Establishing Visions in Context: Towards a Model of Requirements Processes

Matthias Jarke, Klaus Pohl
Informatik V, RWTH Aachen,
Ahornstr. 55, 52056 Aachen, Germany
{jarke, pohl}@informatik.rwth-aachen.de

Abstract

A model of requirements determination as the process of establishing visions in context explains how both new ideas and existing habits influence diversity in a family of information systems applications. ***Visions*** *are operationalized as non-functional requirements which are broken down according to constraints imposed by context, and traded off against other non-functional requirements.* ***Context*** *is organized according to four "worlds", taking into account the need for considering application domain (subject world), organizational context (usage world), existing systems (system world), and the development environment itself (development world).* ***Process*** *is modeled as driven by context-dependent decisions which, together with external factors, cause moves within a three-dimensional space of cognitive understanding, social agreement, and technical representation. The framework leads to a formally based and computer-supported requirements engineering environment which is currently developed and practically evaluated by the ESPRIT project NATURE.*

1 Introduction

In software engineering, the requirements phase has traditionally been perceived as the fuzzy and somewhat ugly part in which a formal specification is gained from informal ideas. In contrast to this tradition stands the growing demand for "corporate ownership" of information technology. Organizations cannot take full advantage of information systems which they don´t understand and therefore don´t trust. They need to understand software systems in application terms, i.e. from the requirements point of view. It is also from this perspective that they wish to evaluate software-intensive systems and to direct their change. This paper offers an integrated framework for requirements engineering processes, developed in an ESPRIT project called NATURE (Novel Approaches to Theories Underlying Requirements Engineering). Under requirements *engineering* (RE), we understand the two tasks of requirements *determination* and of requirements *management* but most of this paper focuses on the former task.

Like any expensive activity, RE is not conducted out of the blue. Before an RE team is established, some need for change is defined by rationalising perceived opportunities or threats, or by political decision-making.

The need for change is typically stated in a simple manner, which we call the *system vision*. A classical example is John F. Kennedy´s *"send a man to the moon before the end of the decade"*. Thus, RE is not an undirected analysis process, but a process of transforming the vision into a requirements specification which can then, in the design and implementation tasks, serve as a framework for making the necessary changes in the real world.

Within the world in which the vision is to be realized, many *habits* exist. Some are based on formally stated goals, policies, or competing visions. Others are just regularly observable phenomena for which no predefined structure or reasons are known a priori.

The task of RE is therefore twofold. First, relevant habits must be analysed and the goals, policies and visions behind them must be made explicit. This is essentially a goal-directed abstraction process of existing practice (Dardenne et al. 1992). Second, the new vision must be established as a mission in this context. Propagation of their consequences leads to the detection and resolution of conflicts among the different viewpoints (Finkelstein et al. 1990). During this RE process, the vision often shifts. Large projects often appoint a "vision holder" to make sure that it does not get totally lost in the constraints of current practice.

Many RE methods emphasize one of these aspects at the expense of the other. Top-down decomposition methods such as Structured Analysis (McMenamin and Palmer 1984) do not say how context influences decomposition, whereas bottom-up approaches (ethnographics is an extreme example) so far fail to provide guidance how analysis can be focused on the vision (Goguen and Linde 1993).

The need for supporting both aspects simultaneously becomes more important as we are now in a situation where software systems have become pervasive in the real world. Many systems must evolve in multiple contexts and diverse configurations. Buzzwords such as Composite Systems (Fickas 1987) or Cooperative Information Systems (Brodie and Ceri 1992) accentuate this observation.

Requirements processes must become traceable and reusable. Since context is complex and changing, some structure is needed by which traces can be organized.

In section 2, we propose a framework of four worlds of information systems modeling as a prediction for role-based context influences, and of three dimensions along which requirements engineering activity can be described. This framework drives a situated and decision-oriented RE process model where the four worlds serve to characterize expected situation patterns, and the three dimensions help characterize consequences of decisions. The model is further differentiated according to the contribution each type of decision makes to development goals (functional or non-functional), thus explicitly supporting a vision- as well as context-aware development process. Finally, in sections 4 and 5, we use a simple example to illustrate the interplay of these components and the kind of support offered by the NATURE environment.

2 Modeling the Context of Requirements Engineering

Establishing a vision in context remains an empty phrase if we do not understand what parts of the world are relevant and how they are related to the development process. Due to the diversity of RE tasks, even an open basic formalism such as the "rich pictures" of soft systems methodology (Avison, Golder and Shah 1992) is not enough. We need some kind of domain ontology — a basic understanding of what IS requirements engineering is concerned with. This ontology should have a very simple structure to be acceptable to a broad class of developers in that it is easily understood and does not overly constrain their freedoms in the RE process.

This section identifies two meta models with these features. It presents a structure of four interrelated worlds as a useful partitioning of the context of information systems requirements engineering; and it characterizes three dimensions along which the problems and impact of requirements engineering activity can be described. Recently proposed sophisticated (and typically very complex) information system architectures (Scheer 1990; Heym and Österle 1992) are compatible with our approach but take a much more normative position towards details of what should be presented. They are thus not targeted towards more open tasks such as requirements determination.

2.1 Four Worlds of Information Systems Modeling

The context of information systems modeling is in general quite unpredictable. RE simplifies the problem in two ways. First, it concentrates on identifying recurrent patterns and tends to ignore spurious events unless they concern important non-functional requirements such as safety. Second, certain habits can be predicted through a basic ontological theory about the object of the RE process: the information system.

The theory we are going to follow is intentionally very simple (figure 1). It sees an information system in analogy to *a sharable telescope through which a user community observes a domain of interest* more effectively than without the IS. The domain of interest may or may not overlap with the user community itself, it may or may not be changeable by the user community. But it makes sense to distinguish, from a cognitive as well as a social viewpoint, between the *usage world*, the *subject domain world*, and the *system world*, and to describe their interrelationships. A fourth world, the *development world*, has the basic task of assisting the vision holder in realizing the vision in the context of the other worlds. But it must also consider its internal development context of people, methods, experiences, and tools.

There have been many other partitionings of IS knowledge which we do not have the space to discuss in detail. We argue for the four-worlds approach on pragmatic grounds: helping to determine complete, agreed requirements and getting the right people involved in the process.

Each world is associated with certain *groups of stakeholders* who should be considered in requirements decision making. The model makes the social prediction that different areas of expertise, different languages, and different interests exist and need to be integrated.

Moreover, it predicts certain *role-bound non-functional goals* which can be associated with the relationships between the worlds; dealing with such standard viewpoint resolution and negotiation tasks becomes a natural target for RE methods and tools.

Finally, we can expect that representatives in each of the worlds will have *implicit or explicit models* about their own as well as the other worlds which leads to further social, cognitive, and technical problems. We found that these models are (and should be) quite different for each of the four worlds; this is exploited in the design of the NATURE environment (Jarke and NATURE Team 1993).

Let us elaborate these general observations for each of the four worlds.

The ***subject world*** is the domain the system is intended to maintain information about — traditionally studied in database design. Stakeholders are the objects being represented (e.g., in a criminal record or hospital system), or people who have stakes in these objects but are not system users (e.g., owners of real estate about which managed by a brokerage system). Relationships to the usage world are often governed by legal concerns such as privacy and ownership. Relationships to the development world are hard to establish — frequently, subjects know nothing about the degree to which they are administered by systems and thus hardly influence its development until it is too late. The relationship between system and subject world can be described by *quality-of-information factors* such as accuracy or timeliness.

From a representational perspective, the subject world is the most general one. Significant support beyond entity-relationship or structured analysis diagrams or their formal counterparts can be offered only by domain specialization. The NATURE environment supports a domain theory of meta concepts that have been empirically shown to discriminate well among domains, and are thus suitable for recognizing reusable domain models that could be brought into a context analysis (Maiden and Sutcliffe 1992).

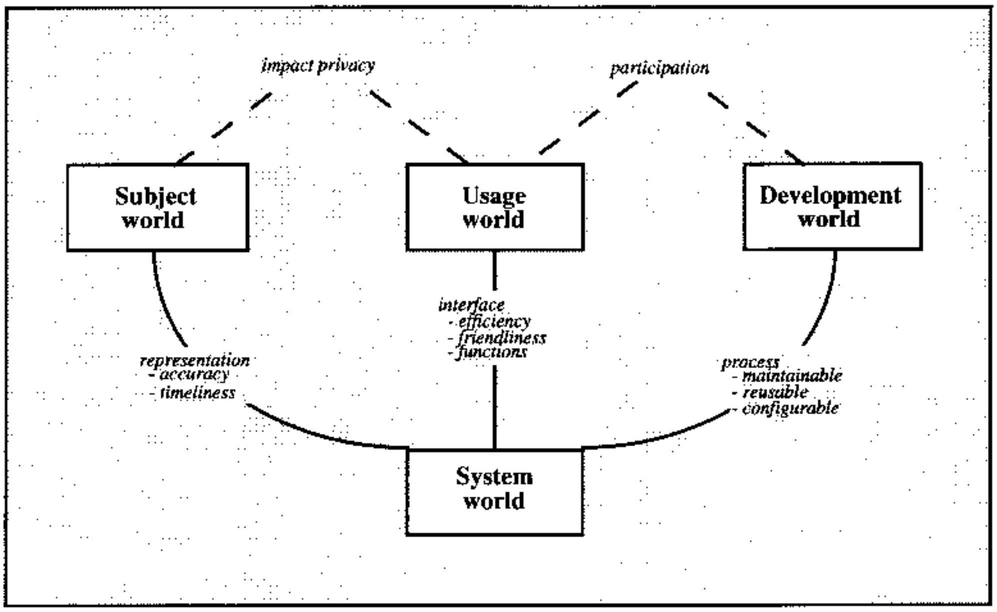

Fig. 1: The "worlds" of information systems modeling

The ***usage world*** comprises as stakeholders the owners, direct and indirect users of the system. The relationships between users and owners can vary widely. An organizational structure often describes it, thus defining the (observed or intended) role of the information system in the work practice in or between organizations. With respect to the development world, the goal of participative design has often been cited. With respect to the system, *quality-of-interaction factors* such as response time, user-friendliness, and rich functionality are of interest as well as business goals such as reduced transaction costs, increased worker/ organization effectiveness, or worker qualification.

From the representational perspective, business modeling as well as research in human-computer interaction are relevant to the usage world. NATURE is taking a speech-act perspective to define the role of information systems as a communication medium in the user community (Winograd and Flores 1986). The NATURE environment provides tools that help to link system functions to the organizational conversations they support.

The ***system world*** is often represented by technical gurus, or simply by the observed fact that the system is very hard to change, either due to its internal complexity or to its established relationships to user and developer communities.

Representationally, it is probably the best understood subworld with many available specification languages (Jarke 1993). It contains a representation (function-oriented, data-oriented, object-oriented, or whatever) of the subject world, maybe also of its own users, together with some mapping of these conceptual specifications to the design and implementation levels. The latter pose their own technological requirements. Due to diversity and evolution in the other subworlds, a system often exists in different versions. It may be part of different usage environments, and it may have lost much of its initial structure by changes before the present "vision" came about.

All of this must be considered in the fourth world, the ***development world***. It must ensure an adequate observation (cognitive aspect), participation (socio-organizational aspect), and representation (technical aspect) of itself and of the other worlds. It must proceed under consideration of resource constraints and competing role-bound and individual goals. More details about the development world are given in the next sections.

The ***vision for change*** could come from any of the worlds. It could be driven by technology push in the system world (e.g., moving from files to databases), changes in the subject domain (e.g., protests by privacy pressure groups, new theories about the subject domain suggesting knowledge reorganization), or long-term development concerns (improved maintainability, change of development responsibility from computer professionals to end users). Most frequently, of course, visions arise in the usage world (application pull). The four-worlds framework is important to position the vision within the context, and to predict where the main obstacles in the RE process may come from and how they could be overcome.

2.2 Three Dimensions of Requirements Engineering Activity

Only a portion of the development process can be explained from the role expectations resulting from the four-worlds model. The situation and progress of a requirements process is equally influenced by the innovative ideas coming from the system vision, and by individual differences and social behavior in and around a requirements team.

These problems cannot and need not be fully explicated in each individual case. However, their "essence" (McMenamin and Palmer 1984) with respect to the requirements process, their influence on the basic inputs and outputs of RE, should be captured as a basis for traceability, reuse, and process guidance.

The RE process starts with an opaque understanding of the problem, many different and conflicting viewpoints, and an informal representation. Ideally, it should end with full agreement on a well-understood requirements specification represented precisely enough to be transformable into a correct implementation. In other words, to be successful, the RE process has to make progress along three dimensions which we call the specification dimension, the agreement dimension, and the representation dimension (Pohl 1993).

The ***specification dimension*** deals with the degree of requirements understanding at a given time. Initially, only the vision is given. The understanding of the system and its environment is more or less opaque. Focusing on this dimension, the aim of RE is to transform the vision into a complete system specification through an iterative process of determination and validation. Standards and guidelines describe what the final requirements specification should include (e.g., IEEE 830, British Standard 6719, ESA PSS-05-0). Empirical studies show that problems in achieving this aim include (Curtis et al. 1988, Goguen and Linde 1993):

- incomplete domain knowledge, poor and hazy system goals
- premature search reduction on a small set of sub-optimal solutions
- inadequate articulation of requirements

The ***agreement dimension*** deals with the degree of agreement reached on a requirements specification. Each person involved in the RE process has a personal view of integrating vision and habits. Few requirements may be initially shared among the team. It is important to recognize that formal view integration techniques and agreement on the integrated view are two separate things. A detected conflict or misunderstanding must be resolved through communication among people. A process of negotiation — comprising bargaining as well as mutual learning — and coordinated subgroup actvities may lead towards better understood personal views that have at least sufficient commonality to start building a system. The process of gaining a common view is affected by social, psychological, and political factors. Its problems are often categorized at three different levels (Ellis, Gibbs, and Rein 1991):

- communication, the technical ability to exchange messages
- interoperability, the technical and social ability of mutual understanding
- cooperation on a common task.

The ***representation dimension*** copes with the different representations used for expressing and managing knowledge about the requirements process and its products. First, both abstract requirements specifications and concrete examples and scenarios are needed in RE. Second, the representation should enable the description of states and transitions within the considered worlds, as well as perceived intentions behind them. Third and most obviously, three categories of representation are distinguished with respect to formal semantics: informal (natural languages, graphics), semi-formal (SA / ER diagrams) and formal specification and knowledge representation languages, such as VDM (Jones 1990), Z (Spivey 1989) ERAE (Hagelstein 1988), or the Telos language (Mylopoulos et al. 1990) we are using in NATURE.

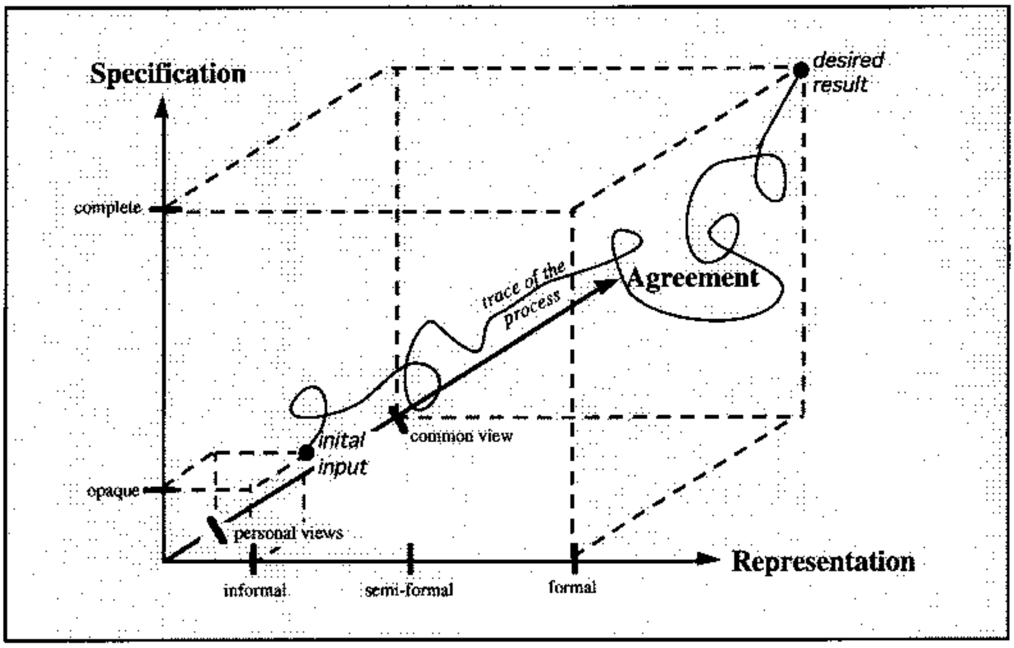

Fig. 2: The three dimensions of RE.

Due to the unique advantages of each category, people prefer different representations in different situations. Recent surveys (Lubars, Potts, and Richter 1993) confirm that people use informal representation at the beginning of the RE process, then move to semi-formal graphics, at the end also to formal representations. Ideally, the final specification should be transformable to executable code, at least at the level of rapid prototyping. The different representations used must be kept consistent and the transformation between them must be supported and recorded.

The representation dimension is orthogonal to the specification dimension. For example, formality can hide a poor understanding of the system, whereas precise ideas can be represented using natural language (law makers try to do so).

The trace of the RE process is a curve within the three dimensions (cf. figure 2). The traditional requirements capture process would start near the origin with different personal views, little system understanding and a completely informal representation; it should end with agreement on a well-understood and formally described specification. Aberrations, even loops, occur since the worlds of IS modeling are changing while they are analyzed. Nevertheless, the three-dimensions framework has proven very useful in the following tasks (Pohl 1993):

- analyze local or global problems in large requirements processes concerning progress in any dimension, and recognizing generalizable method patterns from experience

- evaluate standards and methods proposed for RE; for example, the published standards only support the specification dimension but say little about agreement and have only implicit assumptions about representations
- configure methods and environments for comprehensive RE process support such that not only the context but also the vision is considered.

3. A Vision and Context-Driven RE Process Model

We have so far seen how information about the context of RE can be usefully organized with explicit consideration of typical goals (the four worlds). We have also proposed three dimensions of how to look at the intent and impact of RE activity. The next step is to devise a process model that not only takes this information into consideration but also helps explicitly to establish and maintain the given system vision in this context.

We shall tackle this in two steps. The basic process meta model incorporates our four worlds to identify situations as opportunities to act in the RE process, and the three dimensions as a way to analyse the impact of requirements decisions. Then, we expand this basic model by the explicit consideration of goal-oriented activity, thus supporting vision-driven development, control of vision conformance, and meta-level process improvements towards avoiding repeating mistakes and reusing successful process patterns.

3.1 Dealing with Context Knowledge

It is obvious from our discussion so far that, in requirements engineering, the context creates a lot of surprises during the move along the three dimensions. Since RE is much less structured than other phases of the development process, local micro-planning and backtracking are inherent (Ramesh and Dhar 1992, Souquieres and Levy 1993), and therefore process trace is essential. The more knowledge the trace provides about what has been done, why, when, and by whom, the more efficient the backtracking will be. In fact, even the definition of requirements process models is usually a slow iterative process in which methods only gradually emerge from experiences.

For these reasons, traditional process models which prescribe complex activity patterns appear unsuited for requirements engineering. In fact, requirements *determination* may be a better word than requirements *engineering*. Requirements engineers need to react with flexible analysis decisions to rapidly changing situations (Suchman 1987). Local reuse of chunks of experience is more important than a global step-by-step procedure that will probably be circumvented anyway. The NATURE process model, NATPROC, therefore advocates a situated and decision-oriented approach. (Grosz and Rolland 1991, Schmitt 1993).

NATPROC associates the situation the requirements engineer faces to the decisions he can take to solve the local problem, and to the actions which can be performed for applying the decision (figure 3). The set of triplets <situation, decision, action> comprise the basic RE process trace.

The formalism is also used to collect libraries of development process chunks which the designer can reuse to evolve the requirements. These chunks are composed of frequently-used generic patterns that the analyst can tailor to his needs. The situation part of the triplet is such a pattern, the decision part reflects the tailorization, and the action part is composed of the actual transformations to be performed.

The concept of situated and decision-oriented process models is intentionally very broad. Existing process meta models can be characterized by the ways how they describe situations, actions, and objects in this generic framework:

- Models based on finite automata or Petri nets take an uninterpreted "state" as the description of the situation. Thus, the situation is characterized by a (limited) memory of the process history; their action part changes this state.
- "Opportunistic" models, including MARVEL (Kaiser, Feiler, and Popovich 1988) or the DAIDA model (Rose et al. 1991) define situations through the state of the object base.
- Models based on extended net formalisms (e.g., predicate transition nets) additionally consider the object state through a precondition when chosing an action (Madhavji 1992).

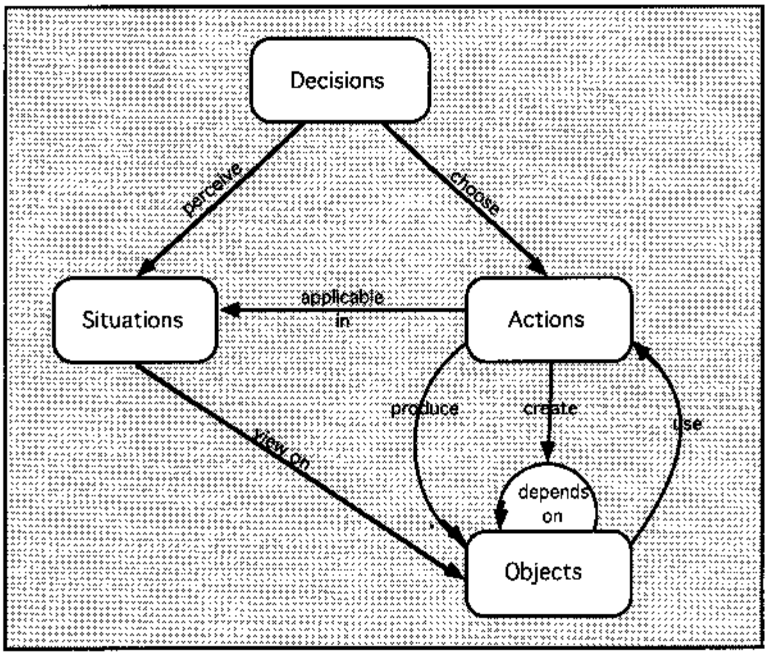

Fig. 3: The NATPROC model is based on situated decisions

While this generality gives us the needed flexibility in the RE process, there is also the danger that little structure is offered for process capture and guidance. We therefore anchor this process model in the discussion of context modeling in section 2.

Knowledge about possible process situations can be structured using the four worlds of information systems; the latter can also be used to explain a decision. The state of an object and therefore the situation changes by performing process actions. Expected changes of object state and therefore situation can be predicted within the three dimensions.

For example, a user interface RE activity would be concerned with an aspect of the link between usage and system world; it could be described in terms of its impact on increasing or decreasing the agreement about the specification, the understanding of the UI requirements, or the formality of the representation. Voting on a proposed requirement would only promote the agreement dimension, whereas an in-depth discussion might increase both agreement and understanding. To give another example, too much time spent with formalizing a certain requirement might emphasize hidden but unimportant disagreements and could thus actually constitute a step backward in agreement; even understanding might suffer due to waste of time with formalism rather than substance.

3.2 Supporting the Vision by Goal-Oriented Modeling

Types of product, situation, decision, and action can be defined within NATPROC and organized in a generalization hierarchy. By refining these interrelated type lattices, we can model RE activities at macro and micro levels of detail (Schmitt 1993). The context is structured by the known situations and methods within the four worlds and the three dimensions. However, we lack means to represent how the the vision of the system can drive the process.

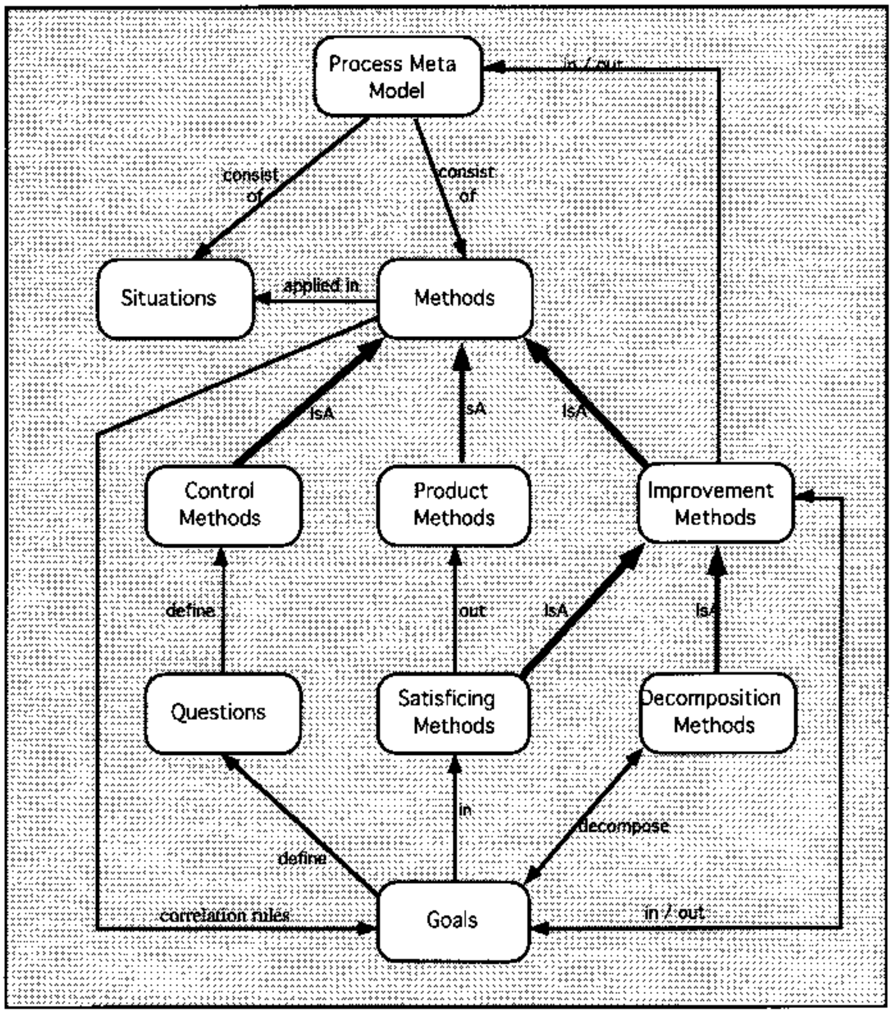

Fig. 4: Classifying methods by their relationships to goals

To support the concept of vision and its trade-off against rationale and reality of the existing habits, we introduce the notion of non-functional *goal* into the NATPROC model. Figure 4 shows how this notion can determine the organization of the process model, by classifying methods according to the role they play with respect to goal achievement:

- *Product methods* are designed to satisfy goals through requirements determination, design, or implementation activities.
- *Control methods* measure achievement of goals through answering questions.
- *Improvement methods* help users exploit experiences or new goals to re-define the process itself; examples of improvement methods include the decomposition of goals into sub-goals or the choice and definition of suitable product methods through satisficing methods.

A typical RE process would start with a simple goal (the vision) and a very loose process model for the first few exploratory steps. It would thus begin with decomposition methods of goals into subgoals, until a sufficiently concrete level has been reached that the goals become comparable with those offered in the four-worlds framework. In the further decomposition, the RE team can reuse knowledge about the correlations between the goals derived from the vision, and those identified as relevant context goals (from the four worlds); this is represented in figure 4 by the link "correlation rules". The RE process ends when problems are reached for which basic solution strategies are known; this is represented by the fact that there are satisficing methods linking identified goals to product methods.

Recall from figure 3 that each decomposition or satisficing decision is documented by dependencies; thus, backtracking and re-decomposition or a new choice of satisficing method is supported. If this backtrack constitutes a generalizable experience, another subclass of improvement methods (not shown in figure 4) is invoked to record it as a reusable process chunk. The relationships between a given requirement and the vision also determines the criticality of a requirement.

It can be shown (Pohl and Jarke 1992) that the model in figure 4 subsumes three recent proposals that address, separately, reasoning with non-functional goals by relating them to what we call product methods (Mylopoulos, Chung, and Nixon 1992); improvement-oriented quality measurement (Oivo and Basili 1992); and goal-oriented process change management (Madhavji 1992). Further semantics can be given to this goal-oriented differentiation of methods by associating them with the object types they work on and appropriate specializations of the situations they are applicable to.

4 An Example

We illustrate the interaction of the presented models with snapshots from a requirements engineering process for the well-known library example (Wing 1990), given the system vision *"establish a user-friendly lending system for books within our university"*.

The simplified example is as follows. The overall system vision is broken down into subgoals using ***goal decomposition*** methods. The context of this decomposition process is structured using the ***four worlds*** of information systems engineering. Then, the subgoals are further specified. Section 4.2 illustrates the choice among possible solutions for a particular subgoal, *user identification* by reusing or creating ***satisficing*** methods. The progress made is described using the ***three dimensions*** of RE. During the process of goal decomposition and the selection of particular satisfying methods, ***control*** methods assure, that the overall vision is not lost. Specific requirements which describe the constraints for the chosen solution are specified by applying pre-defined ***production*** methods (section 4.3). Finally, section 4.4 gives an example for process ***improvement*** gained during the specification process.

4.1 Initial Decomposition of the Vision

Starting with the overall vision, the requirements engineering team uses the four worlds approach to identify stakeholders and subgoals for the library information system. Some of the interrelations and subgoals are depicted in figure 5.

With the vision of a user friendly book lending system in mind, the user representative proposes a system function for retrieving information about books as a new subgoal (relation between *usage* and *system* world). Together with a system specialist, he reaches a common understanding that this implies the requirement for adequate representation of the book information using, e.g., keywords. Moreover, each user should have the possibility to see his lending data at any time, but this information must not be available to other users (*subject/usage* world). To avoid waiting for a loaned book, he suggests that for often requested books more than one copy should be available (*subject/usage* world).

The systems analyst adds user identification as an important subgoal (*system/usage*). To keep development and maintenance costs down, he also proposes reuse as a further goal for the system development (*system/development*).

We don´t look at the process of agreement on the subgoals in detail. But it should be recognized, that each particular subgoal is based on experience or knowledge of the persons involved. For instance, the user representative may have introduced the information retrieval function since he remembers a situation where he spent hours of browsing through the library catalogues searching for a book of which he neither knew the exact title nor the author. The system specialist may have thought about unauthorized usage of systems in general and therefore raised the subgoal of user identification.

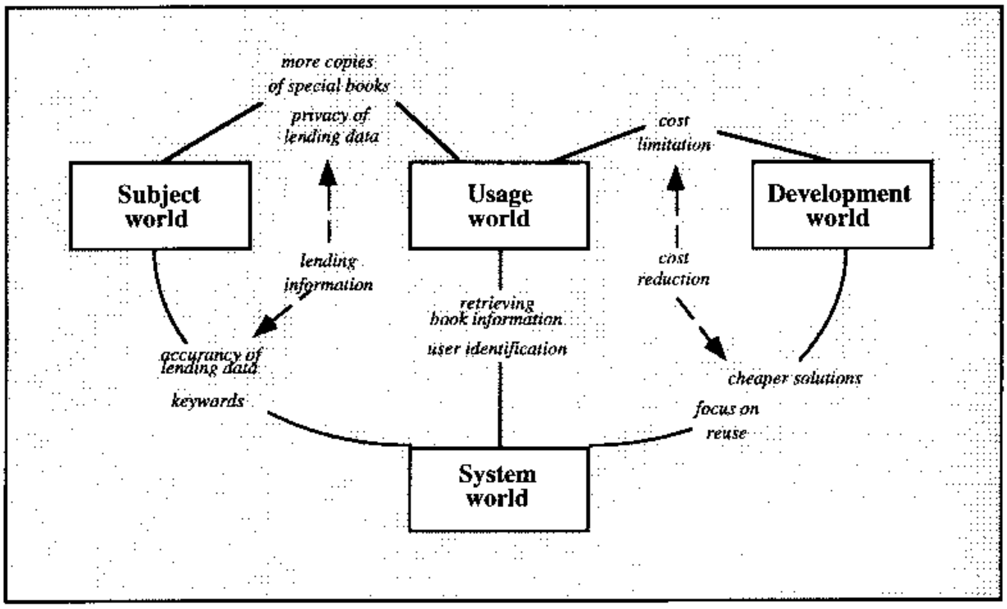

Fig. 5: The "worlds" of information systems modeling

Summarizing, out of the overall system vision *"establish a user-friendly lending system for books within our university"* subgoals are gained, each of them based on different assumptions, experiences and knowledge. The process of goal decomposition can also be viewed under the three dimensions. Identifying a new subgoal leads a step forward in the specification dimension, the agreement on an particular subgoal moves along the agreement dimension, and the representation of a particular goal is reflected by the representation dimension.

4.2 Specialization of Subgoals Using Satisficing Methods

Even when a subgoal is agreed on, the persons involved in the process may have different aims about its further specialization. We focus on the further specification of the subgoal *user identification.* Since the situation of specifying a user identification is well known, a process chunk exists which offers four alternative refinements (*identification cards, password, finger print and voice control*) each with annotations describing the strenghs and weaknesses.

The system specialist is very innovative and sees the choice of voice control or finger print identification systems as a chance for getting experience in a new technique. But he qualifies this, stating that an enormous amount on disk space must be available to store voice and finger print data of each user. The cost department representative argues that there isn´t enough money for either voice control nor finger print identification. He prefers the identification card or password solution. In this discussion, everyone gets a lot of background knowledge about the others and their roles. Even if no agreement is reached, the requirements engineering process has improved in the specification dimension.

First, an agreement is reached that a password system should be used for user identification. But a control method in the agreement dimension requires that each new subgoal must be checked against the overall vision. Within this method, the user representative argues that it is inconvenient for users to learn a password and they easily forget it and are then unable to lend books. Thus, explicit consideration of the vision leads to a backtracking on the specification dimension (password as a subgoal is rejected); vision and the above-mentioned context restrictions instead lead the team to choose the identification card requirement.

4.3 Specification of Subgoals Using Product Methods

A process chunk in the library of process experiences offers possible requirements to be defined for identification cards, e.g., overall number of users, creation of new cards per day, validation time for cards, authorization to create new cards, etc.

By looking at the necessary requirements, the system analyst recognizes that for each potential user an identification card system is already established in the university (student and employment card system), the RE team decides that this system should also be used for the library. Therefore, a more detailed specification of this component is unnecessary.

4.4 Improving the Process Definition

The observation, that identification systems may already exist and that instead of specifying a new user identification system, existing ones may be reused, leads to a general improvement of the defined process chunk.

A control function for checking if a user identification system already exists within the usage world must be performed before a selection between possible alternatives is made. Moreover, the existence of a user identification system serves as an argument for the corresponding solution and may hence influence future choice. Thus, the possible reuse of existing user identification systems is built into the process model. Hence during future specification activities, it is assured, that the RE teams checks for possible reuse.

5. Tool Support in NATURE

Starting from the idea that RE involves establishing a vision in context, we first presented some organizational principles for representing context information and for determining the origin of visions, other goals, and requirements methods and tasks: the four worlds of information systems modeling and the three dimensions of RE.

Our process model takes this context explicitly into account through its notion of situation-based action. To help establishing visions in this model, we categorized methods according to the role they play with respect to RE goals. This way, the innovative aspect introduced by a system vision can be balanced against the existing experiential context.

The models presented here are meant to be descriptive in the sense of a schema for recording requirements processes and their rationales as a means of traceability in requirements engineering and beyond that, in the further system development process. To mention a few goals of traceability (Ramesh and Edwards 1993), the four worlds support accountability, the agreement dimension supports rationale, and the goal-oriented process model leads to coverage of criticality. However, the models are also meant to be prescriptive in the sense of providing guidance about how to organize the requirements process, giving hints where to look for who-what-why questions. They also allow users to create and evolve (through change management) families of process models in diverse yet coherent ways.

The ESPRIT project NATURE has developed a first prototype of a comprehensive RE environment intended to support this approach (Jarke and NATURE Team 1993). Much of the NATURE environment is driven by the idea of comprehensive reuse of existing experiences from all four worlds. Its tools support reverse modeling of existing systems and similarity-based retrieval of system models, classification of modeling situations with respect to subject domain models, libraries of reusable process chunks, predefined usage patterns modeled as speech-act types, and their interrelationships. To leave room for diversity and creativity, the NATPROC support environment offers these only as open opportunities for reuse (modeled as situations in which decisions can be based), not as fixed process structures or generic enterprise models. It combines this with a goal-driven group decision support approach where the actual process goals come from the system vision, and dependencies to the vision are maintained and highlighted throughout the process. Through capturing these dependencies along all three dimensions of RE (including formal and informal representations, rationales and group processes), acquisition of reusable information is also integrated.

A major design goal for the NATURE environment was to achieve requirements engineering *by* reuse and *for* reuse with as minimal additional documentation load for the user as possible. Information capture happens as a side effect of normal user activity, and within the context of well-known standard designer interface technology. For this purpose, informal hypertext and semi-formal graphics like ER and SA diagrams have been extended and linked, on the one hand through a knowledge-based repository, ConceptBase (Jarke and Staudt 1993), for process and data integration, on the other through a communication server for intelligent message exchange by which tools can directly influence the behavior of other tools. Of course, these active components can also influence human user behavior, for example, reminding them to consider the relationships of their possibly very detailed work to the project vision.

Acknowledgements

This work was supported in part by ESPRIT Basic Research Project 6353 (NATURE) and in part by the Ministry of Science and Research of Nordrhein-Westfalen.

References

Avison, D.E.; Golder, P.A.; and Shah, H.U. "A Tool Kit for Soft Systems Methodology." Proc. IFIP WG 8.2 The Impact of Computer Supported Technologies on Information Systems Development, Minneapolis, Mn, June 1992, pp.273 - 287.

Brodie, M.L.; and Ceri, S. "On Intelligent and Cooperative Information Systems: A Workshop Summary." Intelligent and Cooperative Information Systems, Volume 1, Number 2, 1992, pp. 249-289.

Curtis, B.; Krasner, H.; and Iscoe, N. "Field Study of the Software Design Process for Large Systems." Communications of the ACM, Volume 33, Number 11 1988, pp. 1268-1287.

Dardenne, A.; Fickas, S.; and van Lamsweerde, A. "Goal-Directed Concept Acquisition in Requirements Elicitation." Proc. 6th IEEE Workshop System Specification and Design, Como, Italy, pp. 14-21.

Ellis, C.A.; Gibbs, S. J.; and Rein, G. L. "Groupware - Some Issues and Experiences." Communication of the ACM, Volume 34, Number 1, 1991, pp. 39-58.

Fickas, S. "Language Support for the Specification and Development of Composite Systems." ACM Transaction Programming Languages and Systems, Volume 9, Number 2, 1987, pp. 198-234.

Finkelstein, A.; Kramer, J.; and Goedicke, M. "Viewpoint-Oriented Software Development." Proc. Conf. Le Génie Logiciel et ses Applications, Toulouse, pp. 337-351.

Goguen, J.A.; and Linde, C. "Techniques for Requirements Elicitation." Proceedings IEEE Symposium on Requirements Engineering, San Diego, January 1993, pp. 152-164.

Grosz, G.; and Rolland, C. "Using Artificial Intelligence Techniques to Formalize the Information System Design Process." Proceedings International Conference on Database and Expert Systems Applications, Berlin 1991, pp. 374-380.

Hagelstein, J. "Declarative Approach to Information Systems Requirements." Knowledge Base Systems, Volume 1, Number 4, 1988, pp. 211-220.

Heym, M.; and Österle, H. "A Reference Model for Information Systems Development." Proceedings IFIP WG 8.2 The Impact of Computer Supported Technologies on Information System Development, Minneapolis, Mn, June 1992, pp. 215-239.

Jarke, M. (ed.) "Database Application Development with DAIDA." Springer-Verlag, 1993.

Jarke, M.; and Staudt, M. "An Application Perspective to Deductive Object Bases." Proceedings ACM-SIGMOD Workshop on Combining Declarative and Object-Oriented Databases, Washington, DC, May 1993, pp. 17-30.

Jarke, M.; and NATURE Team "Requirements Engineering - An Integrated Perspective of Representation, Process, and Domain." Proceedings 4th European Software Engineering Conference, Garmisch-Partenkirchen, Germany, September 1993.

Jones, C.B. "Systematic Software Development Using VDM" 2nd edition, Prentice Hall, 1990.

Kaiser, G.E.; Feiler, P.H.; and Popovich, S.S. "Intelligent Assistance for Software Development and Maintenance." IEEE Software, May 1988, pp. 40-49.

Lubars, M.; Potts, C.; and Richter C. "A Review of the State of the Practice in Requirements Modelling." Proceedings IEEE Symposium on Requirements Engineering, San Diego, January 1993, pp. 2-14.

Madhavji, N.H. "Environment Evolution: The PRISM Model of Change." IEEE Transaction on Software Engineering, Volume 18, Number 5, 1992, pp. 380-392.

Maiden, N.; and Sutcliffe, A. "Exploiting Reusable Specifications through Analogy." Communication of ACM, Volume 35, Number 4, 1992, pp. 55-64.

McMenamin, S.M.; and Palmer, J.F. "Essential System Analysis." Yourdon Press, 1984.

Mylopoulos, J.; Borgida, A.; Jarke, M.; and Koubarakis M. "Telos – Representing Knowledge about Information Systems." ACM Transaction on Information Systems, Volume 8, Number 4, 1990, pp. 325-362.

Mylopoulos, J.; Chung, L.; and Nixon, B. "Representing and Using Non-Functional Requirements: A Process-Oriented Approach." IEEE Transaction on Software Engineering, Volume 18, Number 6, 1992, pp. 483-497.

Oivo, M; and Basili, V. R. "Representing Software Engineering Models: The TAME Goal Oriented Approach." IEEE Transaction on Software Engineering, Volume 18, Number 10, 1992, pp. 886 -898.

Pohl, K. "The Three Dimensions of Requirements Engineering." Proceedings CAiSE '93, Paris, Springer Verlag, June 1993, pp 275 - 292.

Pohl, K.; and Jarke, M. "Quality Information Systems: Repository Support for Evolving Process Models." Aachener Informatik-Berichte 92-36, RWTH Aachen, Germany.

Ramesh, B.; and Dhar, V. "Supporting Systems Development by Capturing Deliberations During Requirements Engineering." IEEE Transaction on Software Engineering, Volume 18, Number 6, 1992, pp. 498-510.

Ramesh, B.; and Edwards, M. "Issues in the Development of a Requirements Traceability Model." Proceedings IEEE Symposium on Requirements Engineering, San Diego, Ca, January 1993, pp. 256-259.

Rose, T.; Jarke, M.; Gocek, M.; Maltzahn, C.; and Nissen, H. "A Decision-Based Configuration Process Environment." Software Engineering Journal, Volume 6, Number 5, 1991.

Scheer, A.-W. "Architektur integrierter Informationssysteme", Springer Verlag, 1990.

Schmitt, J.-R. "Product Modeling for Requirements Engineering Process Modeling." Proceedings IFIP WG 8.1 Working Conference on Information System Development Process, Como, Italy, September 1993.

Souquières, J.; and Lévy N. "Description of Specification Development." Proceedings IEEE Symposium on Requirements Engineering, San Diego, Ca, Jan. 1993, pp. 16-224.

Suchman, L. A. "Plans and Situated Actions." Cambridge Universiy Press, 1987.

Spivey, J.M. "The Z Notation" Prentice Hall, 1989.

Wing, J. "A Specifier's Introduction to Formal Methods, IEEE Computer, Volume 23, Number 9, 1990, pp. 8-26.

Winograd, T.; and Flores F.; "Understanding Computers and Cognition: A New Foundation for Design." Ablex, Norwood, NJ, 1986.