Goal Driven Business Modelling – Supporting Decision Making within Information Systems Development

Stephan Jacobs
Informatik V
RWTH Aachen
Ahornstr. 55, 52074 Aachen, Germany
jacobs@informatik.rwth-aachen.de

Roland Holten
Institut für Wirtschaftsinformatik
Westfälische Wilhelms-Univerisität Münster
Grevener Str. 91, 48159 Münster, Germany
ISROHO@wi.uni-muenster.de

Abstract

Within information systems development business modelling is often used to structure goal decomposition and goal satisfaction. Business modells serve as a framework for a concrete information systems project. However, the concept of goal is not explicit in the leading business reference models.

In this paper we show how goals can be used to drive the modelling process. Goals are not only used as a starting point of development but serve as criteria to evaluate actions and decisions throughout the design.

We show how business and goal models can be integrated using a common process meta model. An environment to support decision making in the business modelling process has been developed to demonstrate our approach.

1 Introduction

The main objectives of business modelling within information systems development are first, to improve and document the knowledge regarding the current and the desirable situation of the business, second to reach a clear and structured set of documents on goals and concepts of the future business, and third to develop a basis for designing an adequate information system to reach business goals [2]. Business goals are used as a starting point to be satisfied by concrete design decisions, i.e. business modelling follows a top-down strategy.

Business modelling offers detailed schemes about the information system and its context. Different views are proposed to describe the system from several perspectives. Using a common, structured, graphical notation improves the understanding of the goals and concepts and enhances the communication between the participants in the design process.

As these models build a frame to be referenced, they are sometimes called *frameworks* or *reference models*. They guide the transformation of vague ideas into a consistent specification by offering special detailed documents which have to be instantiated.

For example the framework by the IFIP Working Group 8.1 [21, 20], KADS [34] or ARIS [25, 26] all describe different submodels like *functional submodel* or *organizational submodel* which have to be instantiated in phases such as *business analysis* or *system design*.

This work has been funded by the AI-Research Network (Forschungsnetz KI-NRW), North Rhine-Westphalia, Germany

In the following we use the reference model by Österle and Gutzwiller [22] which has been developed as a superset of existing frameworks. This model distinguishes between the analysis and the specification phase. Fundamental components like *business function* or *data store* build the core of the business model. These components are described from different perspectives, e.g. the *functional*, the *communication*, and the *data oriented perspective*.

Some business models focus on the *business process* when analyzing an organization. However, the *development process* of new business models and information systems is described only implicitly. Documents are offered to be instantiated, but the process of instantiation is *not* supported. There is no hint if the forms are filled correctly, there is no suggestion on shortcomings and benefits of the current solution, and there is no guidance what to do next. Thus, business modelling takes a *product* oriented view. It has to be complemented by some kind of *process* oriented support.

According to [4, 24, 10] decision making is a fundamental action in the design process. In the following we concentrate on design decisions within business modelling and information systems development. So far, decisions like *Does the current solution meet all requirements?* are made arbitrarily. Taking a quality oriented point of view *customer satisfaction* has to drive all actions and decisions within the design process [16]. This demands that goals, i.e. the customer wishes, have to be used as *design criteria* through the whole process of business modelling. The role of goals has to change from a starting point of a top-down satisfaction to central criteria driving all decisions within the design process. Goals have to be used to estimate current models, to evaluate single alternatives, and thus help to guide the development process according to the visions on the to be built information system.

In this paper we briefly describe the current way of business modelling and show the benfits of explicit described goals (section 2) We then show how goals can be used to manage information systems development using a *decision oriented process model* which integrates goals and business models (section 3). Based on this model a decision support system is developed to be used in business modelling (section 4).

2 Business Modelling - An Example

Within this chapter we describe the business modelling process using an example enterprise *XprisE* (e*X*ample enter*prisE*) and its processes.

Each of the different business models describes different submodels like *functional submodel* or *organizational submodel* which have to be instantiated in phases such as *business analysis* or *system design*. Subdividing the modelling process in different phases and submodels helps to structure the modelling process into clear pieces.

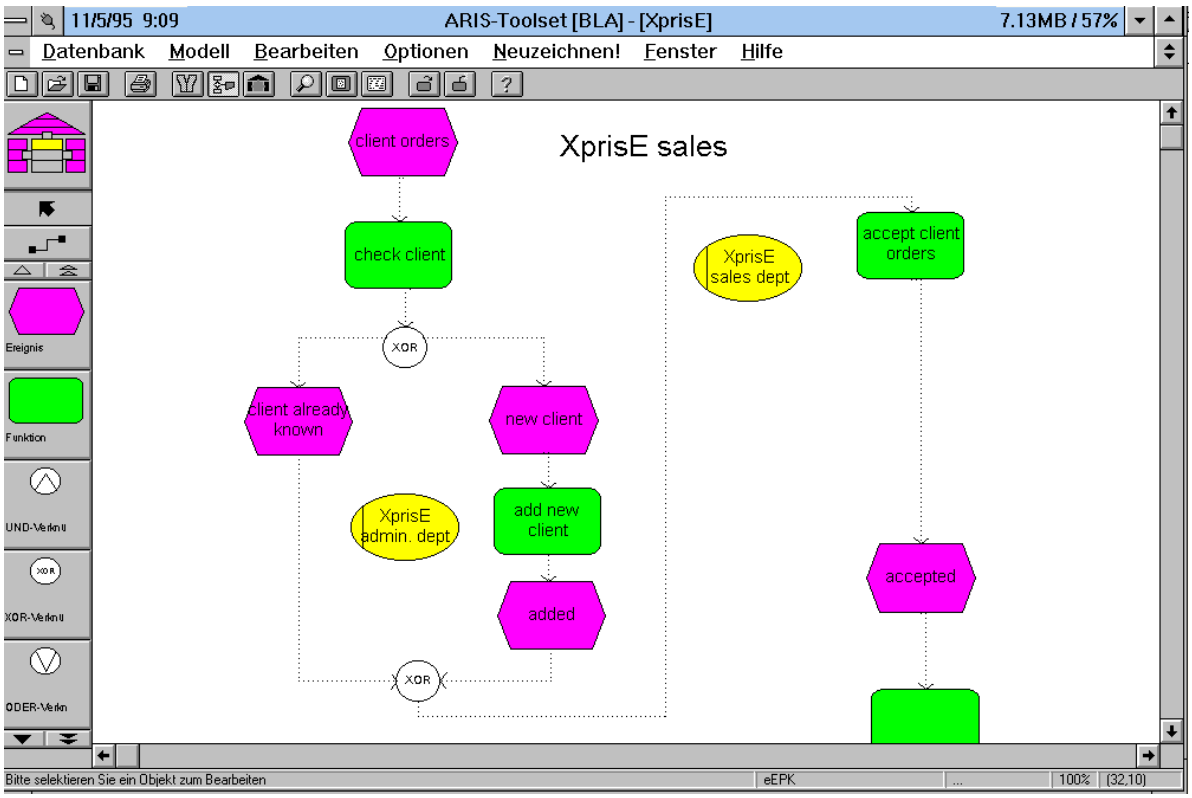

Figure 1: Modelling event-driven process chains with ARIS [27]

2.1 XprisE

In general, business modelling starts with some kind of goal or vision. Visions are typically originated e.g. by troubles ("the shelves are always overloaded"), by changed laws ("hospitals will be paid not longer for the number of patients, but for the number of operations"), envisaged opportunities ("changing our way of working we can increase our profit by 20%"), or other kind of goals. Within our example the management of enterprise XprisE finds out that the sales department does not work well. XprisE's customers complain that orders are lost, customer data are wrong and the like. The management decides to reengineer the department's processes.

After the vision is claimed and the decision is made to reengineer the department the modelling process starts with an analysis of the *As-Is-State*. The current state is described using the different submodels. The processes e.g. are described using *operation chains*. Within this phase shortcomings of current processes often become evident. In the case of XprisE customer orders can only be accepted if the customer's data are already stored in the information system. However, *accepting orders* is performed by the sales department, whereas *adding new customers* is done within the administration department. Not to loose new customers the employees of the sales department accept orders if they get the data which is needed to record the new customer. This data is given to the administration. Within this process typical failures are detected. Often, the customer information is incomplete. The information is not readable, if it is handwritten. The sales department simply forgets to forward the information in time. After all, the registration of the customer and so his order is not accepted.

Within the next step of business modelling the envisaged *To-Be-State* is described. The languages for describing this state are independent of the underlying technology but are (semi-)formalized so that they can be used as a starting point for design decisions. The To-Be-State is modelled within all the submodels. The management of XprisE decides to change the responsibilities for registration of new customers. From now on the sales department is enabled to record new customers. So, if new customers want to order something, one person is responsible for the whole operation including the recording of the new customer.

In the next phase design specifications are made. For example, the user interfaces are defined and requirements for the underlying information systems are described in terms of relations and queries. XprisE's computer department together with some representatives of the sales department defines an interface which automatically asks for all the information demanded to accept a new customer.

2.2 Shortcomings

Having a closer look on today's business modelling practise, it is not really a pure modelling process. Reference models offer standardized solutions for standard business processes like *sale* or *customer registration*. These standards are supported by business modelling tools like e.g. ARIS [25, 26]. Formalized event-driven process chains [27] describing different ways of handling the sales processes can be picked out as a part of the To-Be-State. Thus, business modelling is in a certain sense a *selection process*. Designers can select between several standard solutions. The selection process highlights the decisional aspects in modelling. However, most of the business modelling tools do not support decision making.

As predefined parts of the To-Be-State are picked out in terms of business chains there is one step skipped: *Goals are not explicitly asserted*. The managers of XprisE do no explicitly claim for "customer registration and sales should be in one hand". The goal is directly satisfied by the corresponding business chain. Skipping the goals leads to several shortcomings:

- **Goals do not become clear.** If there are several goals which are all mentioned only implicitly, there is the risk of failing conflicts between goals and ignoring relationships between goals.
- **The rationale for the To-Be-State is missing.** After implementing the new concepts it remains unclear *why* special processes are like they are. As (business) process improvement is an ongoing process, it is crucial to know the rationales behind the design decisions.
- **No chance for validation.** Techniques, originated in the design, like process chains are not well suited for stating requirements and goals [9]. It is difficult to really distinguish between pure requirements and design decisions. As a consequence the final system can only be validated against the process chains but not against the rationales behind them.
- **No decision support:** Customer satisfaction is the final quality goal. Thus, it should be the driving force within the modelling process. Goals could be used as design criteria for the decisions in business modelling, if they were explicit.

2.3 Requirements on Goals in Business Modelling

From the discussion above it should be clear, that

- **Goals have to be made explicit within business modelling.** There has to be an explicit representation of goals. Goals then can be used within decision making. Goals are necessary to keep the modelling process traceable. Goals can be used as design rationales in later phases of business modelling.
- **Goals have to be used within business modelling.** Current business modelling processes have to be reengineered to include the usage of goals. For example, when standard problems are used by selecting a standard solution, the goals can guide the selection process.
- **Goals have to be developed within business modelling:** Beside the different submodels another submodel has to be developed focusing on goals and their interrelationships. Within concrete modelling processes concrete goals have to be developed.

We propose that fulfilling the above requirements is only a little but very effective overhead. Goals are used implicitly within current modelling processes. Thus, it is only a little step in writing them down. On the other hand, goals can be (nearly) automatically integrated as design rationales in the modelling process [12, 15].

3 Integrating Goals and Business Models in a Common Process

The following section introduces the model which integrates business modelling and goals. First, the general framework consisting of four abstraction levels is outlined. Subsequently, a process meta model is described which offers a meta model to describe goals and business models. At last, we show how the single concepts can be integrated into the framework and how business and goal models mutually contribute to each other.

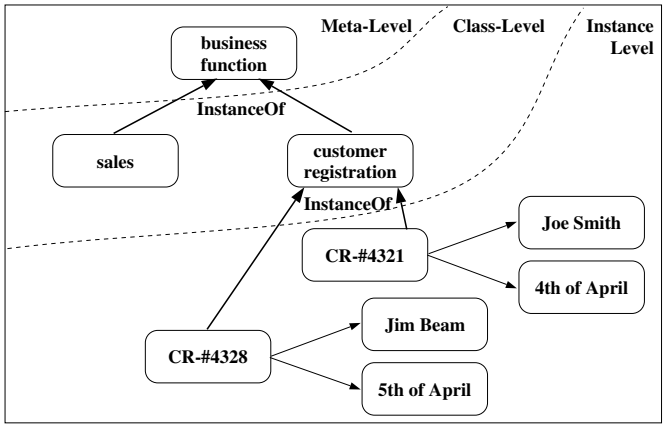

Figure 2: Different abstraction levels within business modelling

3.1 Different Abstraction Levels

Business modelling requires three different abstraction layers, speaking *about* business modelling requires a fourth one.

On an abstract level (meta-level), generic concepts like *event* or *business functions* are defined. These concepts can be instantiated to describe generic processes on a more concrete level (class-level). For example, the XprisE *sales-* and *customer-registration-processes* are developed on this level. Again these generic processes are instantiated by concrete, real-world processes on a concrete level (instance-level). For example, the customer-registration *CR-#4321* of *Joe Smith* on *4th of April* is described here (cf. figure 2).

Two adjoined levels form a so-called *level pair*. Level pairs can be characterized by different kind of processes.

- *Business processes* are performed on the lower level pair, i.e. real world data is stored on the instance-level as an instance of some generic description. These processes are executed by employees of XprisE.
- *Business modelling processes* are performed on the upper level pair by developing e.g. new business functions on the meta-level using concepts from the meta-meta-level. These processes are performed by business engineers.

However, changing the way of modelling, reengineering business modelling, or just speaking about business modelling requires an additional meta-meta level. Now, the meta-meta-level and the meta-level form a supplementary level-pair. The processes on this level-pair are sometimes called *change processes* or *meta-processes*. They are performed by *method engineers*.

Whereas the business engineer uses his notation defined on the meta-level to model business processes, the method engineer needs some kind of onthology to form this notation. This onthology is presented as a *ProcessMetaModell* on the meta-meta level. (Of course, our onthology will describe goals.)

It is important to clearly distinct between the different abstraction levels and the processes performed on the level pairs. The four levels offer a framework which separate different activities on one hand, and show dependencies between theses activities on the other hand. The notion of different abstraction levels in the development of information systems and organizational structures has led to the International Resource Dictionary System (IRDS) Standard (ISO/IEC 10027). In the following, we adopt this framework.

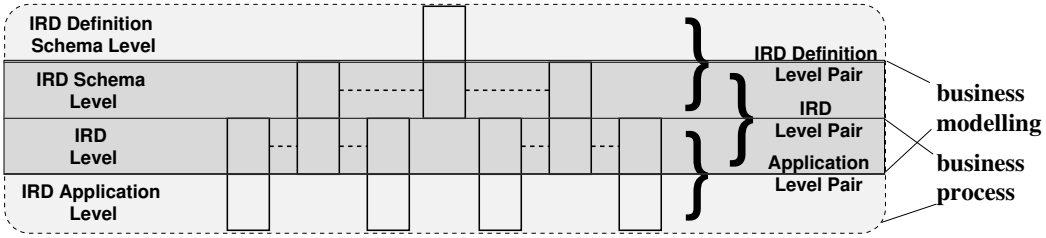

Figure 3: The four data levels and the three interlocking level pairs of the ISO/IEC IRDS framework

Within the IRDS-Standard a specific terminology has been developed. The four levels are called *IRD-Application Level* (instance-level), *IRD-Level* (class-level), *IRD Schema Level* (meta-level), and *IRD Definition Schema Level* (meta-meta-level). The three level pairs are named *Application Level Pair*, *IRD Level Pair*, and *IRD Definition Level Pair* (cf. figure 3).

3.2 The Process Meta Model

In the following we introduce the *Process Meta Model* (PMM) developed in ESPRIT project 6353, NATURE (Novel Approaches to Theories Underlying Requirements Engineering). The model was developed to describe processes in the domain of requirements engineering.

The PMM combines the decision oriented approach with situational aspects following observations by [29] and experiences in the ALECSI project [6]. This means, any product transformation is based on and creates certain situations influencing the development process.

In the following we give a short survey of the model; a detailed description can be found in [23].

The central concepts of the PMM are *intention, product,* and *context* (cf. figure 4).

- *Intention* describes the goals to be reached in a certain situation. It describes the rationale of product transformations.
- *Product* is any kind of document. Speaking of business models the documents describing the different perspectives on the business can be regarded as products. Most often, the concept of situation is solely based on the current state of the products.
- *Context* combines situations and intentions. Here products, their possible transformations and the goals to be reached are put together. There are three specializations of context: *Plan Context, Choice Context* and *Executive Context*.
  Choice context models possible choices the developer can make during the process concerning the former processes direction. Even the definition of new processes is possible. There exist *arguments* which correspond to the possible *alternatives*. *Executive contexts* directly result in transformations of certain products. No choice is possible and the actions are defined exactly. *Plan contexts* define a certain order in a set of any contexts. Here sequences and process abstractions can be defined.

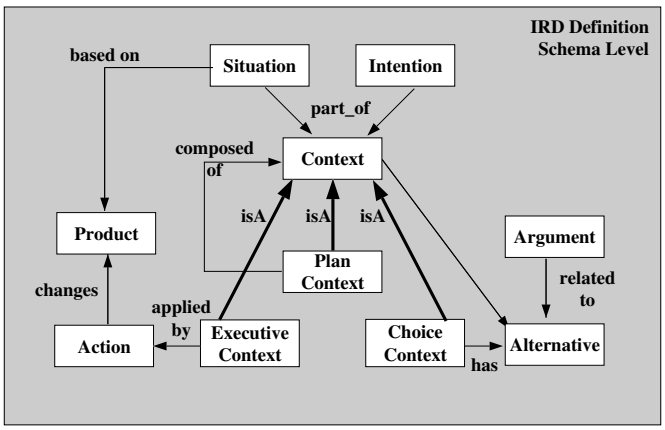

Figure 4: The Process Meta Model by [23]

The core idea of the PMM is to combine the actual product, the envisaged goals, and different kinds of alternatives to proceed. This can be regarded as a general framework which combines performing, planning, and decision making.

The PMM is well suited to serve as a means to integrate business modelling and goals for the following reasons: First, the concepts *product* and *intention* can be used as metaclasses for business models and goal models. Second, the PMM integrates a product oriented view (product) with process orientation (context). Third, different kinds of contexts can be used to describe how goals can drive the process of business modelling. Especially choice contexts can be used to model decisional aspects of design processes. And fourth, the different kinds of contexts can be regarded as different ways to handle goals: The designer refines the intention by choosing one alternative in the choice context. Thus, the right part of figure 4 can be considered as goal refinement. The middle part of the PMM, i.e. the plan context, describes how subgoals can be fulfilled by subcontexts. Thus, the middle part can be regarded as goal decomposition. At last the left part, i.e. the executive context and the action, change the product and by that fulfill the initial vision of the context. Thus, the left part can be called goal satisfaction [18].

3.3 Business Models as Products

As mentioned before the reference models in business modelling serve as frameworks to be instantiated. There is no process support on how to instantiate these models. Therefore, the perspectives are static and have to be instantiated within the PMM as *products*. Figure 5 shows obviously, that only the left part of the whole PMM is used, i.e. business modelling can be regarded as goal satisfaction or product transformation. Basic actions of business modelling like *defining a business function*, can all be regarded as executive contexts.

However, there is no explicit modelling of these actions. There is only an implicit understanding of the development process. Thus, there is no support for general methodical planning of the development process (Plan Context) nor a consideration of decisional aspects (Choice Context).

In figure 5 we have instantiated a part of the functional perspective by [22] as a product into the PMM. Since only a small part of

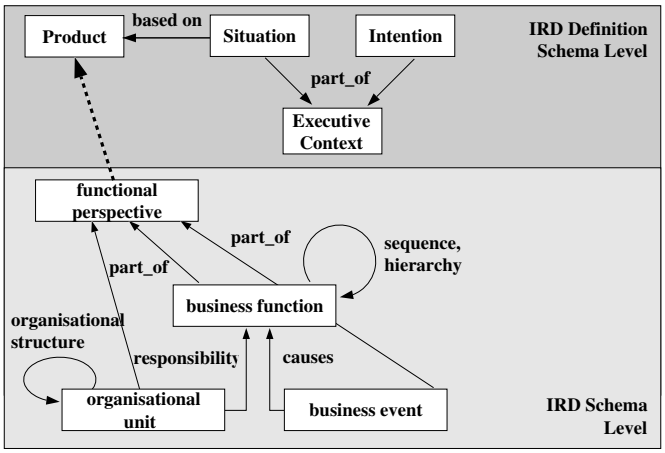

Figure 5: Integration of the functional view by into the PMM

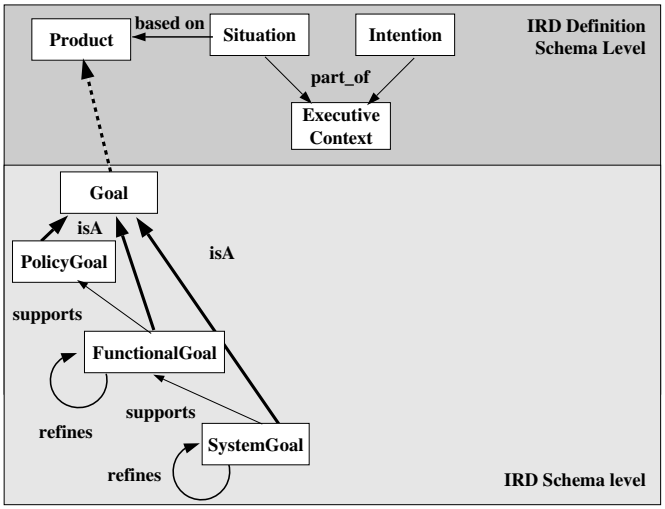

Figure 6: Goal satisfaction as product transformation [31]

the PMM is instantiated, the remaining entities are dropped for simplification. As there is no explicit modelling of goals, the *Intention* is not instantiated.

A description of all perspectives by [22] modelled as instances of the PMM can be found in [8].

3.4 Goals as Products versus Goals as Intentions

Goals are used twofold and thus are instantiated twice in the PMM.

First, goals are stepwise transformed from abstract goals into concrete design decisions. Goals successively satisfy the next goal. Goal satisfaction for special kinds of goals are e.g. described by [3] (accuracy goals, security goals) or [19] (performance goals). A more general approach is followed by [31]. They define three different levels of abstractions, namely *policy goals*, *functional goals*, and *system goals*. However, focusing on transformation of abstract goals into more concrete ones again is a process of product transformation, i.e. goals can be regarded as some kind of product. From a structural point of view, the process of goal satisfaction is very similar to business modelling. Both processes offer some kind of predefined abstraction layers which have to be instantiated during the development process.

Figure 6 looks quite similar to the instantiation of business models as product. Again, there is no instantiation of *Intention* and as business modelling is regarded only as a transformation process there are only *Executive Contexts*.

However, goals are not only the starting point of development processes or some kinds of states during a process of transformation. They describe the objectives to be reached in certain situations, i.e. they describe the rationale of a transformation. Thus, they are instantiated as *intentions* within the PMM, i.e. they can be regarded as part of the *choice context*. More concretely, goals can be used as criteria to evaluate the benefits and shortcomings of contexts in a given situation.

As there are always several, partially conflicting intentions, an action can be validated against *all* intentions, not only the one by which it was initially stimulated. Therefore, it is possible to not only use single goals but complete goal systems. Standards like IEEE, DoD, or ESA, or goal systems like the one by [14] describing different classes of goals like *performance goals* (*efficiency*, *integrity*, ...), *design goals* (*correctness*, *verifiability*, ...), and *adoption goals* (*interoperability*, *portability*, ...) can be used.

In figure 7 we have instantiated a part of the goal system by [33]. This goal system hierarchically describes organizational requirements on the development of information systems. In figure 7 additionally the IRD Level is instantiated with some new goals of the example enterprise *XprisE*.

Suppose, in a first step the above described shortcomings are solved. The envisaged To-Be-State is reached. The sales-department is responsible for the registration of new users. Now, new shortcomings are analyzed. The finance department recognizes that regularly orders are delivered but not paid for. The goal *avoid unpaid orders* is instantiated into the goal model. To tackle this problem the managements decides to proof the creditworthiness of new customers. The first idea is to reengineer the processes so that the new business function *proof creditworthiness* will be performed by the administration department before customers are registered. In traditional business modelling it would no problem to form the next To-Be-State in this way. However, keeping the goals behind the To-Be-State in mind leads to a different situation: the former goal *customer registration and sales should be in one hand* is in conflict with the envisaged proof of creditworthiness by the administration department.

3.5 Driving Business Modelling with Goals

Modelling goals as intentions has two benefits:

- First, the modelling process is not only guided by structural aspects of the business model but by the visions and objectives of the system to be built.
- Second, decisional aspects of the development process can be expressed in the common model. Questions like *Does the current state fulfill all requirements?* or *Do we need another iteration in the development process?* can be answered systematically.

3.5.1 Goal Oriented Modelling

Business modelling focuses on structural aspects, i.e. relationships between different perspectives are defined, common (graphical) notations are offered, etc. However, as business modelling is a transformation process, it is unclear to which degree the result satisfies the initial goals. This phenomenon is well known in

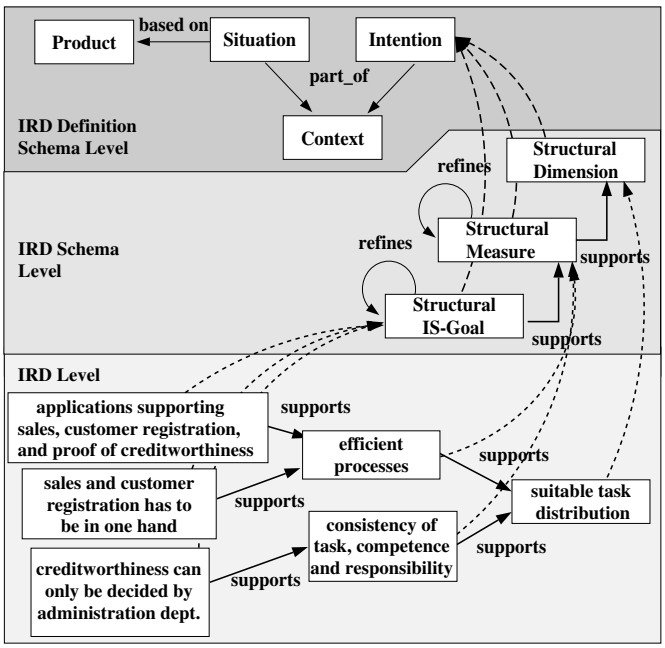

Figure 7: The goal system by [33] instantiated as intention of design decisions

design. Methods like Quality Function Deployment (QFD) [30] are developed to keep the initial goals in mind, to guarantee that customer requirements are integrated into the design [1]. QFD explicitly represents customer wishes to gain a goal oriented process. Thus, defining technical characteristics, fixing parts of the final product, or determination of production steps are all performed according to the customer wishes. [32] emphasizes the notion of explicitly stated goals during the design process to identify a *feasible combination of solutions*.

Focusing only on the framework of the business models leads to a well-structured description of business processes and related information systems. Focusing additionally on explicit goals, leads to well-structured models which satisfy the initial goal of the business modelling process.

Picking up the idea of QFD, actual and former goals can be brought face-to-face with the analyzed As-Is-State. Shortcomings of the current state can be identified. The As-Is-State can be reengineered and transformed into a new To-Be-State. As the two perspectives, the goals and the To-Be-State, are always vis-à-vis, it is possible to immediately validate each design decision against the goals. Shortcomings due to goal conflicts can be minimized. Trade-offs are made explicit.

3.5.2 Supporting Decision Making within Business Modelling

Since the intentions of contexts are explicitly modelled in terms of goal hierarchies, it is possible to evaluate each context, i.e. each product, situation, and action against every goal. Moreover, it is possible to look for goals which are in conflict and goals which mutually support each other.

Thus, besides product transformation another core activity of information system development can be modelled and supported, namely control and evaluation of products and situations.

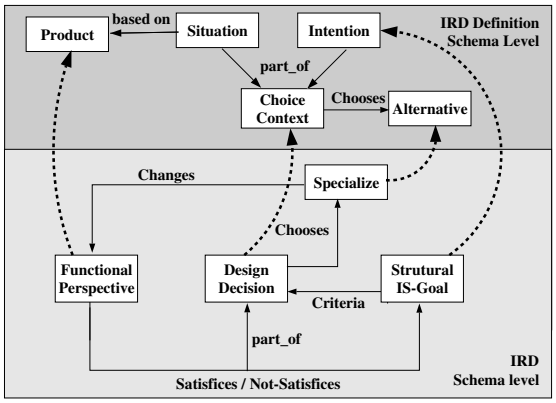

Figure 8: Using goals as criteria of design decisions

Taking the perspective from the PMM, evaluation is a part of the *Choice Context*. Knowledge about the goals, and estimation of products is the foundation of decision making within the modelling process, i.e. deciding how to proceed in the modelling process.

In figure 8 *Design Decision* is represented as an instance of *Choice Context*. The relationship between *Structural IS-Goal* and the product *Functional Perspective* is described by the links *Satisfices* and *Not-Satisfices*. The satisfaction can be regarded as a simple evaluation of the product against the goals. Thus, it can be used as a part of the *Design Decision* how to proceed. In figure 8 the evaluation leads to the decision to *specialize* the current functional perspective.

Business modelling is, like other design processes, no simple top-down process, but can be characterized by iterations and back-tracking moves. Having the possibility to evaluate certain products gives the possibility to systematically support the decisions how to proceed. Moreover, it is possible to support bottom-up processes like reuse or the change of goals within a running project. For example it is possible to analyze an already existing document within a new context. Bringing the new requirements, i.e. the new instance of the goal system, face-to-face with the documents, it is possible to identify shortcomings and benefits, and thus, drive the further process according to this analysis.

In the case of XprisE a design decision has to be made. The two goals *Sale and customer registration has to be in one hand* and *proof creditworthiness*, which ultimately can only be done by the administration department, are in conflict and have to be resolved. No optimal solution is possible which completely fulfills both goals. So, a satisficing solution, a trade-off has to be found. Carefully considering the pros and cons of different solutions the business engineers suggest a solution where employees of the sales department have are free to offer credits up to a certain limit. If this limit is exceeded, a more formal check of creditworthiness is performed by the administration department. The goals serve as criteria within this decision making process.

3.6 Driving Goal Modelling with Business Models

The role of intention and product can be turned around, i.e. the goal system can be built up and made more concrete, using business models as intentions. For example the goal model by [33] can be instantiated on a very abstract level, i.e. there are only goals like *consistency of task and responsibility*. These abstract

goals can be made more concrete using different perspectives of a business model.

A business function like *add new customers* can lead to a refinement of the goal into a design decision like *Creditworthiness can only be decided by departmental head*. When starting business modelling, the goal system which is now more concrete can manage the development process of the information systems.

Summarizing, the principle of mutual dependency, i.e. the goals depend on the system to be built and the system depends on the goals, can be used to drive the development of the information system *and* of the goal system.

4 Tool Support for Business Modelling Decisions

We have introduced a model which demonstrates mutual support of business modelling and goal modelling. Using this idea effectively requires tool support for the following reasons: First, already existing business and goal models are stored electronically. Thus, these models can be reused having computer support. Second, the models offer a huge amount of information. This information can only be visualized from various perspectives in a flexible manner by the means of computers. And third, decision making has to be integrated into the overall business modelling environment to automatically record and trace the decision making process.

Subsequently we present a tool based on our model. We present the overall design decisions, the architecture, and the functionality of our tool.

4.1 Architecture

Decision support in business modelling has to fulfill the following requirements:

- **Computer Support:** Business Modelling produces structured, interrelated documents. These documents are typically stored in some kind of information systems to support the management of the documents. To use these documents automatically the decision support system should be interoperable with business modelling tools. Moreover, the goal models have to be documented and recorded as well. These documents should also be accessible by the decision tool. At last, to guarantee traceability the dependencies between goals and perspectives and between decisions and executed actions should be recorded. These requirement can best be supported by computer technology.
- **Qualitative Decision Support:** Design decisions can be characterized as fuzzy. Simon [28] introduces the term saticficing for decisions that look for satisfactory solutions rather than optimal ones. Thus, quantitative decision support using some multicriteria function to offer a complete evaluation of design alternatives does not seem to be the appropriate means to support design decisions. Qualitative decision support is well suited when dealing with lots of complex and ill-structured information which can hardly be formalized (e.g. the goals). Qualitative decision support means to arrange the foundation of the decision in a way that interconnections, relations, and dependencies become more clear. Because of the chaotic character a simple divide-and-conquer strategy, where single results can be composed to a common decision, will fail.

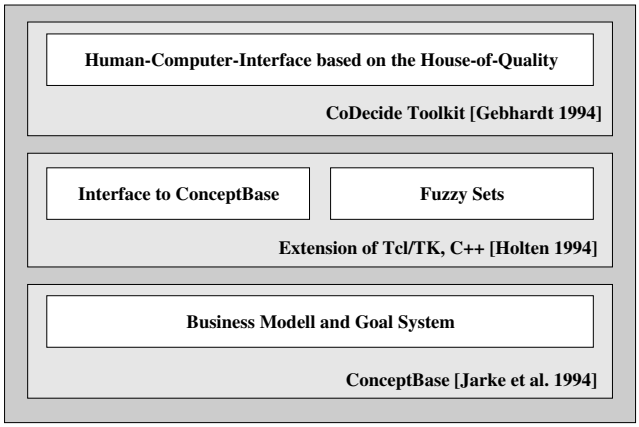

Figure 9: Architecture of a decision support tool within business modelling

- **Visualization:** Bringing several complex perspectives and a criteria-system face-to-face requires a good visualization technique not to get confused about the amount of information. Powerful techniques are required to gain different views: There must be the possibility to (graphically) restructure the information, e.g. change abstract level. Entities have to be grouped and rearranged to enable different perspectives on the information.

These requirements lead to a three-level-architecture of our system (cf. fig. 9). On the basic level there is an information system containing the business and goal models. On the middle level there are fuzzy sets to describe decisional aspects, i.e. to support the satisficing of design criteria by alternative solutions. On the top level there is a sophisticated human-computer-interface based on the House-of-Quality [7], which brings the goal system and the business model face-to-face.

The figure shows how theses layers have been realized with conceptual modelling and user interface generation technique. The following sections elaborate each layer in more detail.

4.2 Conceptual Model Management

Since the prototype shall only demonstrate the ideas of our paper, we did not integrate a goal model into one of the existing business modelling tool sets. We have modelled parts of the reference model by [22] and parts of the goal model by [33] and instantiated it into the meta model by [23].

As a modelling language we chose *O-Telos* [17, 13] implemented in the deductive objectbase manager *ConceptBase* [11]. As Telos offers an infinite number of abstraction levels, it is well suited to represent models based on the IRDS framework. Moreover, Telos presents typical abstraction principles like, inheritance, specialization, or attributes used in the models above. At last, Telos offers the ability to describe relations between objects in terms of constraints and deductive rules. This property can be used, e.g. to guarantee cardinalities between relations.

During the design process of a concrete information system it is possible to develop different perspectives on the system as instances of the business model and special goals as instances of the goal system (cf. fig. 7). The development process results in a data scheme of the information systems. Moreover, there is a description of database transactions. Thus, the model can be used as a prototype where application data can be instantiated on the next abstraction level.

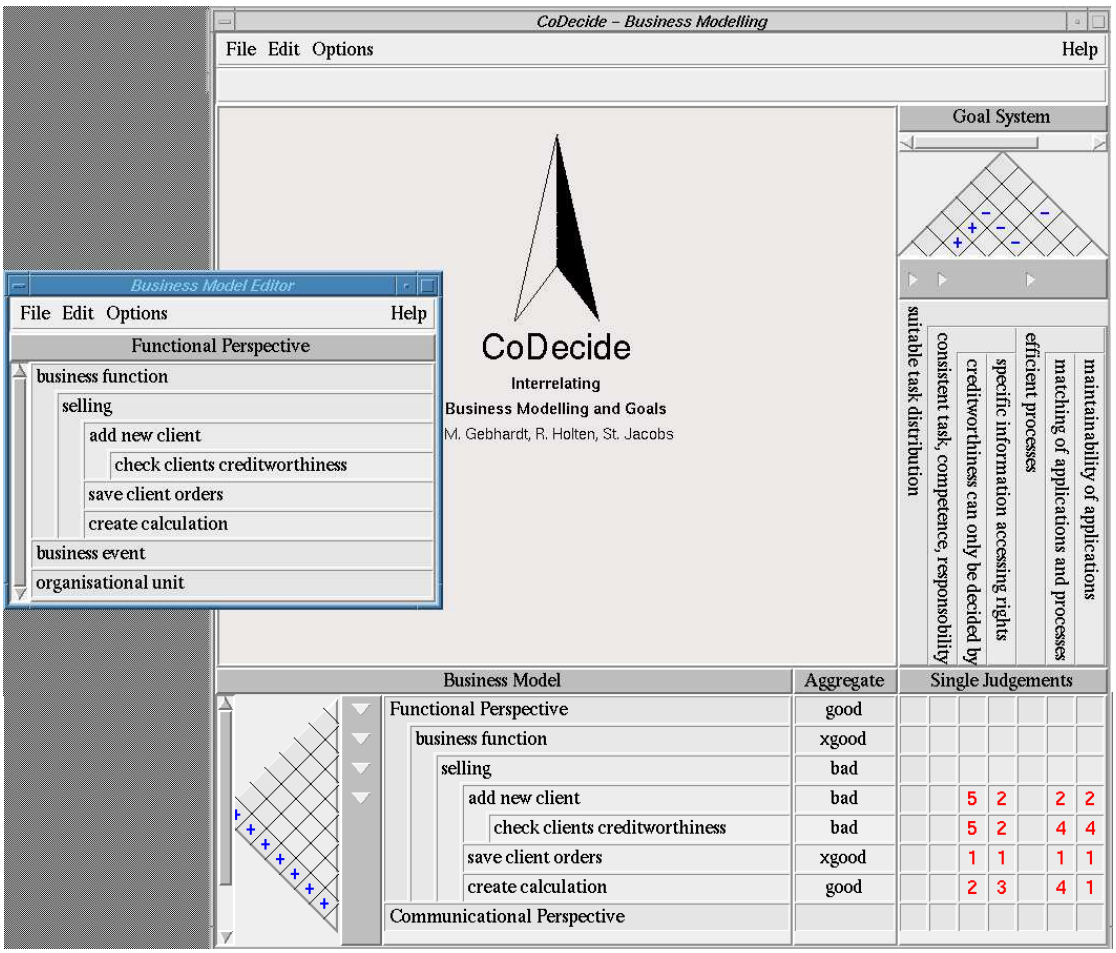

Figure 10: Screendump of our tool to support decision making within business modelling

4.3 Qualitative Decision Support

Qualitative multicriteria support is based on an implementation of fuzzy set aggregation techniques. The situation becomes more difficult when single fuzzy estimations shall be summed into a final valuation. Thus, fuzziness comes from four aspects:

- the estimation of a goal, e.g. to what degree is the goal *efficient processes* satisficed
- the estimation of an object against goals, e.g. to what degree is the goal *suitable task distribution* satificed by the *functional perspective*
- the estimation of an object, e.g. to what degree does the *functional perspective* satisfice any goals
- the estimation of the aggregate of a set of estimations, e.g. what is the final estimation of *suitable task distribution* if the estimations of *efficient processes* and *consistency of task, competence and responsibility* are given.

The last two aspects require the *aggregation* of single estimations. However, according to the fuzziness, it is not well suited to use traditional multiple criteria functions. To describe the degree of satisfaction we used an ordinal scaled function given by the set of ordered labels *xgood, good, mid, bad, xbad*.

We have implemented the aggregation of estimations using *linguistic variables* represented by an ordered set of fuzzy sets [37, 38]. According to the parameters of fuzzy sets and their aggregation we reached a high flexibility concerning the appropriateness of modelling human decision making.

4.4 Visualization of Viewpoints and their Relationships

Based on the House-of-Quality [7] by the method Quality Function Deployment [30] we have developed an interface to represent the models and their instances and bring them face-to-face using the CoDecide-Toolkit [5]. The perspectives of the business model are represented in the horizontal lines and the goals in the vertical ones (cf. fig 10). The horizontal rows build a three-level architecture. The upper two level present the *perspective* of the business model and the belonging *entities*, whereas the lower level consists of *concrete instances*. That means, there are two abstraction levels mixed in the hierarchy. The same holds for the goal hierarchy: The upper two level represent abstract classes of goals, e.g. the dimensions and criteria by [33]. The third level makes these goals more concrete by defining concrete design decisions.

In figure 10 the *functional* and the *communicational perspective* of the business model are shown. The functional perspective is described in more detail by the entities *business function*, *business event*, and *organizational unit*. The term business function is made more concrete by demands on specific functions, namely *add new*

customer, *save client orders*, *check client creditworthiness*, and *create calculation*. Figure 10 represents the dimension *suitable task distribution*. It is divided into the criteria *consistent task, competence, and responsibility* and *efficient processes*. These criteria are decomposed into concrete goals on the information system, e.g. *Creditworthiness can only be decided by departmental head*.

The matrix, which brings the two hierarchies face-to-face, enables to evaluate the single instances of the business model against each goal. The single values are aggregated into a final evaluation. The result is represented in the column left beside the matrix. In figure 10 the rating of the goal *Creditworthiness can only be decided by departmental head* against the business function *check client creditworthiness* is set to *xbad* (visualized by a *5*) as there is no explicit control who performs the check. The various ratings of the above business function are aggregated into the final mark *bad*, which shows that the functional descripiton has still to be improved.

There are two half-matrices formed like triangles. The first one builds the roof of the goals, the other is fixed left beside the perspectives of the business model. Whereas the correlation matrix in the middle enables the user to represent relationships between two kinds of models (the goal and the business model), the half-matrices allow to express relationships within one model. E.g. the goals are not consistent, i.e. there are goals which are in conflict, e.g. performance and costs. The triangles now enable the user to explicitly express positive or negative dependencies in form of positive and negative integers. Thus, additional information is represented which can support design decisions.

In figure 10 the columns *efficient process* and *Creditworthiness can only be decided by departmental head* are related by a "-". The minus shows that there is a negative correlation between the two goals, as the efficiency of processes can be hindered by this kind of restrictions. The "+" between the *functional perspective* and the *communication perspective* displays the consistency between these perspectives in the sense that all *business functions* are modelled in both perspectives.

Our decision support tool is thought to be a contribution to already existing toolsets. The tool supports decision making within business modelling. However, the implementation of the decision, i.e. adding, updating, or deleting of entities in the business model, has to be performed with the normal business modelling toolset. Thus, in figure 10 our decision support tool is shown in combination with a tool to model the functional perspective.

5 Summary and Outlook

Business modelling and goal modelling can be regarded as top-down methods, which successively decompose and satisfy single modelling states. We have introduced an approach which uses goals as a means to drive and manage the business modelling process. Vice versa, goal modelling can be supported by business models.

The business and the goal model are instantiated into a common meta model. Goals are not only used as the source of the whole development process, but as the overall intention of each action during the development process. Thus, the result (product) of each action can be evaluated against goals. Goals are used as criteria to drive the decisions within the development process.

We have developed requirements on a tool to support decision making within goal oriented business modelling. The goals are used as decisional criteria, whereas the single perspectives of the business model are the subject of decisions. We have developed a prototype which combines a sophisticated user-interface to support qualitative decision making founded on a business and a goal model implemented within an information system.

Supporting decisions within the design process, enables to systematically support iterations and backtracking moves which are characteristic for the design. For example, reusing old design and tuning it to a new set of goals can be supported by systematically analyzing the products against the new criteria.

6 References

[1] Akao, Y. *Quality Function Deployment: Integrating Customer Requirements into Product Design*. 1990.

[2] Bubenko, J. A. Extending the Scope of Information Modelling. In *Fourth International Workshop on the Deductive Approach to Information Systems and Databases, Invited Paper, Lloret, Costa Brava, Spain* (9 1993).

[3] Chung, L. *Representing and Using Non-Functional Requirements for Information System Development: A Process-Oriented Approach*. PhD thesis, University of Toronto, 1993.

[4] Dowson, M. Integrated Project Support with IStar. *IEEE Software* (11 1987), 6–15.

[5] Gebhardt, M. Coherent Design through Coupled Views (in German). Master's thesis, RWTH Aachen, 1994.

[6] Grosz, G., and Rolland, C. Using Artificial Intelligence Techniques to Formalize the Information Systems Design Process. In *Proceedings of the Int. Conference on Database and Expert Systems Applications* (1990), pp. 374–380.

[7] Hauser, J., and Clausing, D. The House of Quality. *Harvard Business Review* (5 1988), 63–73.

[8] Holten, R. Integration of Goals and Business Models (in German). Master's thesis, RWTH Aachen, 1994.

[9] Hsia, P., Davis, A., and Kung, D. Status Report: Requirements Engineering. *IEEE Software* (11 1993), 75–79.

[10] Jarke, M. *Database Application Engineering with DAIDA*. Springer-Verlag, 1993.

[11] Jarke, M., Eherer, S., Gallersdörfer, R., Jeusfeld, M., and Staudt, M. ConceptBase - A Deductive Object Base Manager. *International Journal of Intelligent Informations Systems, Special Issue on Advances in Deductive Object-Oriented Databases* (1994).

[12] Jarke, M., Pohl, K., Dömges, R., Jacobs, S., and Nissen, H. W. Requirements Information Management: The NATURE approach. *Ingeniere des systemes d'information 2*, 6 (1994), 609–637.

[13] Jeusfeld, M. *Änderungskontrolle in deduktiven Datenbanken*. Infix, 1992.

[14] Keller, S., Kahn, L., and Panara, R. Specifying Software Quality Requirements with Metrics. *Systems and Software Requirements Engineering - IEEE Computer Society Press - Tutorial* (1990), 145–163.

[15] Klein, M. Capturing Design Rationale in Concurrent Engineering Teams. *Computer 27*, 1 (1 1993), 39–47.

[16] Merli, G. *Total Manufacturing Management*. Productivity Press, Cambridge, 1990.

[17] Mylopoulos, J., Borgida, A., Jarke, M., and Koubarakis, M. TELOS: Representing Knowledge about Information Systems. *ACM Transactions on Information Systems 8*, 4 (1990), 325–362.

[18] Mylopoulos, J., Chung, L., and Nixon, B. Representing and Using Non-Functional Requirements: A Process Oriented Approach. *IEEE Transactions on Software Engineering 18*, 6 (6 1992), 483–497.

[19] Nixon, B. A. Dealing with Performance Requirements During the Development of Information Systems. In *Proceedings of the RE'93, IEEE International Symposium on Requirements Engineering, San Diego, California* (1 1993), pp. 42–49.

[20] Olle, T., Hagelstein, J., Macdonald, I. G., Rolland, C., Sol, H. G., Van Assche, F. J. M., and Verrijn-Stuart, A. A. *Information Systems Methodologies - Second Edition*. Addison-Wesley, 1991.

[21] Olle, W. T., Hagelstein, J., Macdonald, I. G., Rolland, C., Sol, H. G., Van Asche, F. J., and Verrijn-Stuart, A. A. *Information Systems Methodologies*. Addison-Wesley, 1988.

[22] Österle, H., and Gutzwiller, T. *Konzepte angewandter Analyse- und Design-Methoden*. Angewandte Informationstechnik, 1992.

[23] Pohl, K., Dömges, R., and Jarke, M. Decision Oriented Process Modelling. In *Proceedings of the 9th International Software Process Workshop* (1994).

[24] Potts, C. A Generic Model for Representing Design Methods. In *Proceedings of the 11th International Conference on Software Engineering, ICSE, Pittsburgh, PA* (1989), pp. 217–226.

[25] Scheer, A.-W. *Architecture of Integrated Information Systems; Foundation for Business Modelling*. Springer, 1991.

[26] Scheer, A.-W. Architecture of Integrated Information Systems. In *Proceedings of IFIP Working Group 5.3 The Design of Information Infrastructure Systems for Manufacturing; Tokyo; 1993* (1993), G. J. Yoshikawa H, Ed., pp. 177–187.

[27] Scheer, A.-W. *Wirtschaftsinformatik, 4.Auflage*. Springer, 1994.

[28] Simon, H. A. *The Sciences of the Artificial Second Edition*. The MIT Press, 1981.

[29] Suchman, L. A. *Plans and Situated Actions*. Cambridge University Press, 1987.

[30] Sullivan, L. Quality Function Deployment. *Quality Progress* (6 1986), 39–50.

[31] Sutcliffe, A. G., and Maiden, N. A. M. Transforming Informal Requirements: Models for Requirements Elaboration and Refinement. Tech. rep., 1993.

[32] Vriens, D. Problem-Setting Support. In *Proceedings of the IFIP WG A-26 Conference Decision Support in Public Administration* (1993), pp. 29–41.

[33] Welge, M. K. *Managing the Business; vol.2: The Organisation (in German)*. Poeschel, 1987.

[34] Wielinga, B., Schreiber, A., and Breuker, J. KADS: A Modelling Approacch to Knowledge Engineering. In *Readings in Knowledge Acquisition and Learning*, B. Buchanan and D. Wilkins, Eds. Morgan Kaufmann Publishers, 1993, pp. 92–116.

[35] Yu, E., and Mylopoulos, J. An Actor Dependency Model of Organizational Work - With Application to Business Process Reengineering. In *Proceedings of the Conference on Organizational Computing Systems, COOCS'93, Milpitas, CA* (11 1993), pp. 258–268.

[36] Yu, E., and Mylopoulos, J. From E-R to "A-R" - Modelling Strategic Actor Relationships for Business Process Reengineering. In *Proceedings of the 13th International Conference on the Entity-Relationship Approach, ER'94, Business Modelling and Re-Engineering. Manchester, UK* (12 1994), pp. 548–565.

[37] Zadeh, L. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions on System, Man, and Cybernetics 3* (1973), 28–44.

[38] Zimmermann, H.-J. *Fuzzy Set Theory and its Applications (Second Edition)*. Kluwer Academic Publishers, 1991.