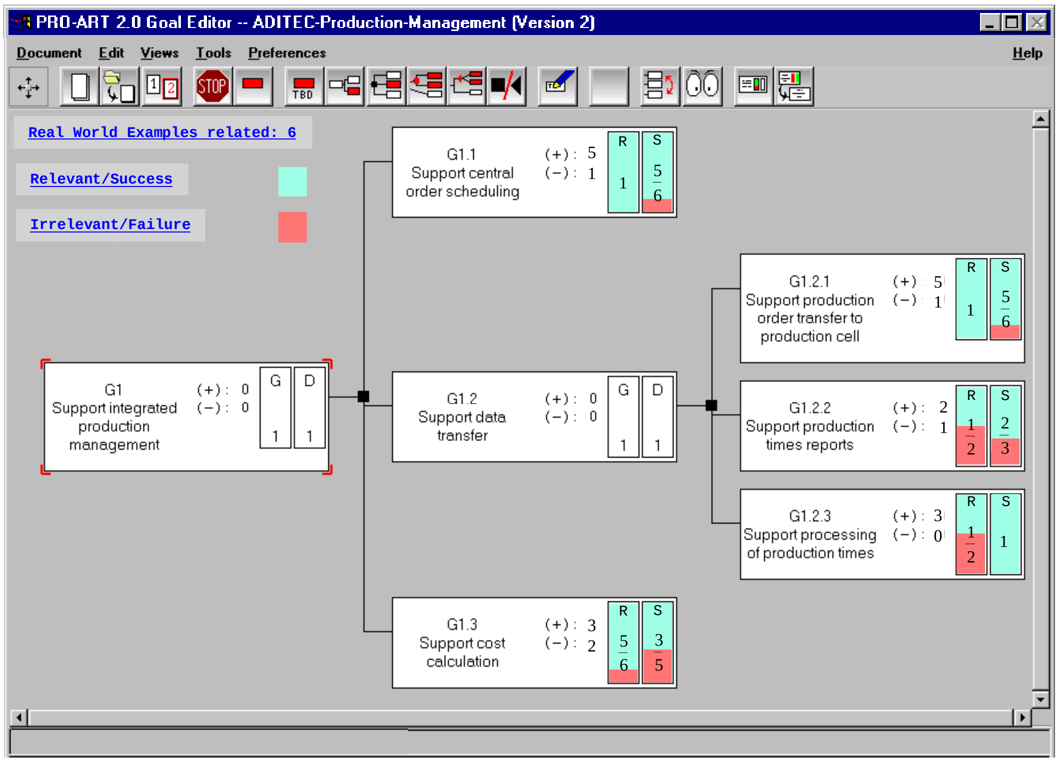

Figure 3. Annotated goal model displaying relevance and success of goals in respect to interrelations.

agrees on attainment success, i.e. the types of the created interrelations.

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

***Example*:** Moving on to the example; before Leonard starts his actual review, he wants to obtain a quick overview about relevance and success of the goals at stake in respect to the used examples. Figure 3 depicts a screen dump of the goal-modelling editor displaying the two indexes. In the upper left corner of the window, one can see that altogether six examples have been considered during James' elicitation and validation session. Each goal is annotated with two bars displaying the Relevance (R-bar) respectively Success Index (S-bar) as numbers as well as equally proportioned two-coloured bars. 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.

Using the annotations Leonard can now quickly decide if and to which 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 he wants to take a closer look at. Here, Leonard is surprised that the report of production times (goal *"G1.2.2: Support production times reports"*) has only be observed in three examples (Relevance Index: $\frac{3}{6} = \frac{1}{2}$) and here it was unsuccessful in one case (Success Index: $\frac{2}{2+1} = \frac{2}{3}$) as well. Because reporting seamed to be 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 already mentioned in the introduction, many approaches as Participatory Design realised the advantage of

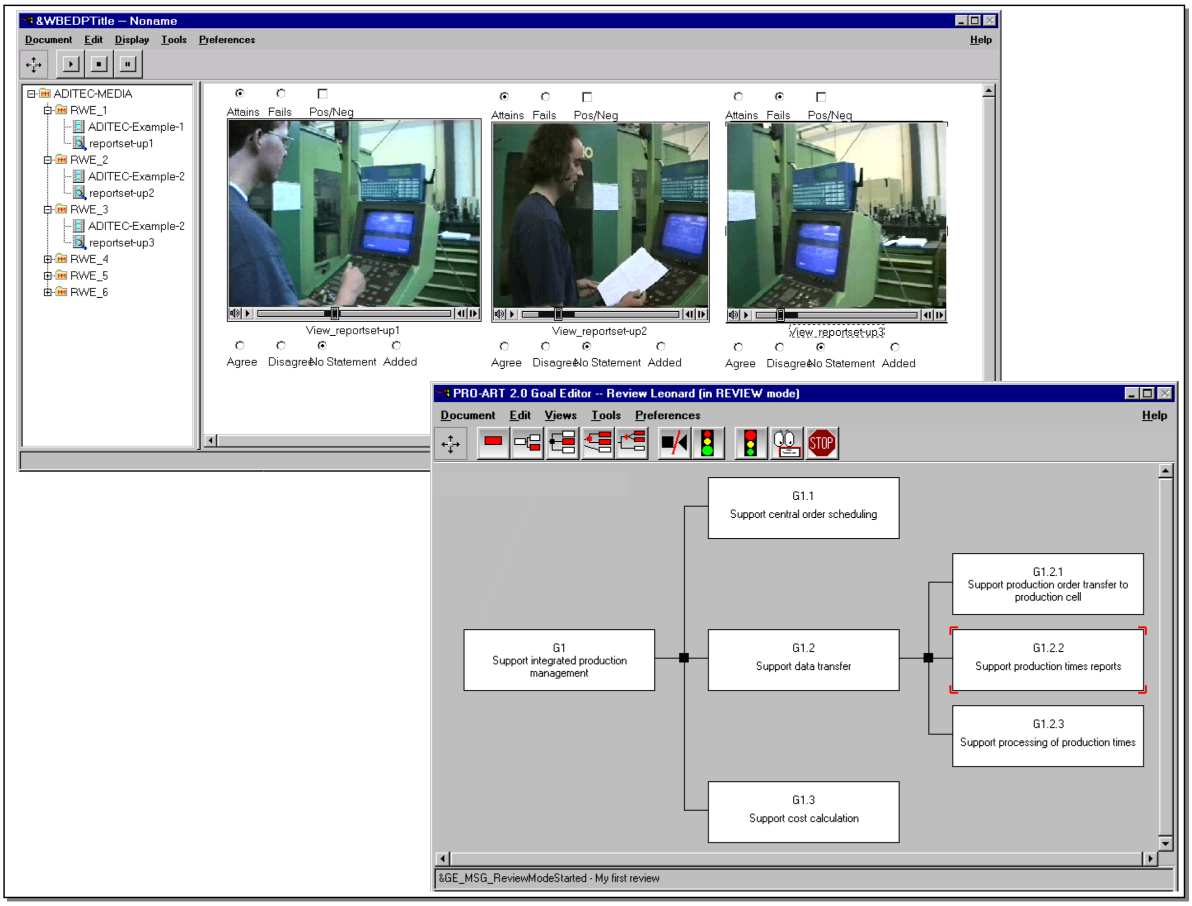

Figure 4: Retrieved RWEFs for goal review

capturing current system usage using rich media. Many of the benefits identified there are also valid for our approach.

Capturing observed system usage in a persistent way has the advantages that it can be recalled at any time and place. This supports the opportunity that aspects of the current system might be discovered late, e.g. after several observations of the same recording. In addition, late reflections and their confirmations on the captured recordings are possible. Especially, when expressive media like video has been used the recordings can provide a better understanding of the usage domain to analysts and stakeholders who did not attend the site visit without presumptuous abstractions and omissions the current-state model might have. When the video recordings are prepared, they can comprise a focused presentation of temporally and spatially distributed aspects of the considered system, e.g. things which might take hours to observe at several different places can now be surveyed within minutes in front of the computer screen. Prepared video can also be used to give a coherent and story-like overview of the system functionality at stake at the beginning of a review session.

A main improvement introduced by our technique for reviewing specific components of current-state models is established through the systematic interrelation of the models components with parts of the recordings during elicitation. The interrelations provide access paths to the otherwise unstructured media recordings, which can be used as an example reference base for the concepts. This is, while reviewing a specific concept the reviewer does not have to view hours of video recordings he might know already or which are not related to the concept at stake. Instead, he makes use of the interrelations to get the scenes he wants directly.

Example: Figure 4 depicts a first snapshot of the Leonard's reviewing session. In the editor the goal *" G1.2.2: Support production times reports"* has been selected (recognisable by the extra edges surrounding goal G1.2.2) and is now being reviewed. To support this task the environment retrieved all real world example fragments related to the selected goal and displays them in the Whiteboard Editor on the left of Figure 4. The retrieved fragments in this case are three video clips showing workers reporting production times. They are annotated with the interrelation types (indicated by the gadgets on top) assigned by James during his initial elicitation and validation phase. Thus, James found two attaining and one failing example fragments for the report of production times.

3.3 Explanation of Conceptual Models and Real World Examples

The main application for extended traceability provided by the typed interrelations is to support explanation of the current-state model to the reviewer. As illustrated above the access structure established between concepts and real world example fragments 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, we introduced for goal models in Sect. 2, makes it possible to retrieve specifically examples of attaining and failing a given goal. In particular, the use of `Positive` and `Negative` links allows illustrating reference examples of goal attainment and failure, respectively. Using these features allows that new team members and stakeholders, who are even not familiar with the goal-modelling notation, can be easily and rapidly drawn into the project by providing answers to their basic questions like "What does this goal mean?" and "How is it attained or failed?"

On the other hand, recorded system usage can be quite specific and detailed that it therefore might be useful to have the goals behind the observed actions at hand to quickly assess the purpose of the activities. It is therefore practical to use the typed interrelations in the other direction, as described above, to retrieve explanation for the real world examples. For instance, the review session we suggested in Sect. 3.2 which started by watching the story-like video example can be continued by accessing and displaying the goals behind certain scenes using our tool environment. For the retrieved goals other examples from different usage situations showing attainments and failures could be retrieved using the features described in the last paragraph.

Example: After using the typed interrelations to access specific example fragments for the goal *" G1.2.2: Support production times reports"* Leonard plays the presented video clips for explanation and further information about the goal.

The related RWEFs are used on the one hand to explain the abstract goal concept with concrete examples to Leonard. On the other hand, seeing these concrete situations from his well-known working environment will easily trigger new thoughts on further insights from Leonard 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

As suggested above, performing the review will lead to review statements concerning the components of the current-state model themselves or the types used for the interrelations. Review statements on these artefacts will be persistently recorded by our tool environment for later references, e.g. comparison with other review results. The following types of review statements are recorded:

1. The reviewer can express a review statement concerning the appropriateness of example fragments 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 could attach a textual comment to each RWEF[6].
2. The reviewer can create additional real world example fragments. While reviewing already related fragments, he also has the possibility to watch the originating coherent examples. When he discovers that important captured scenes might be missing for the reviewed goal he can create new fragments, which are then linked to the goal using a selected link type.
3. The reviewer can express a review statement 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 knowledge that results into modifications of the goal model itself. The goal can be redefined, refined or deleted. In addition, the position of the goal and the refinement type (and/or) 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.

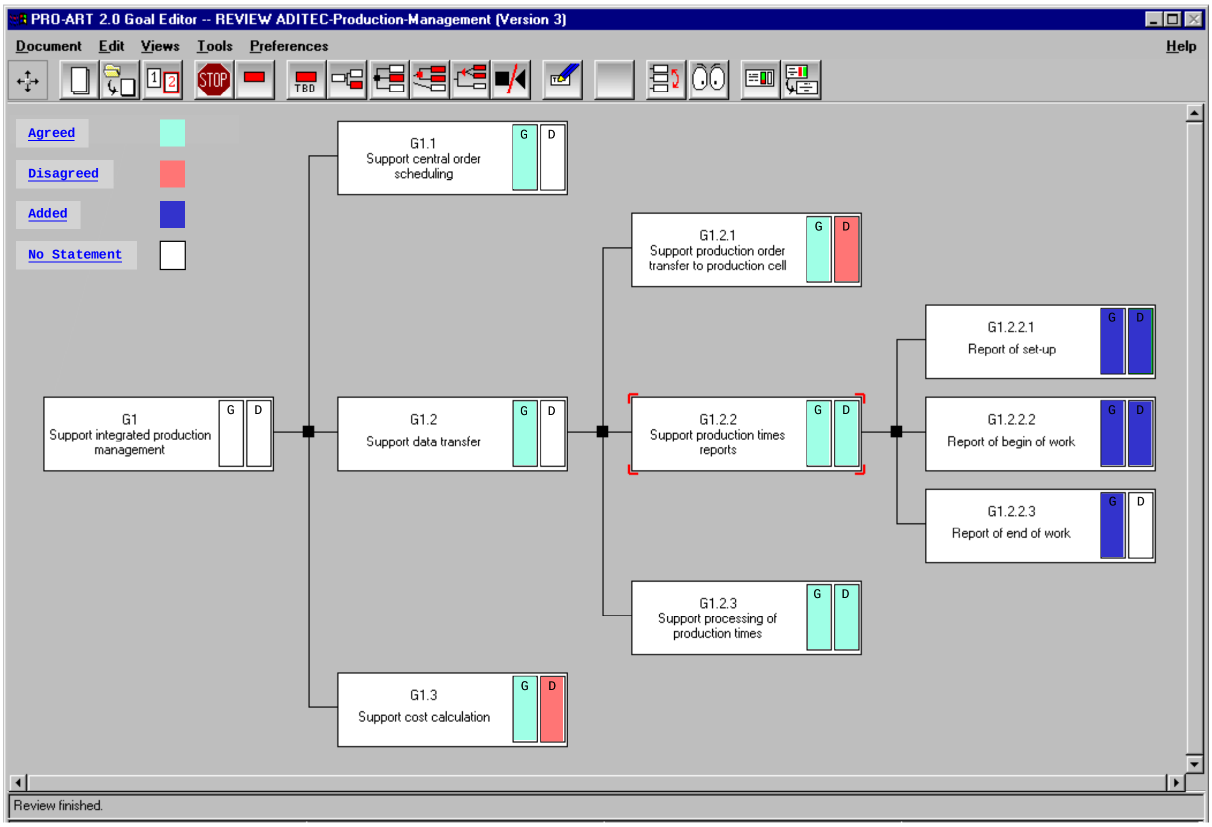

Figure 5. Visualisation of review results.

No matter which case the reviewer chooses in his review all reactions will be recorded by our modelling environment specifically traceable for this review session. This is achieved by applying the following principle: Performing a review session is considered as a transactional unit in the modelling environment recording all review actions made by the reviewer as well as all changes to the conceptual models. This transactional nature of a review session allows several reviews of the same state of a model to be performed by several different reviewers. A so-called merging session will finally present all reviewing results and allows fixing a newly negotiated version of the goal model.

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

Example: For the RWEFs retrieved in Sect. 3.2 and the goal itself CREWS-PRIME offers Leonard to perform one or more of the possible review statements (see Figure 4). First, he decides that the related example fragments were appropriate which he confirms by making *Agree* statements for each clip. Further, he decides that goal G1.2.2 is correct, but it is not fine grained enough 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 subgoals of G1.2.2 and relates the respective parts of the examples to each added goal.

3.5 Visualisation of Conflicts between Reviewer and Model Designer

The results of a finalised review session will be visualised by another form of annotation for the goal model. Goals will be annotated with two coloured bars expressing values of the review statements described in the last section using colour codes for providing a quick overview of the results to other reviewers or the original model designer.

Therefore, the annotation consists of a coloured ***D***-bar for review statements of cases (1) or (2), which were related to real world examples and the type of the ***D***ependency relation 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 are quite simple and easy to overlook:

- *Green* colour denotes that the reviewer agreed on the reviewed products;
- *Red* colour indicates that some problems occurred during review either 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 RWEF or new respectively refined goals;
- *White* shows that this product has not been reviewed at all respectively 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 comprehend the reviewer's statements and find the topics which need further consideration and discussion with the goal model designer.

Example: Leonard 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 Leonard's review. For example, the bars of goal G1.2.2 have been coloured green because of Leonard's agreement. The newly introduced sub-goals are marked blue to reflect their new introduction with new example fragments.

Leonard presents this display to James and they start further discussion on the production times report.

4. Conclusions

In this paper, we presented a novel approach to support reviewing of current-state models. We described how the review process is improved by making abstract concepts better understandable and assessable for the reviewer by providing him with appropriate real world example fragments. This was achieved by establishing extended traceability through the fine-grained interrelation of model components with parts of multimedially captured system usages of the existing system. We outlined a set of new tool supported features made available through extended traceability and demonstrated their usefulness for model reviews with a small case study. Management of the extended traceability is provided by a trace repository on which the described tools have been built.

For this work, we concentrated on the application of extended traceability to support review of current-state models. In [9] we described in detail how our approach is used for requirements elicitation and validation. However, further applications of our technique, which we will explore in future, are possible. For example, besides the interrelation of conceptual models with captured system usage of the existing systems the relation to captured visionary explorations of future systems is possible as well. Approaches like the one introduced by Nicola Millard in [14] in which she captures role-play and storyboards of stakeholders exploring future telecommunication systems can be extended with our technique applying the same principles as described in [9] and in this paper. Models for future systems could be elicited from the captured role-play and storyboards as well as their review and negotiation using our tool environment.

Another application for a reference base of extended traceability could be educational training to introduce stakeholders either into the basic functionality of the existing or future system or for teaching the modelling language in an example based manner.

Acknowledgements. This work is founded 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/). The authors are grateful to their students Sebstian Brandt, Sven Ewald, Markus Hoofe, and Michael Mattern for implementing major parts of the tool environment.

5. References

[1] 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.

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

[3] J. Blomberg, "Ethnographic Field Methods and their Relationshipto Design", in *Participatory Design: Principles and Practices.* D. Schuler, A. Namioka (eds.), Lawrence Earlbaum Associates Inc., Hillsdale, New Jersey, 1993, pp. 123-155.

[4] 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.

[5] 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.

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

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

[8] 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.

[9] P .Haumer, K. Pohl, and K. Weidenhaupt. "Requirements Elicitation and Validation with Real World Scenes", to appear in *IEEE Transactions on Software Engineering*, Special Issue on Scenario Management, 1998.

[10] 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.

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

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

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

[14] 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.

[15] K. Pohl, "PRO-ART: Enabling Requirements Pre-Traceability", *Proc. 2nd Int'l. Conf. on Requirements Eng (ICRE'96)*, Colorado-Springs, Colorado, USA, Apr. 1996, pp. 76-84.

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

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

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

[19] Rational Software Corporation, *Unified Modeling Language.* Available on the World Wide Web: http://www.rational.com, Rational, 2800 San Tomas Expressway, Santa Clara, CA 95051-0951, USA, Jan. 1997.

[20] 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.

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

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

[23] 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.

[24] 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.

[25] E. Yu, "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.