CREWS Report 98-16

IEEE Transactions on Software Engineering, Vol. 24, No. 12, Special Issue on Scenario Management, December, 1998.

Requirements Elicitation and Validation with Real World Scenes

by

Peter Haumer, Klaus Pohl, Klaus Weidenhaupt

Lehrstuhl Informatik V, RWTH Aachen
Ahornstraße 55, 52056 Aachen, Germany
+49 (0)241 80 21 501
{haumer,pohl,weidenh}@i5.informatik.rwth-aachen.de

Requirements Elicitation and Validation with Real World Scenes

Peter Haumer, Klaus Pohl, Klaus Weidenhaupt

Lehrstuhl Informatik V
RWTH Aachen
Ahornstraße 55
52056 Aachen
Germany

{haumer,pohl,weidenh}@informatik.rwth-aachen.de

A requirements specification defines the requirements for the future system at a conceptual level (i.e. class or type level). In contrast, a scenario represents a concrete example of current or future system usage. In early RE phases, scenarios are used to support the definition of high level requirements (goals) to be achieved by the new system. In many cases, those goals can to a large degree be elicited by observing, documenting and analysing scenarios about current system usage, i.e. the new system must often fulfil many of the functional and non-functional goals of the existing system.

To support the elicitation and validation of the goals achieved by the existing system and to illustrate problems of the old system we propose to capture current system usage using rich media (e.g. video, speech, pictures etc.) and to interrelate those observations with the goal definitions. Thus, we particularly aim at making the abstraction process which leads to the definition of the conceptual models more transparent and traceable.

More precisely, we relate the parts of the observations which have caused the definition of a goal or against which a goal was validated with the corresponding goal. These interrelations provide the basis for

- explaining and illustrating a goal model to, e.g., untrained stakeholders and/or new team members, and thereby improving a common understanding of the goal model;
- detecting, analysing and resolving a different interpretation of the observations;
- comparing different observations using computed goal annotations;
- refining or detailing a goal model during later process phases.

Using the PRIME implementation framework, we have implemented the PRIME-CREWS environment which supports the interrelation of conceptual models and captured system usage observations. We report on our experiences with PRIME-CREWS gained in a first experimental case study.

Keywords: requirements management, scenario, scenario-based requirements engineering, requirements traceability, requirements elicitation, requirements validation, goal modelling, rich media, process-integrated environments, software development

Introduction

1.1 Definition of Current-State and Desired-State Models

Traditionally, the role of requirements engineering is to establish a complete, consistent, and unambiguous requirements specification which defines the requirements of the desired system at a conceptual level. In the late seventies and during the eighties a variety of methods for defining the data (e.g. ER modelling [6]), the behaviour (e.g. StateCharts [17]), or the functions of the system (Structured Analysis [12], [46]) had been proposed. In the late eighties and early nineties object oriented techniques (e.g. [4], [41]) appeared which proposed an integration of the three views. Today this stream resulted in the definition of the Unified Modelling Language [37].

As argued in many publications, it is essential to consider the history and functionality of the existing system when defining the requirements for the new system (e.g. McMenamin and Palmer [27]). There are two main reasons for this: (a) the new system has to provide to a large degree the functionality of the old system; (b) making the same error twice can be avoided, i.e. one can learn a lot from the success stories and pitfalls of the existing system. Other authors see current-state analysis as essential foundation for change integration, i.e. incrementally defining the new system based on a conceptualisation of parts of the existing system, like in Lundeberg's ISAC approach [25] or in the RE frameworks proposed by Potts [36] and Jarke and Pohl [22], [23]. Another popular area where the modelling of the current-state serves as a basis for defining the desired future state is business process reengineering, e.g. [42].

Thus, there exist two types of conceptual descriptions in a requirements engineering process. The current-state model, which (partially) defines the functionality and history of the existing system, and the desired-state model which defines the requirements for the future system. Jackson [19] calls the first type of model indicative properties whereas the second type of model is called optative properties. As depicted in Fig. 1, implementing a new system thus requires four major steps:

1. *Reverse analysis*: Defining a current-state model by abstracting from the reality is required, since in most cases no conceptual model of the actual system exists, or the model is not up to date;
2. *Change definition*: Integrating the change definition into the (partial) current-state model, thereby defining the model for the new system;
3. *Change implementation*: Designing, implementing, testing and installing the new system based on the desired-state model;
4. *Legacy integration*: Considering the existing context during the change implementation to empower reuse and to avoid conflicting system implementation.

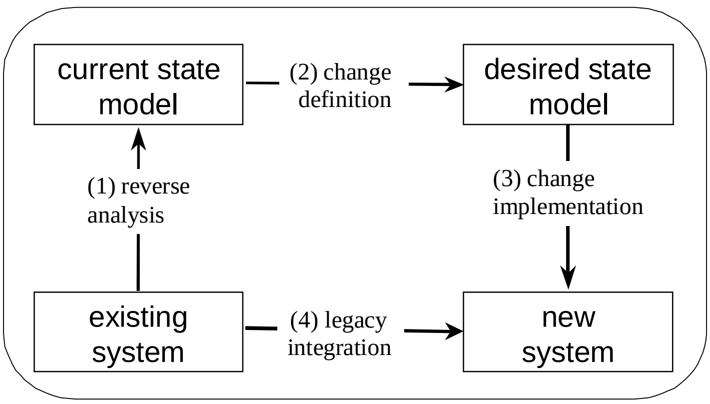

Fig. 1: Four Steps of a Typical Change Process.

The quality of the new defined (or updated) current-state and desired-state models evidently depends on the knowledge elicited from the stakeholders; i.e. it heavily depends on the successful stakeholder involvement in the requirements engineering process.

Scenarios are proposed as an ideal means to support the definition of the current-state model and to drive the change definition, i.e. to achieve better stakeholder involvement. In contrast to conceptual models, scenarios describe concrete examples of current and future system usage. As indicated, e.g., in a survey of industrial projects using scenarios [44] the use of scenarios, in addition to conceptual models, improves the quality of the requirements engineering process. A comprehensive comparison of existing scenario-based techniques has been developed in the ESPRIT Reactive Research Project CREWS[1] [40]. The most popular form of scenarios are use cases as proposed by Jacobson's OOSE [21] and their various extensions, e.g. by Regnell [38],[39] or Cockburn [8].

Complementary experiences in e.g., participatory design (e.g.,[3], [5], [43]) indicate that better stakeholder involvement can be achieved by using rich media (e.g. video, pictures, screen dumps, speech etc.) for recording and discussing current system usage. Amongst others, the use of rich media in requirements engineering processes leads to a better understanding of the usage domain, enforces focused observation of (temporally and/or spatially) distributed aspects, avoids presumptuous abstractions, enables repeatability of results and late reflections [5].

1.2 Overview of the paper

In this paper, we present an approach which bridges the gap between concrete examples of current system usage (scenarios) at the instance level and the conceptual current-state models at the type level. We argue to capture current system usage using rich media and to interrelate those observations with the current-state model. More precisely, we propose to relate the parts of the observations which have caused the definition of a concept of the current-state model or against which a concept was validated with the corresponding concepts. Thereby the conceptual current-state model and the recorded observations are interrelated. Our approach particularly aims at making the abstraction process which leads to the definition of the current-state models more transparent and traceable.

The interrelations between the components of the current-state model and the corresponding parts of the observations provide the basis for

- explaining and illustrating a conceptual model to, e.g., untrained stakeholders or new team members, and thereby improving a common understanding of the model;
- detecting, analysing and resolving different interpretation of the observations;
- comparing different observations using computed annotations based on the interrelations;
- refining or detailing a conceptual model during later process phases.

The rest of the paper is structured as follows. In Sect. 2 we provide an overview of our approach. For early requirements engineering phases we argue to first focus on the elicitation and validation of the *goals* achieved by the current system. In addition, we outline the main benefits gained from interrelating goals and recorded observations. In Sect. 3 we define typed dependency links. We illustrate the use of the dependency links to relate a goal to the parts of the captured observations which have caused its definition and/or against which the goal was validated. In Sect. 4 we propose navigation and visualisation techniques which are based on the fine-grained interrelations between the goal model and the captured observations. These techniques are used for explaining the goal model, for comparing different observations and for supporting the integration of different stakeholder views.

We have implemented our approach in the PRIME-CREWS environment (Sect. 5). PRIME-CREWS has been used in a trial application. In Sect. 6 we sketch some examples from the trial application, summarise our experiences and characterise typical situations in which the proposed approach can and has been successfully applied. Finally, we compare our approach with existing work (Sect. 7) and provide an outlook on future work (Sect. 8).

[1] Cooperative Requirements Engineering With Scenarios.

2 Overall Approach

2.1 Interrelating Captured Observations and Current-State Models

Fig. 2 depicts our overall approach. We propose to capture observations of current system usage using rich media. We call a captured observation *Real World Scene*. The material gathered during an observation may contain information about many system usages. We therefore propose to pre-structure this material into what we call a *Real World Example* (RWE). An RWE is a collection of material that represents *one* 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 are elicited from the RWEs. On the other hand, the current-state models can be validated against the RWEs. In both cases, we propose to interrelate the parts of the observations with the component of the conceptual current-state model which was elicited from the fragment and/or validated against the fragment (see Fig. 2). The requirements engineer has therefore to select the corresponding fragment of the RWE and to indicate the type of the interrelation to be created between the fragment and the component of the current-state model. An interrelated part of a real world example is called *Real World Example Fragment* (RWEF).

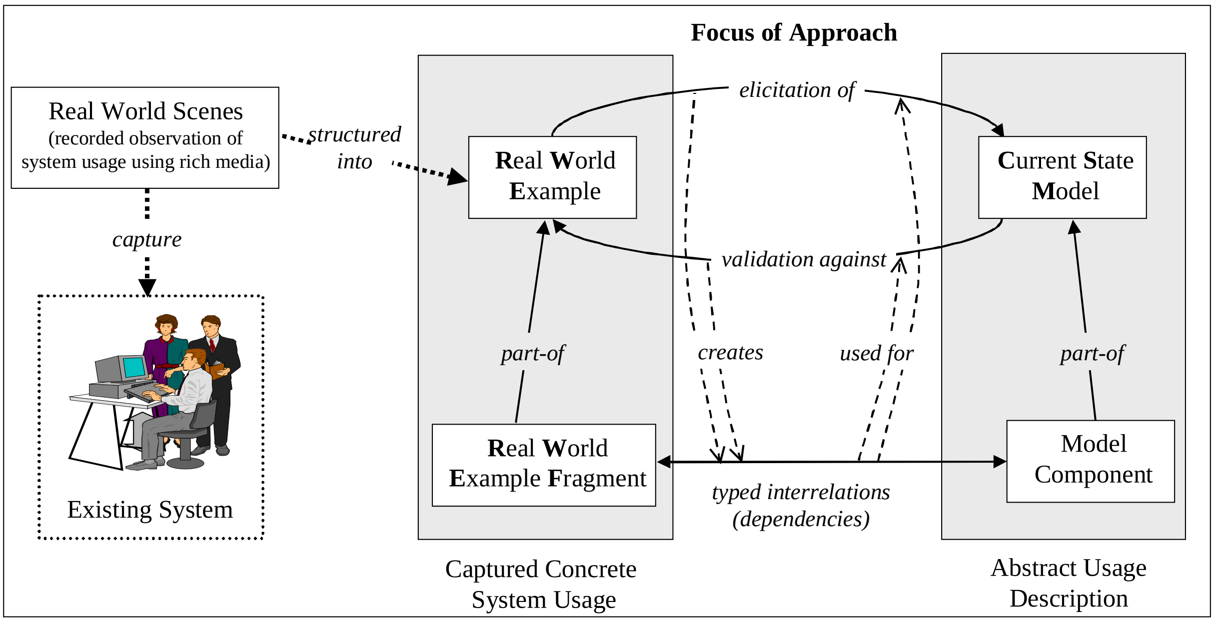

Fig. 2: Interrelating Real World Example Fragments with Model Components.

The interrelation takes place at a *fine-grained* level, since arbitrary fragments of the recorded observations (e.g. a cut-out video clip or even a part of one picture as extreme) can be linked to any element of a conceptual model – in contrast to relating the whole observation to the current-state model. The types of interrelations to be created between the RWEFs and the components of the current-state models depend on the modelling primitives.

Note that we do not propose a new conceptual modelling technique, but argue to embed an appropriate existing technique in our overall approach. Likewise, we neither provide any guidelines for capturing real world scenes nor for the pre-structuring of real world scenes into real world examples. However, the approach described in the following can be embedded in the guidelines proposed, e.g. by McGraw and Harbinson [26] for capturing and evaluating real world observations.

Obviously, our approach is not equally well applicable for any kind of project. It is well suited for projects in which the functionality of the old system could be observed and in which this functionality has to a large degree be provided by the new system (even if the system implementation changes significantly due to, e.g., technological progress) and/or in which observed shortcomings provide the basis to derive new goals for the new systems. Innovative projects where there is no precursor system will benefit less from our approach.

2.2 Benefits

When interrelating the model components with the RWEFs, a fine-grained formal structure is gradually imposed on the initially completely unstructured observations. Moreover this structure is interrelated with the model components defined in the conceptual model. Thus, one can selectively access the RWEFs related to a model component. Vice versa, the components elicited from or validated against an RWEF can be retrieved. Thus bi-directional selective access between the observations and the conceptual models is established.

In this paper we focus on four main advantages gained by the fine-grained interrelations:

1. *Explanation of the current-state model*. Current-state models are typically being built by one (or a group of) stakeholder(s). The abstraction of reality is typically performed in the minds of the stakeholder, i.e. for a person not involved in the model definition, the resulting model and the abstractions made are hard to understand. The typed interrelations empower to explain the model component by retrieving the related RWEFs, i.e. the abstraction process made during the model definition is (partially) traceable. Thereby training of people joining the project is eased and a better model understanding during the whole system development lifecycle is established.
2. *Comparison of different real world examples*. Stakeholders may perform the same task in different ways. Similar, the implementation of the same requirement can differ, i.e. implementing an essential feature can result in different incarnations [27]. Trying to understand the essence behind different incarnations just based on observations is sometimes very difficult. The comparison of different observations can be supported by using a more abstract conceptual description of the observations. The typed interrelations between the RWEFs and the model components support such a comparison;
3. *Definition and comparison of multiple viewpoints*. The definition of a current-state model based on a (set of) real world scenes is always subjective, i.e. depends on the perception of the people defining the model. Since the observations are persistently recorded, each stakeholder can define his viewpoint in a conceptual model according to his perception based on the same set of observations. Viewpoint resolution can be supported based on the type of interrelations between the model components and RWEFs. Different conceptualisation of identical and/or overlapping RWEFs and similar conceptualisation based on non-overlapping RWEFs can be highlighted. In addition, the typed interrelations can be used to support the mediation and negotiation of the stakeholders by grounding discussions on (parts of) real world observations;
4. *Reviewing of conceptual models*. Structured review of current-state models leads to higher model quality. The review can make use of the RWEFs to better understand the abstractions made and thus to better justify and comment on the model (components) under review. The typed interrelations are a prerequisite for selectively accessing the fragments for a model component.

2.3 Goal Modelling

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*, 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. Eric Yu has nicely summarised the various applications of goals in requirements engineering [48]. As he indicates, goals are used:

- *as central means for requirements elicitation and elaboration*, thereby encouraging the stakeholders to ask "why", "how" and "how else" questions (e.g., [2], [11]);
- *for relating the system to organisational and business context* (e.g. [47], [20], [33]);
- *to clarify requirements and stakeholder objectives* without the need to go too much into detail (e.g., [7], [24])

- *to deal with conflicts* (e.g., [28],[7], [11]). Goals can be used as interconnecting mediation points supporting a focus on common objectives first, before going into the details of resolving the conflicts.

The fact that goal modelling is widely used for a large variety of purposes indicates that goals are a fundamental concept within an intentional ontology of early requirements engineering phases [48].

We have thus 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 for goal models, namely the organisation of goals in hierarchical AND/OR reduction/refinement graphs.[2] Goal models are represented as directed acyclic graphs (DAG). Each goal is depicted as a node of the DAG. An AND-reduction of a goal g into two or more sub-goals $g_1,...,g_n$ is represented by angular lines between g and all sub-goals (see Fig. 3, left). The meaning of an AND-reduction is that for fulfilling g, all sub-goals $g_1,...,g_n$ must be fulfilled. OR-refinements are expressed by direct lines between a super-goal g and at least two sub-goals $g_1,...,g_n$ (see Fig. 3, right). In this case, g is fulfilled if at least one goal g_i among $g_1,...,g_n$ is fulfilled. Each sub-goal can itself be further reduced/refined.

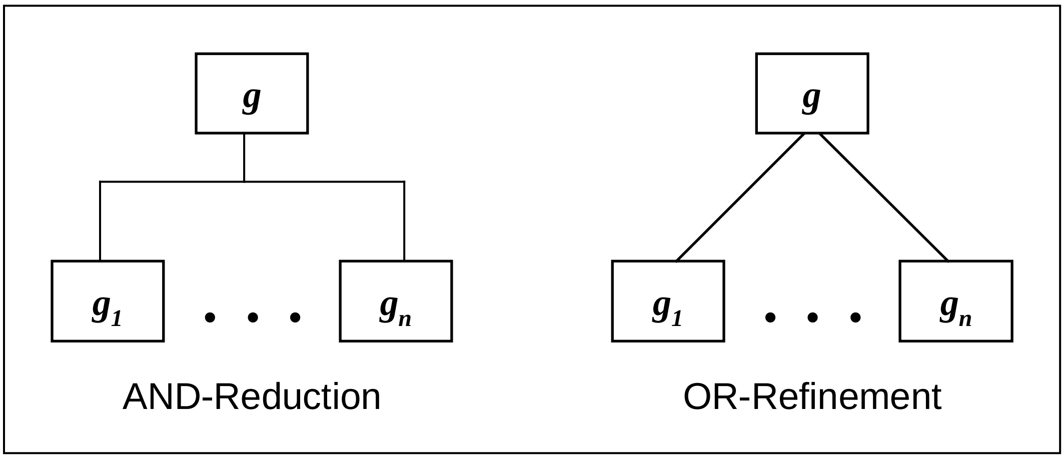

Fig. 3: Building Blocks of Goal Models.

Using goals as central concepts we take advantage of the fact that goal models are more abstract (expressing the why) than data, behaviour, and functional descriptions. Concentrating on goals facilitates the detection of commonalties 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. 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.

However, our approach does not force a strict top-down conceptualisation process. It equally supports a bottom-up conceptualisation in which first observed low-level functional goals are defined and then abstracted into more high-level functional or even business goals. Thus, obviously also a mixture of both, i.e. middle-out conceptualisation, can be supported. As we rely on observable phenomena, the initial abstraction level is normally restricted by what can be captured in an RWE. We would thus typically not start with a very high-level goal like “increase profit by 10 per cent”.

[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 Defining a Current-State Goal Model based on Real World Examples

In this section, we elaborate on the fundamental relationships that can exist between an RWEF and the components of a goal model, and describe a set of method fragments which establish these relationships.

When developing a goal model based on (a set of) RWEs, *elicitation* and *validation* can be distinguished as two conceptually different objectives. The focus of elicitation lies on building up the goal model according to the knowledge gained by analysing the observations[3]. The objective of validation is to collect evidence from the RWEs for the individual goals of an elicited goal model. In a typical analysis session however, elicitation and validation are often heavily intertwined. Whenever the analyst encounters problems in validating a goal model against the RWEs, this leads to the elicitation of new goals, which either have to be attained in addition to existing goals, or may represent alternative goals for achieving a super-goal.

Based on the assumption that the observation material belonging to one RWE has been arranged according to some logical criteria, e.g. to the temporal sequence of events, we recommend to analyse each RWE from the beginning to the end.

If the requirements engineer identifies a goal to be achieved in a part of the RWE and this goal is defined in the current goal model, he extracts this part as an RWEF and interrelates it with the goal using a typed dependency link (Sect. 3.1). If the goal is not yet defined in the hierarchy, he has *elicited* a new goal, which has to be added to the goal model and linked with the RWEF (Sect. 3.2). To support the explanation of the goal model, we propose to characterise certain RWEFs as reference examples for attaining or failing a goal and documenting this by defining a specific dependency link between the goal and the RWEF (Sect. 3.3).

3.1 Validation of Goals

When the analyst identifies a correspondence between a part *r* of an RWE with a goal *g*, he can interpret *r* to have basically one of two possible relationships to *g*:

1. *r* is interpreted by the analyst as an example of how the current system attains *g*. Then *r* is extracted as an RWEF and linked to *g* via a dependency link of type `Attains`. We call *r* an *attainment evidence* for *g*.
2. *r* is interpreted by the analyst as an example that shows a bad, unwanted or a to be avoided attempt of attaining a goal *g* which results in its failure. Then *r* is extracted as an RWEF and linked to *g* via a dependency link of type `Fails`. In this case *r* is called a *failure evidence* for *g*.

The relationships mentioned above only make sense when the analyst can be sure that the goal *g* is actually being tackled in *r*. When *g* is not tackled in *r* at all or the attainment/failure of *g* is not observable, then there is no statement on the relationship between *r* and *g* possible. In this case, *r* and *g* remain unrelated.

3.2 Elicitation of Goals

The analysis of an RWE may also lead to the elicitation of a new goal. Whenever the requirements engineer interprets a part of an RWE as an attempt to achieve a goal which is not yet present in the goal model, four situations can be distinguished as illustrated in Fig. 4.

[3] There may already exist a conceptual (goal) model of the system being analysed. Although these models may provide valuable information on which a refined current-state model can be built, they should be treated with care since they often do not reflect the current system properly. Possible reasons may be that the system has gone through slow evolution (i.e. the old models are out of date), that the system had never been fully implemented (i.e. the old models are too wide) or that the system is used in a different way than originally intended.

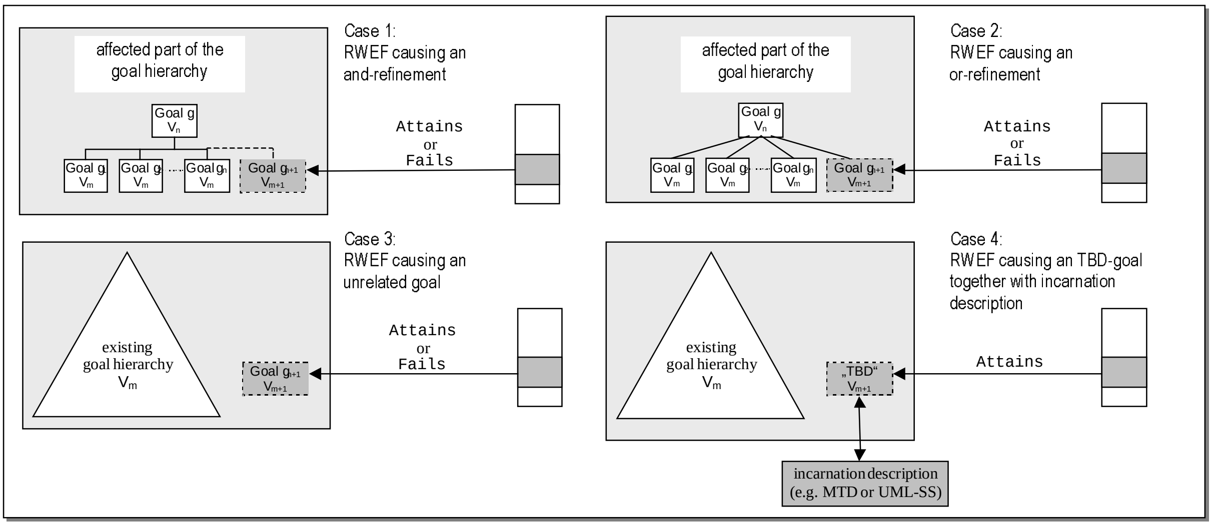

Fig. 4: Elicitation of a New Goal from a RWEF.

1. The stakeholder interprets the new goal as necessary for the attainment of a super-goal *g* in addition to currently defined goals $g_1,...,g_n$. In this case, the new goal is introduced as a sub-goal of *g* and related to $g_1,...,g_n$ via an and-refinement link (case 1 of Fig. 4).
2. The stakeholder interprets the new goal as a further alternative to achieve a certain super goal *g* which is not yet expressed by the existing alternatives $g_1,...,g_n$. Then the new goal is added as a sub-goal of *g* and related to $g_1,...,g_n$ via an or-refinement link (case 2 of Fig. 4).
3. The stakeholder cannot determine the relationship of the new goal to other goals within the current goal hierarchy. Then the new goal is added to the existing goal hierarchy but without a relationship to other goals (case 3 of Fig. 4).
4. The stakeholder is unsure to which goal the RWEF contributes. Then he introduces a "to be determined"-goal (TBD-goal) and relates it to the RWEF. In addition, he may enrich the TBD-goal with an appropriate incarnation description (e.g. MTD, UML Static Structure Diagram, ...) as depicted in case 4 of Fig. 4. When the TBD-goal is elaborated in a later session, the interrelated incarnation description as well as the linked RWEF can be used as a source of additional information.

In all cases, the corresponding part of the RWE is extracted as RWEF and linked to the new goal either via an `Attains` link if the goal was in fact attained or via a `Fails` link otherwise[4].

3.3 Positive and Negative Examples

Once a larger set of RWEs has been matched against a goal model (possibly by different stakeholders), there may exist several attainment evidences for a certain goal which might indicate different ways of attaining *g*. However, by saying that *r* is an attainment evidence for *g* we do not make any statement on the way *how g* was achieved in *r*, i.e. whether it matches the expectations of the analyst or not.

For distinguishing the subset of RWEFs $\{r_{i1},...,r_{im}\}$ that show the most preferred way of attaining *g* among a set of all attainment evidences, we introduce the dependency type `Positive` between all $r_i \in \{r_{i1},...,r_{im}\}$ and *g*. We then call r_i a *positive evidence* of *g*.

Similarly, we may want to state that a set of failure evidences $\{r_{j1},...,r_{jn}\}$ shows the most illustrative ways of failing to achieve *g*. This is expressed by a dependency link of type `Negative` between $r_j \in \{r_{j1},...,r_{jn}\}$ and *g*. In this case we call r_j a *negative evidence* of *g*.

[4] To be able to reproduce which goals and links are added or altered in this (and not an earlier or subsequent) analysis session, all additional goals and modifications are marked with a new version number, e.g. in Fig. 4 the elements of the initial goal model have the version number V_m, whereas the added goal and dependency links have the version number V_{m+1}.

One could argue to assess each RWEF on a continuous scale from, e.g., 0 to 1 where 0 means the worst way of failing *g*, whereas 1 means the ideal way of attaining *g*. We believe however, that it is extremely difficult to exactly quantify the adherence of an RWEF to the expectations of the analyst and therefore rely on the simple qualitative assessment described above. The main purpose of the positive and negative dependency types is to mark certain evidences as *reference examples* of how to attain or fail a goal. This greatly improves the understandability of individual goals in the goal model when, e.g., new team members have to be introduced into the domain and trained.

4 Typical Applications based on the Fine-Grained Interrelations

We now describe three applications that exploit the established relationships between RWEs and the goal model:

- Explanation support by navigating along the established trace relationships between RWEFs and elements of the goal models (Sect. 4.1)
- Review of the (elements of the) goal model and the relationships to the RWEFs (Sect. 4.2)
- Visualisation of the goal model according to the linked evidences (Sect. 4.3)

It is noteworthy that just the restriction on the small amount of link types (four) defined between the goal model and the RWEs enables these applications. For example, the visualisation of correspondences and differences of the real world examples and the stakeholders' viewpoints described below would not be possible if there would be a larger set of interrelations.

4.1 Explanation Support

Formally speaking, the interrelations described in Sect. 3 establish a set *D* of tuples (*r*, *lt*, *g*), where *r* denotes a RWEF, *lt* a link type, and *g* a goal. This imposes, on the one hand, an access structure upon the RWEs; i.e. a set of RWEFs 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 evidences, which are close to the perception of the involved stakeholders.

Queries on *D* providing access paths on both the goal model and the RWEs can be exploited in various explanation situations e.g. for explaining abstract goals (Sect. 4.1.1), for explaining the "why" and "what" behind real world examples or for retrieving correspondences between different examples (Sect. 4.1.2).

4.1.1 Explanation of Goals

Goals are mostly formulated in an abstract way, which makes it difficult for different stakeholders to share a common understanding of their meaning. By using the established traceability links, RWEFs of attaining and failing a given goal *g* can be retrieved from *D*. In particular, the use of `Positive` and `Negative` links allows the retrieval of illustrating *reference* examples of goal attainment and failure, respectively. Thereby, new team members and stakeholders, who are not familiar with the goal-modelling notation, can easily and rapidly be drawn into the project. Fig. 5 (left) illustrates the use of the traceability links for goal explanation.

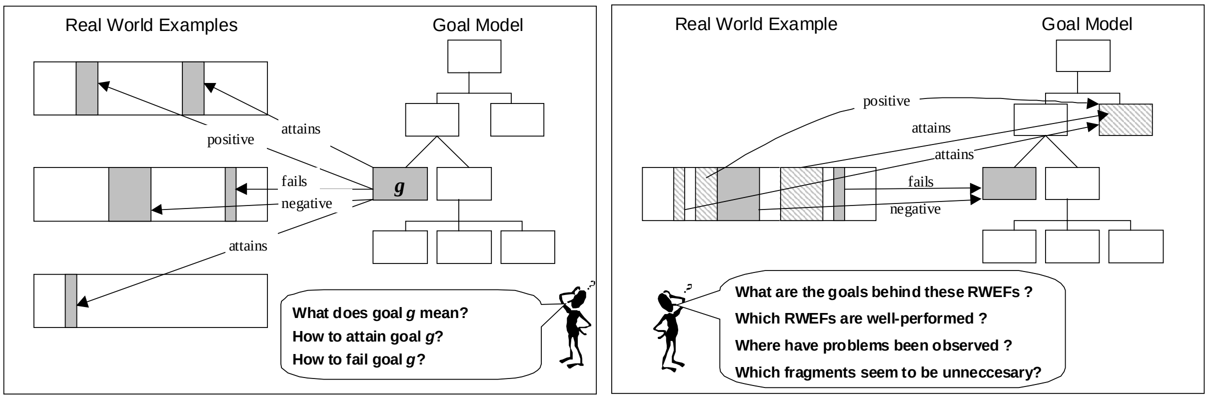

Fig. 5: Goal Explanation with Related RWEF and Explanation of Goals behind RWE.

4.1.2 Explanation of Real World Examples

Conversely, one could take an RWE *e* as starting point and ask for the "Why" and "What", i.e. the goals, behind certain RWEFs r_i of *e*. More precisely, as illustrated in Fig. 5 (right) one could ask for those RWEFs of the given example *e*

- which are well-performed, i.e. which fragments r_i of *e* are related in *D* to a goal via an `Attains` or even `Positive` link, and to which goals g_j?
- where problems have been revealed, i.e. which fragments r_i are related in *D* to a goal via a `Fails` or even `Negative` link; and to which goals g_k?
- which seem to be unnecessary since there is no obvious correspondence to a goal, i.e. for which fragments r_i does not exist a link to any goal in *D* ?

Another application of traceability information seeks at retrieving alternatives for a given RWEF r_1 through their relation to common goals in *D*. Fig. 6 (left) exemplifies a situation where we start with an RWEF r_1 showing a failure of goal g_1. In case (1) we use the traceability information to retrieve an RWEF r_2 of a different example showing the attainment of g_1. In case (2) we go a more indirect way and access a RWEF r_3 of another example which is linked to an alternative goal g_2 of g_1.

Besides the applications of the traceability information mentioned above, one could think of lots of further method fragments which use the goal tree as an access structure to the set of RWEs, e.g. to retrieve all necessary steps to achieve a goal, possibly distributed over more than one RWE. Therefore, one of the important requirements defined for the tool support of our approach (see Sect. 5) is that method knowledge supported by the tools should not be hard-coded, but be adaptable and extendable later from the outside.

4.2 Reviewing the Goal Model

The development of a goal model (and any conceptual model in general) always depends on the requirements engineer's *subjective* interpretation of the real world which he wants to capture in the goal model. For balancing and negotiating the goal model with the perceptions and opinions of other stakeholders, a review of the goal model is crucial. The established relationships to the RWEFs provide an excellent means for illustrating the decisions and considerations underlying the development of the goal model during the reviewing process. The reviewer is facilitated in understanding the original model builder's interpretations, e.g. the attainment of certain goals.

The interpretations and considerations of the original model builder must not necessarily be shared by the reviewer. This possibly leads to a different view, i.e. the existence of a goal as well as its relationship to a certain RWEF may be questioned by the reviewer. For expressing a match or a mismatch, we allow marking a goal or a relationship to an RWEF as either *agreed* or *disagreed* by the reviewer. In addition, the reviewer may enrich the goal model with *added* relationships between goals and RWEFs (as described in Sect. 3.1) or even elicits new goals (as described in Sect. 3.2). Thus, the interpretations and considerations underlying the changes made by reviewer are documented in the same way as those of the original model builder. On the one hand, this facilitates the detection of stakeholder differences. On the other hand, misinterpretations of the conceptual models and the RWEs by the reviewer, which possibly has considerably less knowledge than the original modeller, can easier be (re-)corrected.

In the following sections, we show how different visualisation techniques can be used to improve the reviewing process, e.g. by visualising stakeholder differences or displaying quality data of the existing interrelations.

4.3 Annotated Goal Tree

Besides explanation, the established relationships between a goal model and a set of RWEs can be used to annotate the goal model for different purposes. We first describe the situation where we want to compare two examples with respect to the goal model (Sect. 4.3.1). Next we consider the examination of two different stakeholder views concerning the relations of one example to the goal model (Sect. 4.3.2). We then generalise these applications to an annotation of the goal model which considers multiple examples and multiple stakeholder views (Sect. 4.3.3).

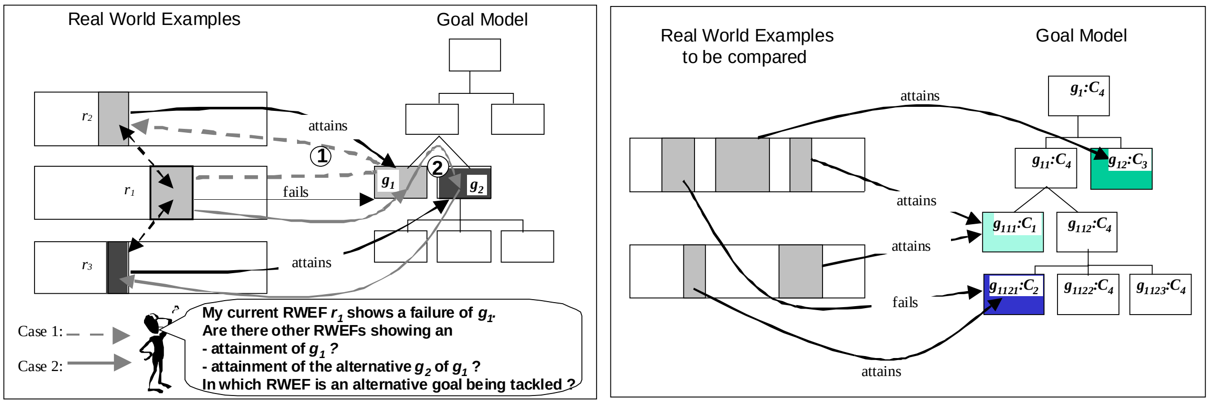

Fig. 6: Finding Alternative Ways to Tackle a Goal (left) and Visualising Differences in RWEs (right).

4.3.1 Comparing Real World Examples

Consider a situation in which the analyst wants to compare two RWEs in order to detect similarities or differences. In general, it is very difficult to compare the RWEs directly at the instance level since correspondences are often blurred by completely different incarnations. In this situation, the established relationships to the goal model provide means to visualise possible correspondences between both examples with respect to the goals they treat. For example, a goal which is tackled in both RWEs could be displayed in a different way than a goal which is only tackled by one RWE. In general, a goal can be related to a RWEF of each RWE using a relationship of the type (`Attains`, `Fails`, unrelated)[5]. There are thus, in principal, nine different dependency combinations (see Tab. 1). The correspondence of both examples with respect to a goal *g* can, however, be characterised using four categories:

C_1: *g* is related to both examples by the same link;

C_2: *g* is related to both examples by different links;

C_3: *g* is only related to one example, but not to the other;

C_4: *g* is related to none of the both examples;

It is sufficient to use four different colours $C_1 – C_4$ to visualise the correspondence of both examples with respect to a goal *g* (see Tab. 1).

Fig. 6 (right) shows a possible visualisation of the goal model where the four combination types are assigned to different goal node representations. It now becomes apparent that e.g., concerning goal

---

[5] For the sake of simplicity, we do not differentiate between attains and positive links, and fails and negative links, respectively.

g_{1121} the examples completely differ. By retrieving the corresponding RWEFs of both RWEs, the attainment of this goal in e_2 and the failure in e_1 can be directly compared and studied.[6]

e_1 / e_2	attains	fails	unrelated
attains	C1	C2	C3
fails	C2	C1	C3
unrelated	C3	C3	C4

Tab. 1: Possible Combinations of Relationships of two RWEs e_1 and e_2 to a Goal g.

4.3.2 Different Viewpoints of Stakeholders

As described in Sect. 4.2, during a review, a stakeholder imposes his own view on a goal model and the relationships between the goals and fragments of an RWE *e*. Since the statements of the reviewer can independently concern a goal itself as well as the relationships between a goal and a related RWEF, we attach two colours to each goal *g*: one for the goal itself, and one for its relationship to an RWEF *r* as indicated in Tab. 2 and Tab. 3.

Change during review	**Visualisation of goal g**
The reviewer has agreed on goal *g*.	C_1
The reviewer has disagreed on goal *g*.	C_2
The reviewer has added goal *g*.	C_3
The reviewer has made no statement on goal *g*.	C_4

Tab. 2: Differences between Stakeholder Views concerning the Goal g itself.

Change during review	**Visualisation of goal g**
The reviewer has agreed on a relationship of type *lt* between *g* and *r*.	C_1
The reviewer has disagreed on a relationship of type *lt* between *g* and *r*.	C_2
The reviewer has added a relationship of type *lt* between *g* and *r*.	C_3
The reviewer has made no statement on a relationship between *g* and *r*.	C_4

Tab. 3: Differences between Stakeholder Views concerning the Relationship between Goal g and RWEF r.

Fig. 7 shows a goal model visualising the review results. Note that each goal node contains two coloured bars: the left for indicating the differences between the reviewer and the model builder concerning the goal itself and the right for indicating differences in the relationships to the RWEF. Using the colours defined in Tab. 2 and Tab. 3 one can see that during the review the existence of goal g_{12} has been disagreed and that the reviewer has added goal g_{13}. Furthermore, it becomes visible that an evidence for goal g_{1121} as well as the goal itself has been confirmed by the reviewer, whereas there is an disagreement on an evidence for g_{1122}. In the case of goal g_{1123} the reviewer has introduced an additional relationship between the goal and an RWEF.

Thus, the annotated goal model provides an excellent basis for detecting conflicts in the perceptions of the different stakeholders. Following the traceability links, the corresponding RWEFs can be retrieved to support the negotiation and the resolution of these conflicts with concrete instances (examples).

[6] Note that the described comparison of two RWE relies on the assumption that the analyst has defined valid dependencies between the RWEFs and the goals. If the established dependencies do not correspond with the reality the computed correspondence between the two RWE could obviously be wrong. But even in this case, the computed correspondence helps in detecting and correcting "incorrect" dependencies.

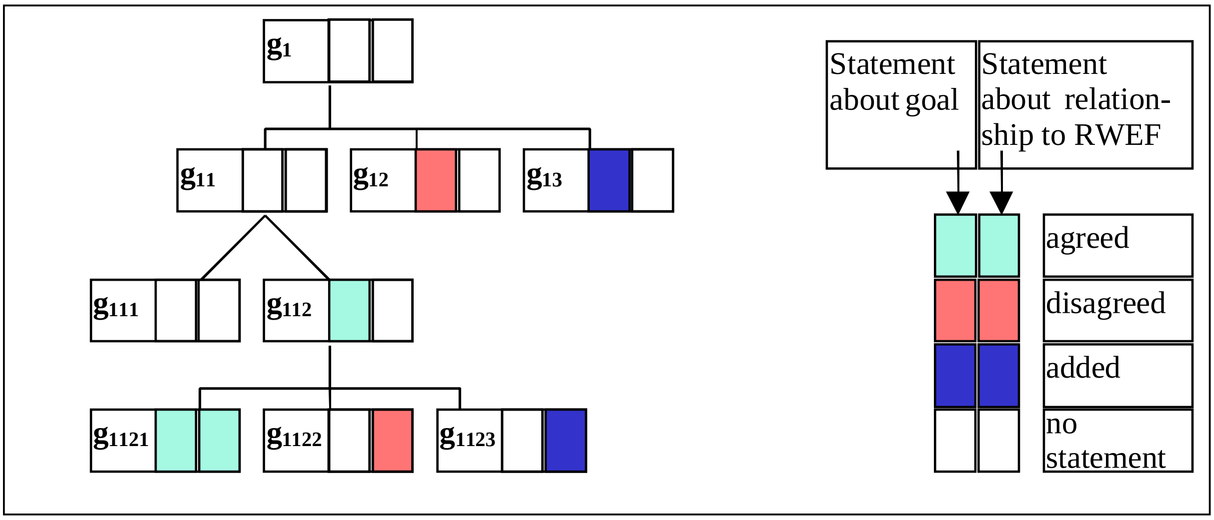

Fig. 7: Goal Tree Visualising the Differences between Stakeholder Views

4.3.3 Relevance and Success of Goals

While we have examined in the previous section how to visualise a goal model according to two different RWEs and according to two stakeholder views on the same RWE, we now present some simple, but effective ways to aggregate a set of evidences for a given goal and define some meaningful indices. These indices can be seen as a generalisation of the visualisations described in the previous sections[7].

Attainment and Failure Sets: As a preparation, we first define for a set of RWEs E and a goal g the

- attainment set $A_g = \{e \in E \,/\, e$ contains an RWEF r which is an attainment evidence for $g\}$;
- failure set $F_g = \{e \in E \,/\, e$ contains an RWEF r which is a failure evidence for $g\}$.

This means that A_g collects all RWEs in which a goal g has been tackled successfully, whereas F_g contains all RWEs where the analyst has observed a failure of g. Note that both sets must not necessarily be disjunctive since one example may contain different fragments where one fragment shows the attainment of g and the other one a failure of g.

Real World Examples Considered: During the relation process, a goal need not necessarily be matched against all RWEs. This occurs, e.g., if the goal belongs to a goal branch below an or-refinement in the goal hierarchy. Then RWEs dealing with one branch are reasonably not taken into account by the other branch(es). Another reason is that certain RWEs may only be used for an in-depth validation of a part of the goal model in a subsequent analysis session.

It is therefore necessary to annotate each goal with the set of RWEs E_g against which it was matched. The absolute number $|E_g|$ of RWEs considered for g is also important for assessing the statistical significance of the indices defined below.

Relevance Index: For assessing the *relevance* of a goal, we count the RWEs in which g is tackled (either successfully or not) in proportion to all RWEs against which it was matched. For this purpose, we define the *relevance index* of g, R_g, as

$$R_g = (|A_g \cup F_g|) \,/\, |E_g|$$

R_g is thus a value ranging from 0 to 1. $R_g = 0$ means that in none of the RWEs the attempt to attain g could be observed. Then, we can say that g is irrelevant with regard to our observations. There is no evidence that this goal is tackled at all in the current system. Conversely, the higher R_g is the more relevant g is.

[7] The set of proposed evidences could be extended by mechanism for combining beliefs in evidence and propagating such beliefs in an inference network as proposed by, e.g., Dempster-Shafer.

Success Index: The second index is concerned with how successfully a given goal g is tackled in current reality as observed in E_g. Here, we relate the attainment evidences of g to all evidences that make any statements about g (either attainment or failure) and define the *success index* S_g *as*

$$S_g = |A_g| / (|A_g| + |F_g|)$$

$S_g = 0$ means that there are only failure evidences of g, whereas $S_g = 1$ means that in each RWE dealing with g this goal is achieved. If $S_g = 0.5$ there are as many attainment evidences as failure evidences. If there are neither attainment nor failure evidences for g, S_g is undefined.

The above defined indices can now be used for displaying the goal tree in a variety of ways, e.g.:

Highlight all goals with a low relevance index (e.g. below 0.2): This query could be used to find those goals for which there are only a few evidences. This could be the starting point for a more detailed analysis of why the goal is not or only slightly tackled according to the RWEs captured. Reasons may be that the goal is not considered in the current system, that it is hard to observe the goal, or that the goal must not be achieved to fulfil a super-goal.

Highlight all goals with a low success index (e.g. below 0.2): Using this query, those goals can be found where the current system lacks support. By following the dependencies to the corresponding failure evidences, the reasons for the current problems can be accessed whenever needed for a more detailed analysis.

Display all super-goals of an AND-reduction, which have a lower success index than the average of their sub-goals: This query also considers the structure of the goal model. If such super-goals can be found, this could indicate that there is a further sub-goal missing in the current goal model which is not attained by the current system. Another reason could be, that one of the sub-goals with a low success index may be particularly critical.

Differences between stakeholder views: Given two stakeholder views (i.e. the ones of the original model builder and of the reviewer), the relevance and success indices can be defined for each view. Let S_g and R_g be the relevance and success indices of the view of the original model builder and S'_g and R'_g be the relevance and success indices of the view of the reviewer. Then we define the *relevance* and *success difference* as

$$\Delta R_g = R_g - R'_g$$

$$\Delta S_g = S_g - S'_g$$

$\Delta R_g > 0$ indicates that the reviewer considers less RWEFs as evidences for the goal g than the original model builder, while $\Delta R_g < 0$ means that the reviewer has found additional evidences. Similarly, $\Delta S_g > 0$ means that the reviewer disagrees with the attainment of g in some RWEFs and/or that he has found additional failure evidences.

The goals with a high mismatch in the stakeholders' views concerning relevance and success, i.e. where $|\Delta R_g|$ and $|\Delta S_g|$ are greater than a certain threshold, can now be depicted in a different colour and should be examined more closely. Again, the possibilities to access the related RWEFs facilitates an effective negotiation between the two stakeholders.

5 The PRIME-CREWS Modelling Environment

The creation, maintenance and use of the fine-grained interrelations between RWEFs and the components of the current-state model requires appropriate tool support. This has been realised in the PRIME-CREWS environment. We first sketch in Sect. 5.1 PRIME (*PRocess Integrated Modelling Environment*), a framework for process integrated modelling environments, which underlies PRIME-CREWS. PRIME offers generic mechanisms for supporting method adaptability as a basis for incorporating method guidance for establishing and applying the interrelations. We then outline in Sect. 5.2 the tools supporting the creation and visualisation of the fine-grained interrelations and describe the trace repository in which the information is being recorded. In Sect. 5.3, we briefly sketch the implementation of the PRIME-CREWS environment.

5.1 Method Adaptable Tools Based on the PRIME Framework

For creative modelling processes the method guidance embodied by a tool environment cannot be fully predefined [31]. This in particular holds for the field of scenario-based elicitation and validation, where knowledge about suitable method guidance is just emerging.

An important requirement is therefore, that the method knowledge underlying the tool support, e.g. the sequence of steps necessary for relating an RWEF to a goal with a certain dependency link type, is not hard-coded in the tools. Instead, it should be defined outside the tools for enabling an easy adaptation. Moreover, the method guidance offered by the tools clearly depends on the conceptual target models used and must be therefore extensible for multiple target formalisms.

Process-Centred Engineering Environments (PCEEs, [15]), which define method knowledge in explicit process models, in principle enable such an adaptation. They can be divided into three conceptually distinguishable domains [14],[31].

The *modelling domain* comprises of all activities for defining and maintaining process models using a formal language with an underlying operational semantics, which enables the mechanical interpretation of the models. The *enactment* domain encompasses what takes place in a PCEE to support (guide, enforce, control) process performance; this is essentially a mechanical interpretation of the process models by a so-called process engine. The *performance domain* is defined as the set of actual activities conducted by human agents and non-human agents (computer tools).

The interactions between these domains characterise the way in which model-based method support is provided. A process model is first instantiated within the modelling domain and passed to the enactment domain, i.e. process parameters like resources and time scheduling are bound to the project-specific values. Based on the interpretation of the instantiated model, the enactment domain supports, controls, and monitors the activities of the performance domain. The performance domain provides feedback information about the current process status to the enactment domain as a prerequisite for adjusting process model enactment to the actual process performance and enabling branches, backtracks, and loops in the process model enactment.

Thus, the fundamental mechanism provided by PCEEs is suitable for process execution based on explicit method definitions. In addition to the coarse-grained project management support provided by PCEEs, fine-grained support is required to enable method-driven developer guidance and trace capture. For example, a fine-grained method fragment guides the developer in performing reviewing activities on goal concepts selected in a goal editor by retrieving related RWEFs displayed in a multimedia tool and displaying information about the annotations back in the goal editor.

This would require that different tools (e.g. goal editor and multimedia editor) work systematically together according to the defined method fragment, and support the developer in annotating, e.g., the right RWEFs for the reviewed goal.

We have extended the PCEE approach resulting in a framework for process-integrated environments called PRIME (see [34] for a detailed description). In contrast to existing PCEEs, method fragments enacted in a PRIME-based environment, influence the behaviour of the tools, e.g. by restricting the selectable menu items and product parts in a tool according to the current process enactment state. In addition, the process integration provided by PRIME empowers the developer to initiate the enactment of predefined method fragments which, e.g. guide him in relating an RWEF to a corresponding goal of a goal model.

Besides elementary method steps, e.g. for creating a goal, we have defined altogether about 40 method fragments for the systematic interrelation of conceptual goal models with RWEs (Sect. 3), for various explanation situations (Sect. 4.1) as well as for reviewing activities (Sect. 4.2) and visualisation modes (Sect. 4.3). These method fragments are continuously adapted and augmented as method knowledge increases.

5.2 Establishing, Managing and Visualising Trace Information

As described in Sect. 4, the key prerequisite for comparing different RWEs, explaining the current-state goal model, comparing multiple stakeholder viewpoints and reviewing goal model components is to establish fine-grained trace relations between RWEFs and individual goals.

5.2.1 Trace Repository for Fine-grained Product and Dependency Models

A primary requirement is the persistent management of all products and traceability information in a logically centralised repository. Due to the amount of information captured and its heterogeneity, (e.g. goal models. real world examples) suitable structuring mechanisms are crucial for enabling an efficient and selective retrieval of the stored information. Moreover, the granularity of the underlying product models must be much finer than the document level since one wants, e.g., store a relationship between an individual goal and an RWEF, which are both parts of larger aggregates.

We have based our PRIME-CREWS trace repository on a product and process repository approach developed for the precursor environment PROART [30]. The repository adopts the four-level architecture of the IRDS standard [18] and structures the product models according to the dimensions *representation*, *specification*, and *agreement* [29] (for a detailed discussion of the repository see [31]). Orthogonal to these dimensions, dependency models are defined which provide dependency types for capturing trace relationships between arbitrary product parts.

For the PRIME-CREWS environment we have extended the repository by additional product models (e.g. goal model and multimedia models) as well as new dependency types which are specifically required for linking RWEFs with elements of goal models, namely the `Attains`, `Fails`, `Positive`, and `Negative` link types.

5.2.2 (Semi-)Automated and Adaptable Trace Capture

Due to the amount of traceability information produced, its recording must be automated as far as possible. Moreover, the kind of trace information captured should be adaptable to project and organisation specific needs [32], [13].

The basic approach is to derive as much trace information as possible out of the context of the current process performance. For this purpose, we augment the defined method fragments with additional method steps for storing traceability information automatically or to remind the developer to provide needed information. An example is the method fragment for validating a goal model component against RWEs. Whenever a goal and a corresponding RWEF have been selected by the user, a link is created between these objects.[8]

5.2.3 The Whiteboard Editor: Structuring Multimedia Artefacts

The structuring of RWEs as a basis for creating fine-grained traceability relationships to goal models is supported by a PRIME–based tool called PRIME-CREWS whiteboard editor. Its main functionality is to store, present, edit and reference all kinds of multimedia representations. The main usability concept behind the tool is the notion of having a whiteboard to collect, review and organise all incoming media after a site visit. Because RWEs can consist of many media objects e.g., a pre-cut video and extracted fragments plus several audio recordings and screen dumps, we organise them in folders each representing one RWE.

The editing facilities of the whiteboard editor support the extraction of arbitrary parts of the media objects, which can be linked to any model component managed by the PRIME-CREWS environment, to establish traceability information. This is accomplished for video and audio by selecting start and end points for extracting arbitrary time fragments and for images by cutting out rectangular parts. It is also possible to specify a characteristic freezer image for a video, which facilitates the identification of the video clip content while it is not running. Each video clip has individual playing and editing controls, which can be used for viewing and cutting of scenes. Several videos can be put side by side and viewed in parallel. Videos and images can be variably enlarged and shrunk.

During a review, each media object related to a goal being considered is presented in the whiteboard editor together with two button bars; one for indicating the dependency relationship stated by the original model builder (Attains, Fails, Positive, Negative), one for indicating the judgement of the reviewer (agreed, disagreed, added, no statement). This is illustrated in the example in Sect. 6.2.

[8] For details about the method-driven trace capture approach employed in PRIME-CREWS see [32], [13].

5.2.4 The Goal Editor: Construction and Visualisation of Annotated Goal Models

For constructing and visualising an annotated goal model according to the trace links to related RWEFs, we implemented a PRIME-based goal editor for the PRIME-CREWS environment.

The goal editor supports the construction of goal models using AND/OR reduction/refinement. For simplicity, goal models are presented as trees in the goal editor. This is no real restriction, assuming that the goal models are a-cyclic graphs and that cross connected goal nodes are made redundant in the tree hierarchy.

Besides goal model construction, the main functionality of the goal editor is the computation of the annotations based on the relationships to RWEFs and the suitable visualisations of computed annotations. This includes computing and visualising the differences of two RWE with respect to their relations to the goal model as described in Sect. 4.3.1, visualising two different stakeholder views as described in Sect. 4.3.2, and visualising the calculated relevance and success indices as described in Sect. 4.3.3. The latter two visualisation modes will be illustrated in more detail in the example presented in Sect. 6.2.

5.3 Implementation

The PRIME-CREWS environment has been implemented with C++ using the PRIME implementation framework [34]. It runs on both Solaris and Windows 95/NT. The goal models, the multimedia artefacts, their relations as well as any other products are persistently stored in a repository implemented on top of a relational database. Currently, we use the Sybase 11 RDBMS and Microsoft Access, but the data storage layer is implemented to support any database providing an ODBC interface. For the user interface, we used the ILOG Views toolkit, while the multimedia facilities within the whiteboard editor have been realised using Apple's QuickTime library.

6 Trial Application

We have validated our approach by a small trial application performed at a manufacturing company called ADITEC. This company produces machine gears for various kinds of industrial devices.

In Sect. 6.1 we outline the design of our study. In Sect. 6.2 we illustrate the main features of our approach by selected examples drawn from the case study. In Sect. 6.3 we summarise our experiences made.

6.1 Overview

The overall aim of the trial application was to improve the production management system employed at the ADITEC company. ADITEC uses a commercial Manufacturing Resource Planing system (MRP), a production scheduling system consisting of an Order Dispatch System (ODS) and individual Machine Terminals (MT) which are mounted at each Production Cell (PC). The MRP system is used to register and plan customer orders. The resulting production plans are then routed to the ODS where the foreman performs a detailed scheduling on a machine-grained level. The resulting orders are sent to the individual MTs. The workers use the MT to fetch their orders and report the production times.

Due to problems with inconsistent and erroneous reporting data, the ADITEC management initiated an analysis of the information flows between the MRP system and the production cells.

For obtaining a first rough picture of the ADITEC problem, we conducted a kick-off workshop with the ADITEC representatives employing traditional interview techniques. Based on this information, we identified those parts of the production management system that were most suited and interesting for our approach. These parts included hard-to-describe aspects like dispatching the machine-grained orders from the ODS and their reception at the individual MTs or the temporal sequence of reporting set-up, production, and measurement data by the workers at the PCs.

Altogether, we captured about 20 observations at seven different work places (the foreman's workplace at the ODS and six production cells). These observations were assembled to six real-world examples. The number of different observation settings (work places, workers) was thus of manageable size, yet sufficient to show significant differences.

The actual RE team comprised five members: the ADITEC technical manager (mechanical engineer) and the ADITEC foreman acting as domain experts and three members of the CREWS team acting as requirements engineers. In addition, the workers were indirectly involved by the observation of their work places.

The tools of the PRIME-CREWS environment were solely operated by ourselves, e.g. for creating elements of the goal model, for cutting out an RWEF, or for linking a goal with an RWEF. This was primarily due to the usability flaws which are unavoidable for an academic research prototype.

6.2 Application Examples

We illustrate five specific application examples drawn from the trial application: defining a current-state model based on RWEs (Sect. 6.2.1); explanation support (Sect. 6.2.2); reviewing a goal model (section 6.2.3); validating a goal model against a larger set of RWEs (Sect. 6.2.4); visualising an annotated goal model (Sect. 6.2.5).

For better understandability, we refer to the stakeholders occurring in the examples by their names: Peter, Klaus, and Patrick (the requirements engineers), Franz (the ADITEC manager), Fred (the foreman), and Walter and Guido (two of the workers). Please note that the verbal narration of the contents of the real world observations is sometimes inevitably lengthy and difficult to understand, whereas it would be instantly clear watching them on video; one reason why we propose the use of video and other rich media instead of textual representations.

For presentation purposes, we mainly consider lower level goals in the examples. We have chosen lower-level goals, since the related RWEFs are relatively small and their content is thus more easily and briefly to narrate. It would have also been possible to illustrate our approach using more higher level goals, e.g. the goal “optimise required set up time”. But in this case, the verbal narration of the contents of the related RWEFs would have been more lengthy.

6.2.1 Defining a Current-State Model Based on Real World Examples

For obtaining an initial goal model of the ADITEC production management system, the requirements engineering team first performed recordings of on-site observations of how an order is dispatched, processed and reported back. For this purpose, Klaus observed the foreman Fred dispatching some orders to the production cells, i.e. entering the production data and generating the orders in the MRP and ODS. Peter recorded the processing of the dispatched orders at the shopfloor by the worker Walter who obtained the orders through his machine terminal.

Creation of new goals based on observations: Returning from the observation visit, Peter used the PRIME-CREWS environment to systematically relate the RWE to an initial goal model. Based on the information gained from the RWE, Peter refined the goal G1.1 (“*Support production order transfer*”) with a sub-goal G1.1.1 (“*Order is scheduled*”). Guided by a method fragment of the PRIME-CREWS environment, Peter was prompted to enter the description for the goal. Then he created a representative cut-out fragment of the real world example in the whiteboard editor. In addition, he selected a characteristic freezer image of the RWEF that could be used for later reference of the scene.

Adding unclassified goals: In addition to the information Peter expected to observe, he detected additional activities: During production, Walter performed some unexpected interactions with the terminal. Rewinding the example several times enabled Peter to recognise that Walter used the terminal after measuring the produced parts. Since Peter could not interpret the goal behind the observed actions, he created a "to-be-determined"-goal (TBD) together with a short description ("Interaction with MT after measuring parts"). The goal was created using a variant of the method steps described above, which also linked the cut-out fragment to the newly created concept.

The results of Peter’s elicitation session were the initial goal model (Fig. 8, right window), the created RWEFs which had caused the creation of the goals and the typed dependency links introduced between the goals and the RWEFs.

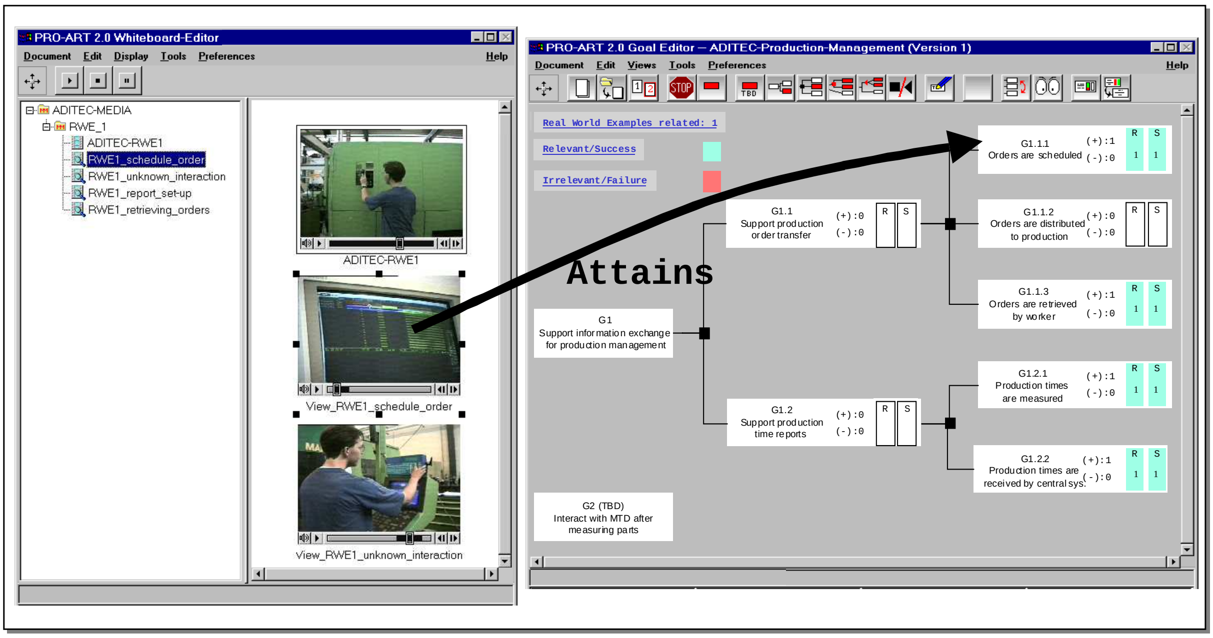

Fig. 8: Explanation of Goal G1.1.1 by Linked RWEF.

6.2.2 Explanation Support

The gradually established traceability information between the goal model and the RWEFs was used in the ADITEC study for a variety of explanation purposes. For example, the new RE team member Patrick who did not participate in the first site visits used the goal model together with the linked RWEFs to become familiar with the processes carried out at ADITEC.

Continuing with our example, the results of the pre-structured elicitation observations were discussed with the domain experts in the next group meeting.

Explaining goals with RWEFs: Peter used the PRIME-CREWS environment to explain the goal tree by retrieving the RWEFs linked to the goals. For example, the value (+):1 of goal G1.1.1 in the goal editor (Fig. 8, right window) indicated that it had been linked to an RWEF by an `Attains` link. To clarify with the domain experts whether the right abstractions had been made, Peter selected this goal in the goal editor. After activating a corresponding method fragment, the whiteboard editor displayed the RWEF *"View_RWE1_schedule_order"* used for eliciting the goals (Fig. 8, left window). On default, the RWEF is presented by its characteristic image. The play time interval of the fragment within the original RWE is visualised by the black bar in the play control of the fragment. By pressing the attached play button this interval was played so that the domain experts got an example for the goal as explanation.

Accessing the goals of an RWEF: Watching a real-world example, Franz wondered how the sequence showing Walter reading some information from the machine terminal was considered in the goal model. He used a method fragment for retrieving the goals related to the RWEF of which the displayed picture of the video is a part of. By following the typed interrelations the goal G1.1.3 ("*Orders are retrieved by worker*") was retrieved and highlighted in the goal editor. Franz agreed that in this sequence the machine terminal was used to display the order information properly.

Resolving the to-be-determined goal: Peter presented the TBD-goal elicited together with the linked RWEF to the domain experts. They realised that quality measurements, which play a crucial role in the production process, were not yet expressed properly in the goal tree. They resolved the TBD goal by changing it into a new goal G1.3 ("*Support quality data exchange*") and elicited further sub-goals from the video example. Thereby, they qualified the existing relationship to the RWEF showing the quality measurements as attainment of the new goals.

6.2.3 Reviewing a Goal Tree

The explanation of individual goals through linked RWEFs is also crucial when a goal model is reviewed by another requirements engineer or domain expert. In the ADITEC study, we let Klaus review the goal model (and the related RWEFs) built by Peter to impose his interpretations of the site visits. The goal editor supported Klaus in retrieving the RWEFs related to the goals. The whiteboard editor displayed, in addition to the retrieved RWEFs, the relationships and their types created by Peter between the RWEFs and the goals. Moreover, Klaus could state in the whiteboard editor if he agreed or disagreed with the interrelation.

Retrieval of linked RWEFs: The system supported Klaus in examining Peter's model by highlighting existing goal-RWEF relations in the goal editor while playing the RWEFs in the whiteboard editor. At some points, Klaus stopped playing the RWEFs and augmented the existing model. For supporting a comparison of Peter's and Klaus' views (see Sect. 6.2.5) the changes made by Klaus have been marked by the PRIME-CREWS environment.

Agreeing with a goal-RWEF relationship: Retrieving and observing the RWEF related to the goal G1.2.2 Klaus confirmed the attainment of the goal, i.e. he agreed with Peter that the RWEF showed Walter trying to report his production times. He expressed the agreement by selecting the agree option in the whiteboard editor.

Disagreeing with a goal-RWEF relationship: Retrieving and observing the RWEF related to the goal G1.3.1, Klaus interpreted the RWEF related to goal G1.3.1 as a failure. This was in contrast to Peter. Klaus therefore selected the disagree option in the whiteboard editor.

Eliciting new goals and adding relationships: In addition, Klaus observed that different production times were reported by the worker with varying success. Therefore, he refined goal G1.2.1 ("*Production times are measured*") into sub-goals describing that set-up times as well as the beginning and the end time of the production have to be reported. He linked the goals to the corresponding RWEFs and selected the *added* option in the whiteboard editor.

6.2.4 Validation against Multiple Real World Examples

During group meetings and reviews, conflicts and poor understanding of certain system aspects became evident which raised the need to analyse these aspects more deeply. In the ADITEC application, the discussion with the domain experts and the review by Klaus elicited some open issues about the usage of machine order data and how quality measurements are performed. To clarify the actual situation at the production cells, the analysis team scheduled another site visit. This time the prime objective was to consolidate and eventually refine the current understanding by a larger set of evidences. Therefore, the RE team performed detailed observations at the production cells involving several different workers at several different machines. As a result, the goal model developed so far was validated against five additional RWEs.

Relating a goal to an attainment evidence: The first RWE showed worker Guido stepping towards his machine terminal calling up the next machine order and browsing through the data displayed. Peter interpreted this scene as an attainment of goal G1.3. Therefore, he created the fragment and related it to the goal concept via an `Attains` link.

Relating a goal to failure evidence: Observing Guido's terminal operations, Peter discovered that although Guido was able to retrieve the production data, he did not manage to report the production times he needed for the lots produced. During production, the terminal refused to accept his production times. Guido did not seem to comprehend why reporting the times was impossible and gave up. The reason was that Guido had forgotten to report the beginning of the set-up. Peter therefore linked goal G2.2 to a combined set of fragments via a `Fails` link.

Distinguishing positive and negative examples for goals: After having analysed all RWEs, Peter distinguished those fragments, which showed the preferred or even ideal way of attaining a certain goal, e.g. the goal G1.3. He was supported by the PRIME-CREWS environment by a method fragment which performed a query on the stored relationship between the goal and the RWEFs and displayed all fragments related to this goal via an `Attains` link in the whiteboard editor. The RWEFs were displayed with their characteristic freezer image, so that Peter could easily identify the right fragment.

Peter decided that the observation of worker Guido retrieving his order data was the best evidence for goal G1.3. He thus linked the RWEF by a `Positive` link to goal G1.3. Using a similar method fragment, Peter also marked certain failure evidences as `Negative` examples.

6.2.5 Visualising an Annotated Goal Tree

During the ADITEC study, the elements of the goal model were elicited, validated, and refined in several sessions. Consequently, most goals were annotated by multiple RWEFs. Moreover, different stakeholder views were imposed on the goal model and its relationships to the RWEFs (in this case Peter's and Klaus' views). As the number of goals, RWEFs and links between them grew, it became increasingly hard to keep an overview and to find those goals, which were not properly treated by the current system or which were regarded differently by different stakeholders. In this situation, the visualisation possibilities offered by the PRIME-CREWS environment provided an excellent orientation support for quickly finding critical goals and related RWEFs.

Visualising the success and relevance of goal attainment: For demonstrating the most critical parts of the current system to the ADITEC management, Peter used the goal editor to display the annotated goal model for all six RWEs analysed. His aim was to focus on those goals which were only partially (low relevance index) or unsuccessfully (low success index) treated in the six RWEs. For each goal, the absolute number of attainment and failure evidences was displayed. For each goal, the R-bar and the S-bar indicate the relevance and success index, respectively. The larger the green share of each bar is, the higher is the relevance and success index (Fig. 9, right window).

For instance, goal G1.1.3 had been tackled in all six RWEs, which resulted to a relevance index of 1. The success index calculated to 2/3 for goal G1.1.3, because four times an attainment had been observed and twice a failure (4 / (4 + 2)). These two values were interpreted in the way that the observations covered this goal to 100% and that it worked quite successfully in the real world (in 2/3 of all cases).

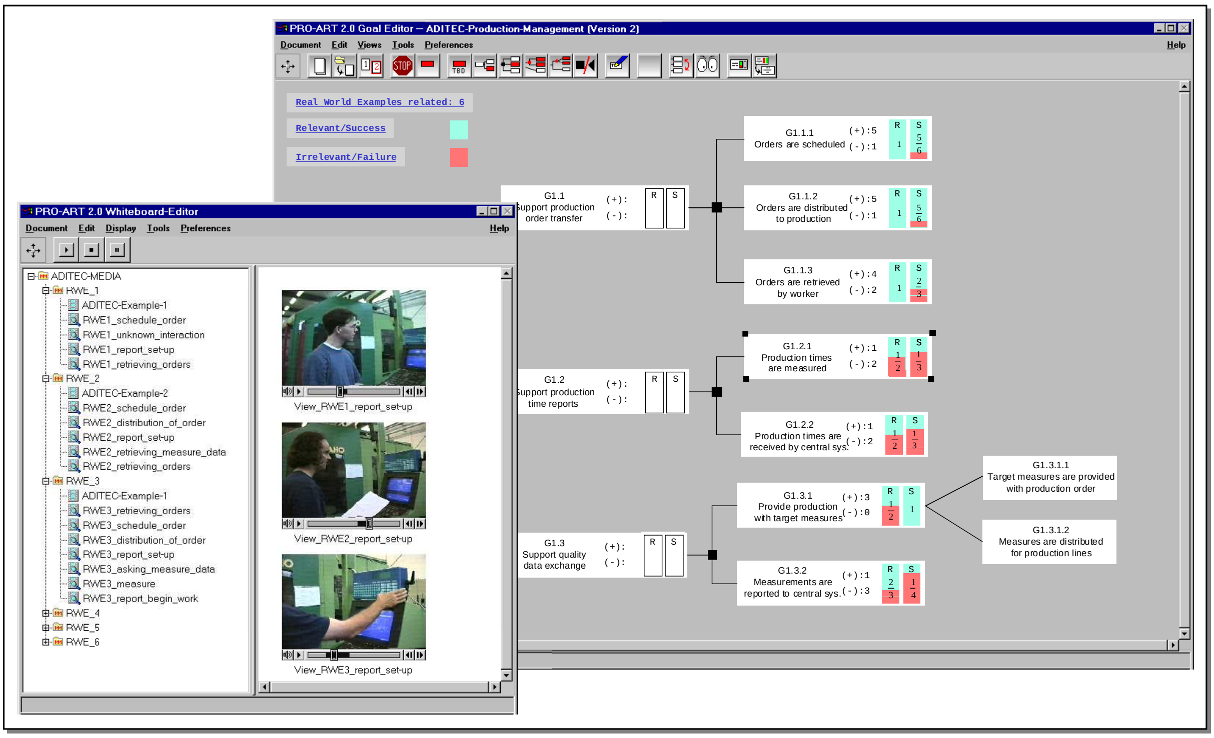

Fig. 9: Visualising Success and Relevance Indices

On the other hand, the values for the goals dealing with reporting data using the machine terminals (G1.2.1, G1.2.2 and G1.3.2) indicated several problems. For example: the workers had only been observed to measure production times (G1.2.1) in three out of six cases and the success index for this activity was even lower: 1/3. For illustrating the problems with goal G1.2.1, Peter selected the goal and retrieved the related RWEFs, which were displayed in the whiteboard editor (Fig. 9, left window).

Visualising conflicts in interpretation: After Klaus had finished his review session (see Sect. 6.2.3), Peter and Klaus used the goal editor to display differences in their interpretations. They invoked a method fragment to display the review results of Klaus and the original goal model of Peter. The method fragments marks variations in the two models. More precisely, the G-bar of each goal node indicates the differences in the goal structure and the D-bar indicates differences in the dependencies defined between the goals and the RWEFs.

As shown in Fig. 10 (right part), the green (light grey) G-bars of goals G1.1.1 to G1.1.3 expressed that Klaus had confirmed the correctness of these goals. For goal G1.2.1, Klaus had added a new link to an RWEF as indicated by the blue (dark grey) D-bar. To get a more detailed picture of Klaus' interpretations, Peter invoked the whiteboard editor, which automatically displayed the RWEFs related to G1.2.1 (Fig. 10, left part). A specific mode was used, where each related RWEF is annotated by its link type to the goal (*Attains*, *Fails*, *Pos./Neg.* buttons in the upper line) and by the interpretation of the reviewer (*Agree*, *Disagree*, *No Statement*, *Added* buttons in the lower line). Here, Peter recognised that Klaus had agreed with his interpretation of *View_RWE1_report_setup* and *View_RWE2_report_setup* as failure evidences, whereas Klaus had identified a third RWEF (*View_RWE3_report_setup*) as an additional attainment evidence.

Finally, the visualisation also showed that Klaus had disapproved the refinement of goal 1.3.1 into the sub-goals G1.3.1.1 and G1.3.1.2. As indicated by the red (grey) D- and G-bars, he had disagreed with the two sub-goals and the dependencies between the sub-goals and the RWEFs.

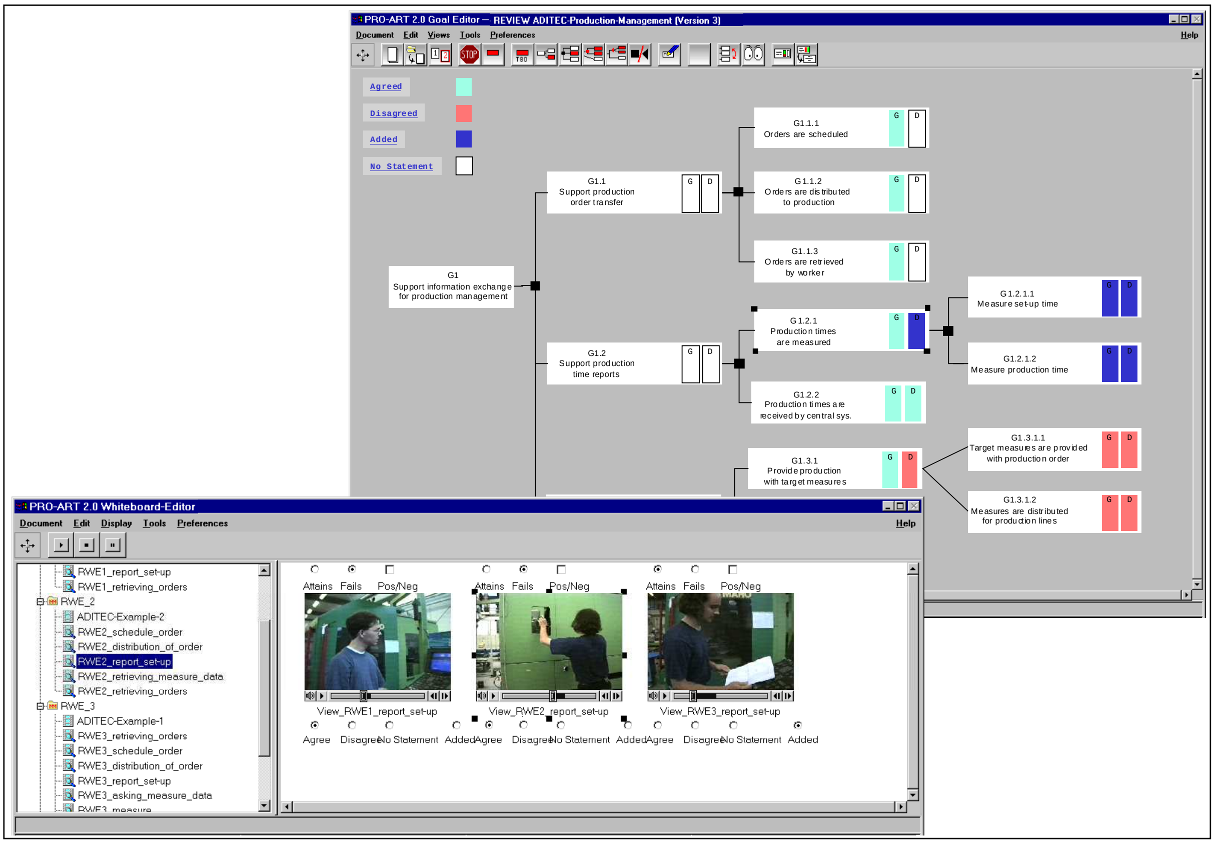

Fig. 10: Visualising Different Stakeholder Views

6.3 Lessons Learnt

The ADITEC 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. Nevertheless, some tentative conclusions can be drawn from this experience.

The ADITEC domain experts found that the fine-grained relations of the goals to the RWEFs significantly facilitated the understandability of the goal models and led to an improved shared understanding of temporally and spatially distributed observations made by different team members. The access to the related RWEFs helped to make the abstraction processes of the requirements engineers more transparent and to match them against the implicit knowledge of the domain experts. From our (i.e. the requirements engineers') point of view, the detection of conflicts in the assessment of certain goals by the visualisation possibilities and the retrieval of related RWEFs was most advantageous. It enabled an efficient negotiation process, which finally resulted in a common understanding of the concepts of the goal model. Moreover, the introduction of new team members with the problem domain and the model definition was significantly facilitated by the captured RWEs and their fine-grained interrelations with the goals defined in the conceptual goal model. In our case, Patrick was able to understand the problems by retrieving and watching the captured RWEFs; a site visit was not required.

On the other hand, our approach clearly adds to the workload of the stakeholders who are in charge of eliciting and validating the requirements of a new system. In the extracts of the ADITEC study presented in the preceding section, it took roughly two person days of additional work to capture the real world scenes at the different work places, to arrange the captured material to the RWEs, and to relate them to the successively defined goals. However, it should be noted, that we spent almost half of the additional time with digitising and cutting the material properly which was to a large part due to the prototypical nature of our video equipment. Once we had transferred the video material in our PRIME-CREWS environment, the actual work of creating links between RWEFs and goal model components was significantly facilitated by the method guidance provided by the environment

According to this experience, the following aspects should be considered before applying our approach:

- *Group of stakeholders with different backgrounds:* If only two or three stakeholders with a similar background of knowledge and experience in the application domain have to negotiate a current-state model, it is likely that they find an agreement without referring back to related RWEFs. But the more stakeholders have to communicate and the more diverse background knowledge they have, the more essential it is to make the abstraction processes transparent. In this situation, the comparison of several real-world fragments related to the same goal and the visualisation of conflicts between the original modeller and his reviewer(s) enables efficient negotiation processes. Even if only few stakeholders are involved in system development, the RWEFs and their interrelations with the goals could be essential to understand the actual system during system maintenance or in the case of future system improvements or replacements.
- *Correspondence of RWEFs and goals:* In some cases, the correspondence between fragments of the RWEs under consideration and the goals to be validated or elicited may not be obvious for an external system analyst by merely watching the RWEs. In these situations, it is necessary to exploit additional sources of information, e.g. to involve domain experts during the extraction and relation of RWEFs to goals. Another technique we used in the ADITEC example is to let the observed workers think aloud about the actions they are performing during the observation. This facilitated the detection of certain goals behind RWEs significantly.
- *Pre-existing system and observable system usage:* Our approach relies on the assumption that there exists a current system which already offers a large portion of the functionality of the new 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.
- *Expensive, hard-to-repeat and hard-to-describe observations:* 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, they are still more objective than relying on written minutes or even the personal memory of the analyst.

- *System usage can not be observed by all stakeholders:* Also in this case, the recorded observations are helpful to provide all stakeholders with information which is as close as possible to reality.
- *Delayed analysis of distributed observations:* If certain tasks or system aspects are spatially and/or temporally distributed an observation by just one stakeholder is almost impossible. In these cases, the persistent recording of the observation lays the foundation for separating the observation task from the analysis of the observation. The advantage is that distributed system aspects can be viewed at once and by all affected stakeholders.

7 Related Work

Some user-centred design and ethnography approaches provide a detailed descriptions for preparing and performing real world observations and for interpreting the observations captured on video and other recording means (e.g., SEP [26], Xerox Parc [5],[43]). Defining such a method, however, was not the focus of our work. We concentrated our efforts on providing support for interrelating recorded observations (RWEs) and conceptual models, especially goal models, and on using these interrelations for explanation, evaluation and comparison of RWEs and goal models.

In Sect. 4.1 we described how RWEFs related with goals can be used to ground communication and explanation of abstract concepts on instances. Systems providing multimedia management facilities like AMORE [10],[45] or Raison d'Etre [9] and approaches supplying paper based access to captured multimedia [5],[26] are used for explanation and communication of concepts and requirements between stakeholders with various backgrounds and less formal training. Another class of approaches relates textual scenarios and conceptual models. ScenIC from Colin Potts et al. [1],[2],[35],[36] and Alistair Cockburn's extended use cases [8] are two approaches that are especially featuring goal models. Whereas in ScenIC, goals are extended with scenarios for explanation and exploration, Cockburn extends use cases with goals for structuring them. The cited publications only support the explanation of abstract concepts with examples, but not vice versa. For instance, AMORE, aggregates multimedia objects representing sources and background information to requirements. For Raison d'Etre a keyword-based search on video transcripts is implemented for assessing media related to a special topic of interest. All these approaches do not provide explanation and information on the examples themselves. As described in Sect. 4.1.2, in PRIME-CREWS the goals behind a real world example can be queried, or the example can be analysed with respect to its quality (positive or negative), problems revealed etc.

Creating conceptual descriptions of how things are performed in the real world in the form of activity lists in SEP [26], storyboards and workflow maps at Xerox Parc [5], or data and control flow diagram with AMORE makes a comparison of real world examples difficult. Stakeholders might perform the same task differently and there might be several completely different ways of reaching a goal. McGraw and Harbinson demand comparison on a higher level, but only define activity-based comparison of observations, which are evaluated using occurrence ratios for the activities. In Sect. 4.3.1 we presented a possible solution for this demand by comparing RWEs in respect to goals. Evaluating dependency relations between RWEs and goals enabled us to display differences and possible correspondences between two examples as well as the determination of overall relevance and success of goals in the real world.

As described in Sect. 4.2, the abstractions performed by analysing RWEs are always personal interpretations of the analyst. Different stakeholders may have different interpretations. Therefore, it is necessary to support several stakeholders in eliciting and validating conceptual models as well as reviewing the others interpretations against the RWE and then to be able to display differences between them. All approaches cited above provide only one interpretation for a real world observations. They are thus not considering these problems.

8 Conclusions

We have presented an approach for bridging the gap between concrete examples of current system usage (scenarios) and conceptual current-state models. To support the definition of a conceptual goal model of the existing system (current-state model) we have proposed to capture system usage using rich media and to interrelate those observations with the goal definitions. More precisely, we have proposed to relate the RWEFs which have caused the definition of a goal or against which a goal was validated with the corresponding goal. Thereby a fine-grained interrelation between the conceptual goal model and the recorded observations is established. Thus, our approach particularly aims at making the abstraction process more transparent and traceable.

We have defined basic method fragments for establishing typed traceability relations between the captured RWEs and the goal models. In addition, we have defined method fragments for using the fine-grained interrelations to visualise the evidence of the defined goals, to compare different stakeholders' viewpoints and to compare different RWEs.

We have implemented our approach in a prototypical environment, PRIME-CREWS. PRIME-CREWS supports the creation and use of the interrelations between the goals and the RWEFs. It offers tools for multimedia management, goal modelling and for the visualisation of the various annotations computed based on the fine-grained interrelations.

We have illustrated the main benefits of our approach using examples drawn from a trial application at ADITEC. According to our experience, interrelating captured observations with conceptual goal models facilitates the definition and agreement of the main features provided by the existing system. Amongst others, the ADITEC trial application has shown that

- new team members and untrained stakeholders use the fine-grained interrelation to access RWEFs as explanations for goals defined in the current-state model;
- comparing different RWEs and comparing goal models defined by different stakeholders is facilitated by the annotated goal trees;
- reviewing conceptual models is significantly improved by using the typed interrelations to understand and justify the abstraction process made during the definition of the conceptual models;
- the visualisation of the review results in an annotated goal tree improves the resolution of detected conflicts/shortcomings.

Our approach of interrelating RWEFs with model components can be adapted to any kind of conceptual target model, i.e. the modelling language used to define the current-state can be manifold. For example, data (e.g., entity relationship models [6] or the static structure diagrams of UML [37]), behaviour (e.g., statecharts [17]), and/or functional (e.g., data flow diagrams [46] or business process models [42]) conceptual models can be used as target languages. In contrast to goal languages, such languages are mostly used during later RE phases.

The adaptation of our approach to another target language, as our own experience with message trace diagrams and static structure diagrams of UML indicates, requires the definition of new interrelation types to be used to relate the RWEFs with the concepts provided by the target language chosen.

Future research is concerned with the

- *propagation of goal-RWEF relations to higher-level goals*: We will investigate how relationships of medium- and low-level goals to RWEFs can be propagated to higher-level goals which are not directly observable, especially when sub-goals of a common super-goal are related to the same RWE by different link types.
- *consideration of additional target languages*: We will consider the static structure and behavioural system description of UML as target languages and investigate in the goal-driven definition of static structure and behaviour models using the interrelated RWEFs and the goals to support their definition and viewpoint resolution.
- *use of existing current-state model:* We will investigate in the application of our approach for validating and adjusting existing current-state and system design models against newly captured real world observations.

- *applicability of patterns:* To improve the guidance offered for the elicitation and validation of conceptual (goal) models from captured observations, we will integrate patterns (e.g., Problem Patterns [19], Analysis Patterns [16]) in our approach.

Acknowledgements. This work is funded by the European Community under ESPRIT Reactive Long Term Research 21.903 CREWS (Cooperative Requirements Engineering with Scenarios). The authors are grateful to their students Sebstian Brandt, Sven Ewald, Markus Hoofe, and Michael Mattern for implementing major parts of the PRIME-CREWS environment.

9 Bibliography

[1] A.I. Antón, W.M. McCracken, and C. Potts, "Goal Decomposition and Scenario Analysis in Business Process Reengineering", *Proc. 6th Int'l. Conf. Advanced Information Systems Eng. (CAiSE'94)*, Springer, Utrecht, NL, June 1994, pages 94-104.

[2] 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.

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

[4] G. Booch, *Object-Oriented Analysis and Design with Applications.* B/C Publishing Company, Inc., 1994.

[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] P.P.S. Chen, "The Entity-Relationship Approach: Towards a Unified View of Data", *ACM Transactions on Database Systems*, Vol. 1, No. 1, Mar. 1976, pp. 9-36.

[7] K.L. Chung, *Representing and Using Non-Functional Requirements for Information System Development: A Process-Oriented Approach*, Ph.D. thesis, also Tech. Rpt. DKBS-TR-93-1, Dept. of Comp. Sci., Univ. of Toronto, Jun. 1993.

[8] A. Cockburn, "Structuring Use Cases with Goals", *Journal of Object-Oriented Programming*, Vol. 10, No. 7, Nov. 1997.

[9] J.M. Carroll et al., "Raison d'Etre: Capturing Design History and Rationale in Multimedia Narratives", *Proc. ACM CHI'94 Conf.*, ACM Press, Boston, Massachusetts, USA, Apr. 1996, pp. 192-197.

[10] M.G. Christel, D.P. Wood, and S.P. Stevens, *"AMORE: The Advanced Multimedia Organizer for Requirements Elicitation"*, Tech. Report CMU/SEI-93-TR-12, ESC-TR-93-189, SE Information Modelling Project, SE Institute, CMU, Pittsburgh, Pennsylvania 15213, Jun. 1993.

[11] 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.

[12] T. DeMarco, *Structured Analysis and System Specification,* Yourdon Press, New York, 1978.

[13] R. Dömges, K. Pohl, and K. Schreck, "A Filter-Mechanism for Method-Driven Trace Capture", *Proc. 10th Int'l. Conf. Advanced Information Systems Engineering (CAiSE'98)*, Springer, Pisa, Italy, Jun. 1998, pp. 237-250.

[14] M. Dowson, "Consistency Maintenance in Process Sensitive Environments", *Proc. Process Sensitive Software Eng. Environments Architectures Workshop*, Boulder, Colorado, USA, Sep. 1992.

[15] A. Finkelstein et al., "Inconsistency Handling in Multi-Perspective Specifications", *IEEE Transactions on Software Eng.*, Vol. 20, No. 8, 1994, pp. 569 - 578.

[16] M. Fowler, *Analysis Patterns: Reusable Object Models.* Addison-Wesley, 1997.

[17] D. Harel. "STATECHARTS: A Visual Formalism for Complex Systems", *Science of Computer Programming*, Vol. 8, 1987, pp. 231-274.

[18] ISO/IEC, *Information Technology - Information Resource Dictionary System (IRDS) Framework.* International Standard ISO/IEC 10027, first edition 1990-06-15, 1990.

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

[20] S. Jacobs and R. Holten, "Goal-Driven Business Modelling – Supporting Decision Making within Information Systems Development. *Proc. Conf. Organizational Computing Systems*, Milpitas, Calif., 1995, pp. 96-105.

[21] I. Jacobson et al., *Object-Oriented Software Engineering: A Use Case Driven Approach.* Addison-Wesley, 1992.

[22] M. Jarke and K. Pohl, "Establishing Visions in Context: Towards a Model of Requirements Processes", *Proc. 14th Int'l. Conf. Information Systems*, Orlando, Florida, USA, Dec. 1993, pp. 23-34.

[23] M. Jarke and K. Pohl, "Requirements Engineering in 2001: (Virtually) Managing a Changing Reality", *Software Eng. Journal*, Nov. 1994.

[24] A. van Lamsweerde, R. Darimont, and P. Massonet, "Goal Directed Elaboration of Requirements for a Meeting Scheduler: Problems and Lessons Learnt", *Proc. 2nd IEEE Int. Symp. on Requirements Eng. (RE'95),* York, England, 1995, pp. 194-203.

[25] M. Lundeberg, G. Goldkuhl, and A. Nilsson, "A Systematic Approach to Information Systems Development (I+II)", *Information Systems*, Vol. 4, No. 2+3, 1979, pp. 1-12 & 93-118.

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

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

[28] B. Nuseibeh, J. Kramer, and A. Finkelstein, "A Framework for Expressing the Relationships Between Multiple Views in requirements Specifications", *IEEE Trans. On Software Eng.,* Vol. 20, No. 10, Oct. 1994, pp.760-773.

[29] K. Pohl, "The Three Dimensions of Requirements Engineering: A Framework and its Application. *Information Systems*, Vol. 3, No. 19, Jun. 1994, pp. 243-258.

[30] 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.

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

[32] K. Pohl, R. Dömges, and M. Jarke, "Towards Method-Driven Trace Capture", *Proc. 9th Intl. Conf. Advanced Information Systems Engineering (CaiSE'97),* Barcelona, Spain, Jun. 1997, pp. 103-116.

[33] K. Pohl and P. Haumer, "Modelling Contextual Information about Scenarios", *3rd International Workshop on Requirements Engineering: Foundation for Software Quality (RESFQ'97),* Barcelona, Spain, Jun. 1997, pp. 187-204.

[34] K. Pohl and K. Weidenhaupt, "A Contextual Approach for Process-Integrated Tools", *Proc. 6th European Software Engineering Conference (ESEC'97) and 5th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE'97)*, Zurich, Switzerland, LNCS 1301, Springer, Sep. 1997, pp. 176-192.

[35] C. Potts, K. Takahashi, and A.I. Antón. "Inquiry Based Requirements Analysis", *IEEE Software*, Vol. 11, No. 2, Apr. 1994, pp. 21-32.

[36] Colin Potts. *Determining Requirements for Evolving Systems*. In Tutorial Notes and Slides, CAiSE '97, Barcelona, Catalonia, Spain, Jun. 1997.

[37] 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.

[38] B. Regnell, K. Kimbler, and A. Wesslen. "Improving the Use Case Driven Approach to Requirements Engineering", *Proc. 2nd IEEE Int. Symp. on Requirements Eng. (RE'95),* York, England, Apr. 1995, pp. 40-47.

[39] B. Regnell, M. Andersson, and J. Bergstrand. "A Hierarchical Use Case Model with Graphical Representation", *Proc. IEEE 2nd International Symposium of Computer-Based Systems (ECBS'96),* IEEE Computer Society Press, Friedrichshafen, Germany, Mar. 1996, pp. 270-277.

[40] C. Rolland et al., "A Proposal for a Scenario Classification Framework", *Requirements Engineering Journal*, Vol. 3, No. 1, 1998, pp. 23-47.

[41] J. Rumbaugh et al., *Object-Oriented Modeling and Design,* Prentice Hall, 1991.

[42] A.W. Scheer, *Architektur Integrierter Infomationssysteme [Architecture of Integrated Information Systems]*. Springer, 1990 (in German).

[43] 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.

[44] 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.

[45] D.P. Wood, M.G. Christel, and S.M. Stevens, "A Multimedia Approach to Requirements Capture And Modelling", *Proc. 1st Int'l. Conf. on Requirements Eng. (ICRE'94),* IEEE Computer Society Press, Colorado Springs, CO, USA, Apr. 1994, pp. 53-56.

[46] E. Yourdon, *Modern Structured Analysis,* Prentice-Hall, Englewood Cliffs, NJ, 1989.

[47] 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.

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