Models for Supporting the Redesign of Organizational Work

Eric S. K. Yu
Department of Computer Science, University of Toronto
Toronto, Ontario, Canada M5S 1A4
eric@cs.toronto.edu

Abstract

Many types of models have been proposed for supporting organizational work. In this paper, we consider models that are used for supporting the *redesign* of organizational work. These models are used to help discover opportunities for improvements in organizations, introducing information technologies where appropriate. To support the redesign of organizational work, models are needed for describing work configurations, and for identifying issues, exploring alternatives, and evaluating them. Several approaches are presented and compared. The i^* framework – consisting of the Strategic Dependency and Strategic Rationale models – is discussed in some detail, as it is expressly designed for modelling and redesigning organizational work. We argue that models which view organizational participants as intentional actors with motivations and intents, and abilities and commitments, are needed to provide richer representations of organizational work to support its effective redesign. The redesign of a bank loan operation is used as illustration.

1 Introduction

Today, many organizations are undergoing rapid and fundamental changes. Organizational computing systems are increasingly called upon to play a part in such changes [Keen91] [Tapscott93]. New technical systems are no longer merely used to automate conventional work activities, but are fundamentally changing the way people work [Davenport93] [Hammer93] . Understanding how people work in an organization, and redesigning their work configurations to make innovatives uses of information technology to bring about superior performance is becoming an important prerequisite step to technical system development [Bubenko93] [Jarke94].

The use of appropriate *models* can be of considerable value in helping understand and redesign organizational work. First of all, modelling techniques can be used to highlight the salient features of a particular way of accomplishing work, so that it may be compared to alternate work arrangements. Depending on the capabilities of the model, analytical techniques may be used to determine whether inputs and outputs are matched, whether activities are properly coordinated, or whether incentives are well-placed. Models that are used to describe work arrangements or configurations might be called *work description models.*

Models can also be used to help in the process of identifying objectives and correlated issues, generating and constructing alternatives, making tradeoffs to meet conflicting demands, and managing the decision making process during work redesign. These models might be called *work design models*, and usually have as their subject matter elements from work description models.

In this paper, we examine and compare a number of work description models and work design models. For work description models, we compare SADT [Ross77] (as an example of models that focus on activities and entity-flows), the Action-Workflow model [Medina-Mora92] (as an example of communication-oriented models), and the Strategic Dependency model [Yu93] [Yu94b] (as an example of the modelling of intentional relationships).

For work design models, we compare gIBIS [Conklin88] (as an example of argumentation and design rationale models), KAOS [Dardenne92] (as an example of agent-oriented requirements engineering models and frameworks), and the Strategic Rationale model (as an example of the modelling of intentional actors' rationales.)

The models of the i^* framework (the Strategic Dependency and Strategic Rationale models) are discussed in more detail, as they are designed expressly for modelling and for redesigning organizational work.

Proceedings, Conf. on Organizational Computing Systems (COOCS'95)
August 13-16, 1995, Milpitas, California, USA.

The concept of intentional actor provides a focal point for recognizing the open-ended, distributed, and evolving nature of organizational work. The two models employ compatible concepts and work closely together to support the iterative process of work redesign.

The models discussed in this paper serve quite different purposes than those that are to be embedded in the systems that actually support or execute the work. The latter (which might be called *work support models*, as opposed to the *work design support models* described in this paper) presuppose that a particular mode of support has already been decided upon, e.g., workflow management vs. problem solving support. Understandably, these models tend to focus on features of work particular to that mode of support. In contrast, work *design* support models must be open to various technologies and modes of support. The aim is to find work arrangements that can better address organizational objectives such as better service, lower cost, fewer delays, and so forth. In so doing, one needs to ask why certain work activities are needed at all, and to choose the appropriate technologies to address a variety of needs, demands and concerns from many participants and stakeholders. The resulting work redesigns may contain *requirements* for new (or changes to existing) technical systems.

Although we consider a number of different models in this paper, our aim is not to survey a particular set of models. Rather, our aim is to highlight the need for the two "levels" of modelling – working description modelling, and work design process modelling – in order to support work redesign. Furthermore, we point to the need for compatible modelling concepts between the work description model and the work design model, as illustrated by the *i** framework.

Section 2 describes a scenario for redesigning the credits and loans operation in a bank, to be used for illustrating the various modelling schemes. Section 3 presents and compares SADT, Action-Workflow, and the Strategic Dependency model as examples of models that describe organizational work. Section 4 presents and compares gIBIS, KAOS, and the Strategic Rationale model as examples of models that support the work redesign process. Section 5 concludes by summarizing the contributions of this work and points to future work.

2 A Work Redesign Scenario

We consider the management of credits and loans in a bank.[1] The business of providing loans to customers constitute an important component of the business of the bank. As a result of increasing deregulation in the financial services sector, the bank has the opportunity to expand its credits and loans business.

When a customer is in need of a loan, he approaches the manager of a branch of the bank, and submits an application, with supporting documents. A branch manager's renumeration is tied to the size of her loan portfolio, so it is important for her to attract customers and to maintain good relationships with them. Although this can be a profitable business, there are considerable risks. Thus a loan application must be screened carefully by the branch manager. The credit-worthiness of the customer is then analyzed by a central credit office, and has to be approved by several bank officers, depending on the amount and type of loan.

Despite the care taken during the application process, a loan can still go bad, since the customer's financial conditions will vary. An area coordinator oversees the performance of the loan portfolios of a number of branch managers in a geographic area. To maintain a good performance on her loan portfolio, the branch manager needs to monitor the activities on her loan accounts. One of the indications of bad accounts come from incidences of overdraft. A major daily activity of a branch manager is therefore to scan the overdraft reports to detect potential problem accounts. When a problem account is suspected, the manager will need to retrieve all pertinent information that the bank has about the client, and may also seek to obtain additional information from external sources. The clients may be asked to provide explanations of unusual activities. If the situation warrants, the branch manager may seek to change the terms and conditions of the loan, with the agreement of the customer. If the customer is unable to repay the loan, the manager may refer the case to the legal department to have the money recovered.

What can be done to achieve the desired objectives of increased market share and profitability? There are many possibilities, especially with the availability of various forms of information technology, e.g., workflow management systems, expert systems, shared databases, etc.

To attract customers, one might consider speeding up the loan application process. This can be done in several ways, e.g., one could automate the transmission of documents, (or the shared access to document images), or reduce the number of approval steps, or one could do both. One could give more decision making power to the branch manager, thus achieving a more decentralized operation. This raises some questions: Does the branch manager have the expertise to make good judgements in loans? Can risks still be controlled? If some

[1] This example is largely based on the case study of the operations of an actual bank in Italy [Schael 93].

approval steps are to be omitted, should this be for all applications, or just applications for certain classes of credit, from certain classes of applicants, below certain amount limits? An aggressive approach might be to take riskier loans, and charge higher margins. These loans would have to be monitored more closely. Would that add to the already heavy work load of the branch managers? Would that hurt customer relations or the reputation of the bank? How can these be addressed?

In the next two sections, we examine to what extent existing models can help in describing work configurations, and in providing systematic support in arriving at solutions for redesign.

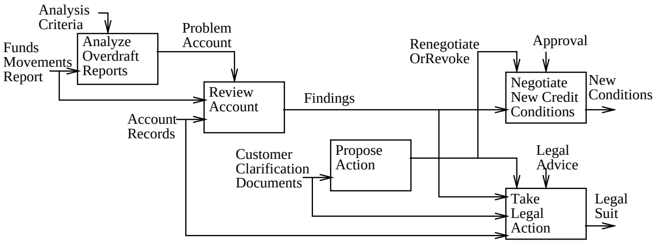

Figure 1: An SADT model of a bank credit management example

3 Models for Describing Organizational Work

Models provide a way of extracting and highlighting certain aspects of the reality being modelled. By using a small number of generic concepts, one focuses attention on certain salient features of the world while deemphasizing others. The choice of modelling concepts in a work description model is therefore crucial in determining what is considered important when comparing different work configurations.

3.1 Models of Activities and Entity Flows

In this class of models, the main focus is on the modelling of activities and how they are inter-related, for example, by the entities that are the inputs and outputs of the activities.

Figure 1 shows an SADT activity diagram of a process for handling problem accounts. Activities are represented as boxes, with input and output entity flows represented as arrows pointing into and out of the boxes. Inputs that are not modified by an activity are called *controls* and appear as arrows pointing into the top of an activity box.[2]

Each activity may be decomposed into sub-activities (in a separate diagram). For example, Review-Account may be decomposed into Retrieve-Loan-Application-Info, Retrieve-Latest-Account-Statement, Compile-History-of-Funds-Movements, etc.

An important analytical aid is to check that all the inputs and outputs are accounted for, e.g., all the flows to and from outside a diagram appear as input/outputs in the corresponding box in the parent diagram, and vice versa. Structured decomposition provides a good abstraction mechanism for hiding or revealing details. SADT (also known as IDEF0) is widely used in analyzing requirements for many types of systems, and is also recently popular as a technique for modelling business processes in business process reengineering.

A limitation of this type of model is that, while it focuses on what activities are supposed to happen, it cannot express why they are supposed to happen. For example: Why do overdraft reports need to be analyzed? Why are explanations and clarification documents from the customer required? And why do changes to the terms and conditions of a loan have to be approved?

From the verbal descriptions of the lending business of the bank, we know that overdrafts provide indications that an account may be in trouble. Since customers' business is valuable to the bank, it is desirable to retain their business whenever possible, while balancing the risks. Thus indications of trouble, such as overdrafts, should be viewed as opportunities for revising the loan conditions, to the satisfaction of the customer. Legal action is a last resort, since it is bad for the customer and also for the bank, and may take a long time to resolve.

This kind of deeper understanding about the operation of the bank cannot be obtained from SADT or similar models, because they cannot convey intentional concepts such as motivations and intentions. Although a higher level activity may provide some hints as to why its sub-activities are needed, it would not convey what other ways for achieving it might be, and what criteria are used to choose among the alternatives.

Another difficulty is the appropriate choice of granularity in modelling [Curtis92]. The model may either appear to be too high-level, and therefore vague, or too detailed, and therefore too constraining. There is no notion of freedom of action by actors.

[2] SADT also provides a dual representation in which entity (data) are inter-related via activities.

3.2 Models of Communication Structures

This class of models focuses on the communication structure inherent in an organizational work setting. Work flows are viewed as forming patterns that fulfill some generic communication needs.

For example, in the Action-Workflow modelling approach [Medina-Mora92] (also called the Business Design Language [Keen91]), the basic unit of modelling is a workflow between a customer and a performer. Each workflow consists of a four-phased loop, representing the stages of proposing, agreeing, performing, and accepting, each involving different types of communicative acts. Subsidiary workflows are seen as originating from, or feeding into one of these phases. Figure 2 shows an Action-Workflow model for handling overdraft in the banking example.[3].

A communication-oriented model provides a higher level description than the activity-entity flow class of models because flows are grouped together according to their contribution to some communicative unit. Also, the identification of a customer-performer pair for each workflow contributes to a notion of who is responsible for what in the work arrangement. This may also suggest the kinds of actions that the performer may take to meet commitment, even if these actions are not all spelt out.

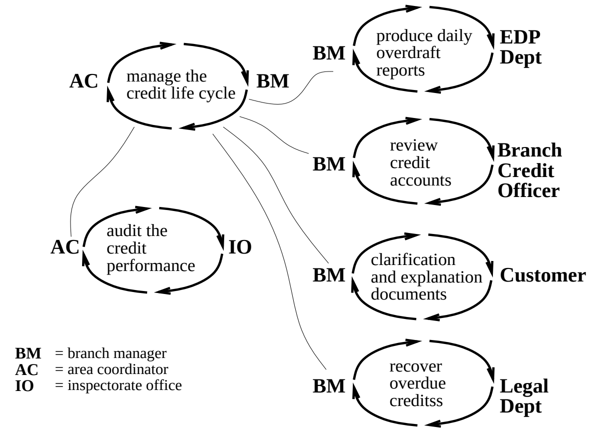

Figure 2: An Action-Workflow model of a bank credit management example

Each workflow loop indicates a commitment by the performer to satisfy the wants of the customer. However, it points to the possibility that customer and performer may need to negotiate the terms of the work, and despite all this, the performer may still fail to meet the customer's expectations.

This kind of model allows analysis based on the communication structure, e.g., whether there are sufficient provisions for clarifying the proposal, or negotiating, and for determining satisfaction. Various types of breakdowns in the workflow can be identified through inspection of the diagrams, e.g., links that are missing.

A major criticism of this class of models has been that they tend to assume (or prescribe) a particular pattern of communication. When analyzing organizational work, it is usually helpful to first obtain an understanding of how the work is actually done. A model that contain a strong prescriptive element would not provide sufficient expressiveness to distinguish between how work is actually done from how it should be done (according to some normative theory).

Another limitation of the Action-Workflow model is that it views workflows only in terms of relationships between two actors. Quite often, although the flow occurs between two actors, expectations (e.g., about the quality of the work) may come from a number of different parties.

3.3 Models of Intentional Relationships

Unlike the above two classes of models, the Strategic Dependency model focuses on intentional relationships among actors, and does not explicitly model workflows nor communication patterns.

In the Strategic Dependency (SD) model,[4] the central concept is that of the *intentional dependency* relationship between actors. In each dependency relationship, a depender actor depends on a dependee actor for something – called a dependum. The dependum can be an activity, an entity, or a condition (described by a statement about the world). The intuitive meaning of a dependency is that if the dependee fails to bring about the condition, perform the task, or furnish the resource, the depender would suffer as a result. Since intentional dependencies have implications on the strategic interests of organizational actors, we also call these strategic dependencies.

Figure 3 shows an SD model of some of the work relationships for managing credit accounts. As in the previous subsections, we focus on the handling of problem accounts. The branch manager depends on daily overdraft reports from EDP. These reports should be accurate and timely. She also depends on credit officer(s) to review problem accounts. When there are questions about an account, the manager depends on the customer to provide necessary explanations and clarification documentation. When an account becomes

[3] adapted from [Schael93]

[4] The Strategic Dependency model was also known as the Actor Dependency model in earlier papers, e.g., [Yu93].

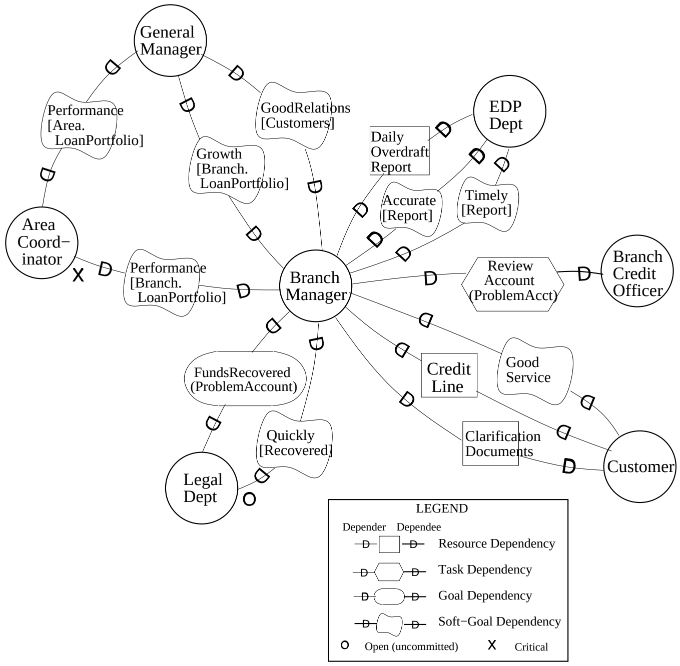

Figure 3: A Strategic Dependency model of a bank credit management example

severely overdue, the manager depends on the legal department of the bank to have the funds recovered. An area coordinator oversees the credit business in a number of branches, and depends on branch managers to achieve good performance in their respective loan portfolios. The bank (as represented by the general manager) also wants the branch manager to expand her loan portfolio, and to maintain good customer relations.

Each dependency can be seen as a matching of a *want* from the depender side to an *ability* on the dependee side. The area coordinator wants to have good performance on a number of credit portfolios, but may not have the ability (or as good an ability) as branch managers (e.g., the latter may know the local business climate and have better local contacts). By depending on branch managers, the area coordinator can achieve good loan performance. By depending on credit officer(s), a branch manager can have many credit accounts reviewed, even if she does not personally have the time to review each one.

The types of dependencies also indicate how the actions of the dependee are constrained by the depender. In a *goal dependency*, since the dependum merely specifies the end-state of actions as a condition in the world, the dependee is free to choose his own courses of action to achieve it. The depender does not care how the condition is achieved. For example, the bank manager is not concerned about how the legal department goes about recovering funds, as long as the funds are recovered. In a *task dependency*, the task description specifies how it is to be accomplished, in terms of subgoals, subtasks, resources, and quality goals (or *softgoals*). The dependee has to follow this course of action. For example, credit officers are expected to review accounts in a certain way. Note that a task specification may still allow considerable freedom since subgoals can be met in different ways. In a *resource dependency*, the dependum is an entity or object, the production of which is not considered to be problematic by the depender. For example, the branch manager's dependency on EDP reports is modelled as a resource dependency. A *softgoal dependency* is similar to a goal dependency, except that the goal involves a concept that is not sharply or objectively defined. The dependee may have different ways for meeting the (soft)goal, but it is the depender who decides which of these are good enough for meeting the goal. The dependencies for good performance, good service, timely and accurate reports are examples of softgoal dependencies in the example.

The model also distinguishes among several degrees of strength of dependency. In an *open dependency*, the depender would like to have the dependum, but if the dependum is not available, the depender would not be affected to a great extent. In a *committed dependency*, if the dependum is not available, some routine course of action in the depender would become unworkable.[5] In a *critical dependency*, the depender believes that all courses of action would become unworkable if the dependum is not available. These dependency types are discussed in more detailed in [Yu93]. An axiomatic characterization of the different types of dependencies are given in [Yu94b].

The modelling of intentional dependencies allow different types of analysis than in activity/entity models or communication-oriented models. Abilities offered by dependees represent opportunities that can be exploited by dependers; but a depender becomes vulnerable to the failure of their dependees. The network of dependencies can therefore be analyzed for opportunities and vulnerabilities. For example, the performance of a branch loan portfolio is likely to be affected if EDP reports are not timely, or if credit reviews are not done, or if bad loans cannot be recovered quickly.

The patterns of dependencies can also be analyzed to see if committed dependencies are likely to be met. For example, the branch manager's dependency on the customer for clarification is likely to be met since the customer depends on the bank for credit. On the other hand, the manager's dependency on EDP for timely and accurate reports may not be met if EDP does not have

[5] The graphical notation takes *committed* to be the default.

any significant dependencies on the branch manager. The concepts of enforcement, assurance, insurance in analyzing dependency networks have been discussed in [Yu94a].

The SD model makes use of knowledge structuring mechanisms developed in conceptual modelling (e.g., [Brodie84]) to organize knowledge. SD models are represented formally using the conceptual modelling language Telos [Mylopoulos90]. Dependums (goals, tasks, etc.), as well as actors, can be organized along the dimensions of classification, generalization, aggregation, and time, thus allowing modelling at various levels of detail. Actors are differentiated into roles, positions, and agents. Agents (such as persons or software systems) *play* roles and *occupy* positions. A position typically *covers* a number of roles. A more careful examination of a work organization typically reveals that intentional dependencies are threaded through agents, roles and positions. Figure 4 shows a fragment of the banking example in terms of agents, roles and positions.

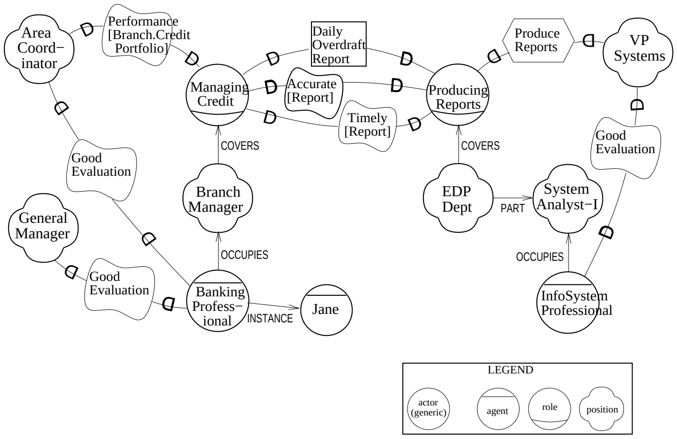

Figure 4: Agents, roles, and positions in the Strategic Dependency model

The modelling of intentional relationships is different from the modelling of activities and entity flows, or of communication structures. Paths of opportunity and vulnerability do not necessarily coincide with workflow or communication structures. Intentional models provide a higher level understanding in the sense that an intentional relationship may embody a number of communicative relationships, and a great many activities and flows. Different communication patterns and flows may be used to fulfill or maintain a given intentional relationship.

Since intentional models do not convey the same picture as the other types of models, they should be viewed as complementary to, but not replacements for the other types of models. Unlike the first two classes of models, models of intentional relationships have yet to be tested in practice on a large scale. Preliminary experiences in using the Strategic Dependency model in practice has been reported in [Briand95].

3.4 Discussion

Each of the above three work description models offers different kinds of basic concepts (or ontology), thus focusing attention on different aspects or dimensions of organizational work. In SADT-like models, the focus is on entities and activities and their inter-relationships. In communication-oriented models, the focus is on pairs of customer-performer relationships, and how commitments are accomplished at several stages. In the intentional relationships model, the focus in on the network of dependencies among actors and their strategic implications.

Each perspective offers a different view, has its own utility, and differs from the others in analytical capabilities. When comparing alternative work configurations during redesign, some differences among alternatives may be more apparent in one type of model than in others. Important differences may be missed if we look at work processes at the "wrong" level. For example, by only modelling, analyzing, and comparing alternatives in activities and entity flows, one may not recognize them as changes in communication patterns and customer-performer relationships. Comparisons based on communication structures may not be recognized as more fundamental shifts in intentional relationships. Although it is possible to enhance organizational performance by only changing activities and entity flows, without reconfiguring intentional relationships, or without even altering communication structures, the more radical and fundamental (and thus high-impact) the changes, the more likely they are to involve changes in intentional relationships.

We also note that different models adopt different approaches for dealing with levels of detail. SADT views details as being hierarchically organized, and can be revealed (or hidden) through structured decomposition (hence *Structured* Analysis). Action-Workflow uses delegation chains as the way to show more detail. Thus, activities that are done by the same person are not decomposed. In the Strategic Dependency model, the use of knowledge structuring mechanisms provides fairly general dimensions for showing or hiding detail. Thus there can be aggregate agents, and roles can have sub-roles. The main focus, however, is not on these decompositions, but on the modelling of the intentional relationships among them. These relationships are not necessarily well-structured; they tend to be more like webs than hierarchies [Kling82]. Work *description* models should provide the facilities for describing the reality as it is, and should avoid being prescriptive.

4 Models for Supporting the Design Process

Although work description models can be invaluable in showing the differences among several proposed work designs, they offer little support for the design process itself. Coming up with a new work configuration that is viable can be a complex process, since one is often faced with large number of design issues and decisions. Models that support the design process can help a designer explore the space of possibilities systematically, weigh pros and cons, and to construct design solutions incrementally. Having a record of this process will also facilitate the revision of designs and their on-going evolution.

4.1 Design Rationale and Argumentation Frameworks

A number of models or frameworks have been developed to support argumentation, or for representing or supporting design reasoning, e.g., [Conklin88] [Potts88] [MacLean91] [Lee92]. This class of models view arguments or design rationales as a network. A number of generic relationships are provided for interrelating them. The contents of the nodes are typically expressed informally in natural language text.

In gIBIS [Conklin88] [Conklin91], there are three types of nodes: *issues*, *positions*, and *arguments*. These may be connected by various link types: a position may *respond to* an issue, an issue may be *supported* or *objected to* by an argument, an issue may be *suggested by* a position, and so forth. Figure 5 shows an example of a gIBIS model of some the arguments (or design rationales) regarding improving the bank loan operation.

One can navigate through the network and examine, replay, revise or augment the arguments. Although the simplicity of the model makes it easy to learn, it is also quite limited in its expressiveness. More recent models have proposed various features to address these shortcomings. However, since these models are not designed specifically for supporting the design of organizational work, they have no special provisions for dealing with this class of applications. References to work design configurations would appear as informal text in the various nodes.

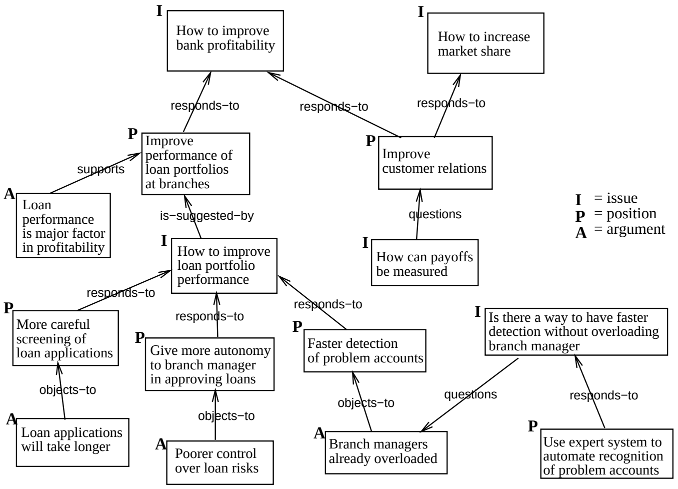
I How to improve bank profitability
I How to increase market share
responds-to
responds-to
responds-to
P Improve performance of loan portfolios at branches
P Improve customer relations
supports
A Loan performance is major factor in profitability
is-suggested-by
questions
I = issue
P = position
A = argument
I How to improve loan portfolio performance
I How can payoffs be measured
responds-to
responds-to
responds-to
P More careful screening of loan applications
P Give more autonomy to branch manager in approving loans
P Faster detection of problem accounts
I Is there a way to have faster detection without overloading branch manager
objects-to
objects-to
objects-to
questions
responds-to
A Loan applications will take longer
A Poorer control over loan risks
A Branch managers already overloaded
P Use expert system to automate recognition of problem accounts

Figure 5: A gIBIS model for work redesign – the banking example

4.2 Agent-Oriented Requirements Engineering Frameworks

In the requirements engineering area, frameworks have been developed that view computer-based systems as components in a larger system that includes humans and hardware and software systems – called composite systems [Feather87] [Fickas92] [Dubois94]. The requirements engineering process starts from global system goals, which are reduced to subgoals in an AND/OR tree. The objective is to reduce goals until they can be assigned as *responsibilities* to *agents*– which are the separate components of the global system.

In KAOS [Dardenne92], goals are expressed informally until they can be operationalized, i.e. expressible as logical constraints. Agents performs actions on objects that they have control over. One of the tasks in requirements engineering is to strength or weaken the conditions on actions so as to *ensure* that the desired constraints are met.

Figure 6 shows a KAOS goal tree for the banking example. That overdrafts be known to a branch manager within one hour is a goal that is precise enough to be operationalized, e.g., expressible as a formula in temporal logic, which can then be ensured, e.g., by implementation in an automated system.

KAOS differs from argumentation and design rationales frameworks in that (among other things) it explicitly manipulates the domain subject matter during the reduction process. For example, it makes use of reduction patterns that involves actions and temporal logic properties.

Where human components of the global systems are concerned, the KAOS approach may appear to be too

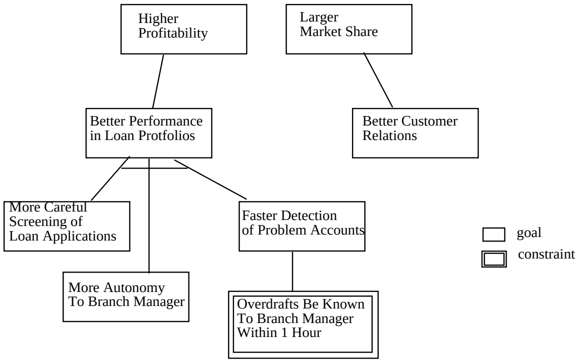
Higher Profitability
Larger Market Share
Better Performance in Loan Protfolios
Better Customer Relations
More Careful Screening of Loan Applications
Faster Detection of Problem Accounts
More Autonomy To Branch Manager
Overdrafts Be Known To Branch Manager Within 1 Hour
goal
constraint

Figure 6: A KAOS model for work redesign – the banking example

rigid, since human behaviour tend to deviate from prescribed actions and strict constraints. The concept of "ensuring" may be more appropriate for the mechanistic components of the global system.

Also, for designing organizational work, it may be hard to identify the boundaries of a "global system", or their top-level goals. AND/OR trees are also quite limiting in dealing with informal goals. A more general treatment of qualitative goals has been developed for dealing with non-functional requirements [Chung93].

4.3 An Intentional Actor Rationale Approach

The Strategic Rationale model is used to model and support the reasoning that organizational actors have about their work relationships. It is represented as a network of nodes and links, as in the preceding two classes of design support frameworks. However, the nodes in the SR model are goals, tasks, resources, and softgoals – the same intentional element types that appear as dependums in the SD model. These elements are linked to each other through means-ends relationships and task-decomposition relationships, representing the reasoning about the intervening connections between an actor's incoming and outgoing dependencies.

Given an outgoing dependency (on a dependee), asking "Why?" (or "What for?") would uncover the *ends* for the existing *means*. Constructing the means-ends hierarchy upwards may eventually lead to some incoming dependencies, i.e., wants and desires that originate from outside an actor. Conversely, starting from an incoming dependency, one could ask "How?" questions to uncover the means used by an actor to meet the ends. Alternatives appear as multiple means for a given end. Task-decomposition links are used to group together the subgoals, subtasks, resources, and softgoals that make up a task specification [Yu94c].

Figure 7 shows a Strategic Rationale model representing some of the reasoning that may be involved in redesigning the bank credit operation. Starting with the branch manager's outgoing dependencies on the EDP department, branch credit officer and so forth, we conclude that these originate from the branch manager's task of handling problem accounts (near centre of figure). This task consists of the subgoal that problem accounts be identified, the subtask of dealing with customers, and a dependency on the branch credit officer to review accounts. (For simplicity, we have omitted the legal department in this figure.) There are also softgoals: that problem identification should be fast, that the work load should not be excessive, and that bad problem accounts should be revoked early to cut losses. The existing means for identifying problem accounts is for the branch manager to manually scan the daily print-outs of overdraft reports from the EDP department. This is a heavy work-load as it takes up one to two hours per day of the branch manager's time. Problem identification is not fast enough as EDP reports may take a few days to reflect customers' transactions at various branches.

New technology offers another means for achieving the goal of having problem accounts identified. The proposal involves delegating problem account detection to an automated event reporting system (through a goal-dependency). The system does this by using expert system technology to recognize patterns of fund movements in customer accounts. While this approach is good for the bank, in that it achieves fast problem identification and at the same time greatly eases the work-load of branch managers, it is bad for the privacy of the customer. The redesign process can explore other alternatives by elaborating on the model, both up and down the means-ends hierarchies (actually a network), and further decomposing task descriptions. The example shows that redesign possibilities may occur within the boundaries of various actors, or may shift responsibilities across actor boundaries, or may involve bypassing certain actors altogether. In any case, the impact of these changes may propagate to a number of stakeholders via dependency links, who may be favourably or adversely affected. At each point, the choice of what further alternatives to consider may be guided by these multitudes of often conflicting goals and strategic interests.

The Strategic Rationale model provides a number of analytical techniques to assist in the process. Given the model of an actor's rationales for its processes, one can analyze them in terms of their *ability*: does the actor have a routine for accomplishing something? – *workability*: are all of its sub-elements workable? – *viability*: are the softgoals sufficiently addressed? – and

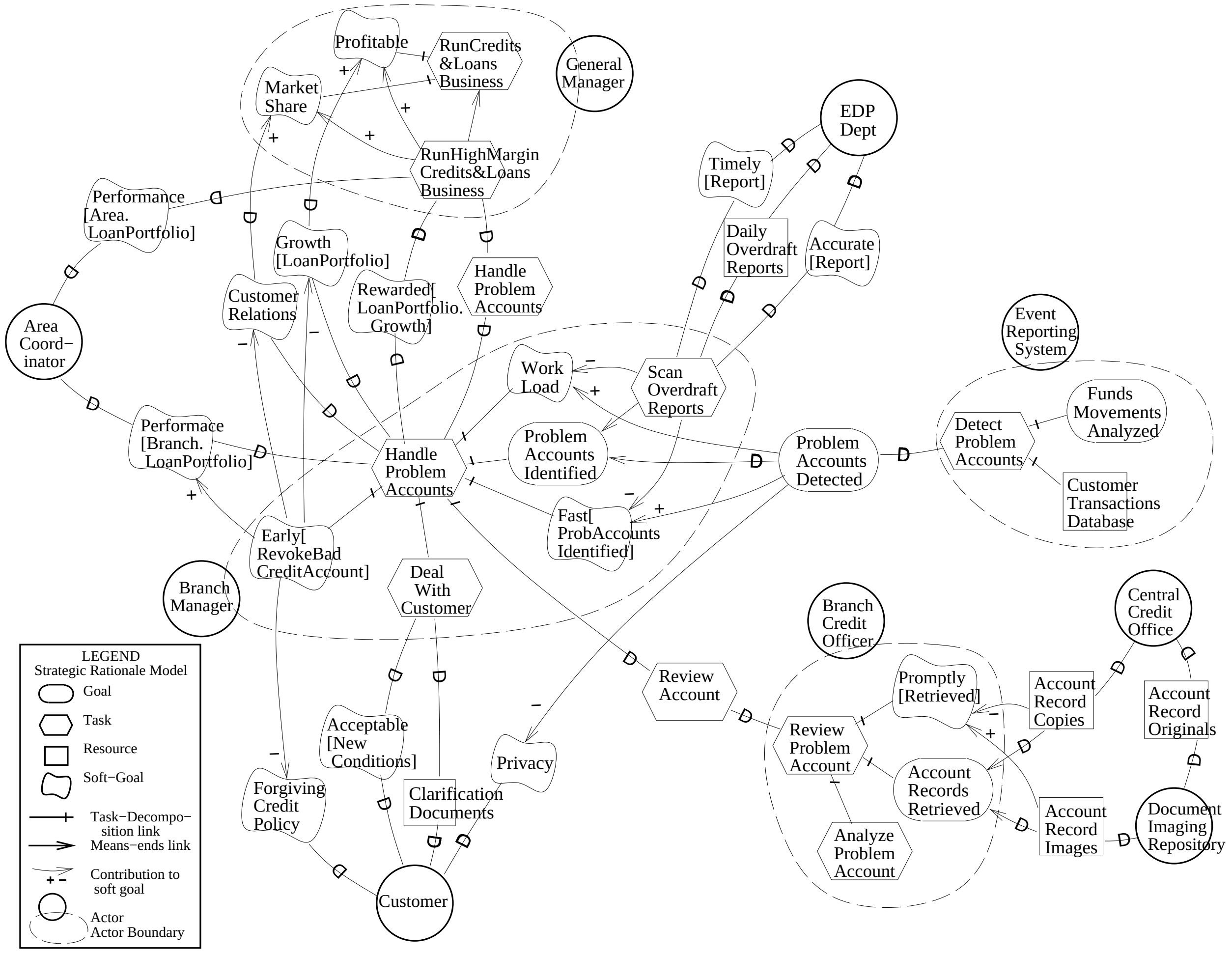

Figure 7: A Strategic Rationale model for work redesign – the banking example

believability: are assumptions adequately justified? The details of these features are presented in [Yu94b].

To assist in the generation of alternatives, and the recognition of correlated issues, the i^* framework makes use of generic *rules* and *methods* [Yu94b]. These techniques are adapted from a framework for dealing with non-functional requirements in software engineering [Mylopoulos92] [Chung93]. Knowledge-based tools can support the construction of and reasoning in the Strategic Rationale model, by drawing on libraries of codified knowledge about information systems and work redesign (e.g., [Turner84]). Despite computer support, the process of work redesign is expected to continue to be highly creative and open-ended, and requiring human judgement throughout.

4.4 Discussion

In models that support the work redesign process, the choice of modelling concepts again has important implications, as in the case of work description models. Argumentation models such as gIBIS provide limited analysis, as they do not make use of concepts specific to work analysis and design. Requirements engineering frameworks such as KAOS offer stronger support since domain concepts, such as agents, actions, and time, are referred to and manipulated during the redesign process. However, the lack of agent autonomy implicit in such frameworks makes them more appropriate for mechanistic agents than for human organizations.

The Strategic Rationale (SR) model differs from the other two classes of design support models in that it incorporates concepts from a work description model – namely the Strategic Dependency (SD) model – and elaborates on them. In the SD model, goals, tasks, resources, and softgoals appear at the "interfaces" between actors, but the internal workings of actors are hidden. Actors are assumed to have their own goals and interests, and pursue them with their knowhow and resources. External dependencies serve as social constraints on their otherwise autonomous behaviour. This approach to modelling organizational work acknowledges that each real-life work situation (each instance of a work process) may require on-the-spot problem solving that takes into account the unique circumstances of the situation, and therefore cannot be fully defined or anticipated in a work process description (e.g., as referred to in [Fikes80]).

In the i^* framework, the description of organizational work in terms of intentional dependencies is seen as constituting a *partial solution* to a set of organizational problems (e.g., those surrounding the handling of problem customer accounts). Each work process instance (e.g., dealing with each particular problem account) involves problem solving (leading to action) that takes the partial solution as a starting point and a guide. In an organizational context, this partial solution serves as a description of the distribution of work, and the expectations that organizational participants have on each other. The redesign of organizational work can therefore be seen as a re-distribution of work, and the reconfiguration of mutual expectations.

Given that an existing organizational configuration can be seen as a partial solution to some set of problems, the redesign process would involve *undoing* the solution, tracing backwards towards the problems, uncovering motivations, intents, and rationales, and then re-generating some other, better, partial solution(s). This process is made complicated by the fact that the partial solution is seldom the solution to a single identifiable problem. Most organizational work processes or configurations are resultants of many interacting forces – the goals, desires, and interests of various parties and stakeholders – and whose influences are threaded through many agents, roles, and positions, criss-crossing the organization.[6] The redesign process is thus necessarily incremental, iterative, and open-ended.

In the Strategic Rationale model, the incorporation of concepts from the work description model (goals, tasks, resources, and softgoals from the Strategic Dependency model) makes it possible to view the work redesign process and the work process as a continuum. Work redesign goals are reduced (or revealed, if proceeding from means to ends) until actor boundaries are reached. The interfaces (as intentional dependencies) between actors become the work process model. Part of the design task is to decide where to place the variable boundary between the work design and the actual work process, i.e., what decisions and choices should be made at "design-time", and what decisions should be left to the "run-time" work process.

As the modelling and redesign of organizational work may involve collecting and using a considerable amount of knowledge, the desirability of computer assistance is apparent. Knowledge representation techniques can be used to help organize, manage, retrieve, and reason with the knowledge [Greenspan94]. Generic knowledge can be organized for reuse. For example, methods for achieving lower cost, faster service, higher quality, better security, greater reliability, etc., can be organized into libraries to assist in generating work redesigns [Yu94b] [Chung93].

As hinted at in the above presentation of the Strategic Dependency and Strategic Rationales models, the use of formal representation techniques need not imply assumptions of rigid mechanistic agent behaviour. The SD model assumes a high degree of agent autonomy and open-endedness. The SR model, though conducive to systematic reasoning and decision making, still relies on human designers to make decisions and judgements. The concept of softgoals provides semi-formal mechanisms (based on qualitative reasoning [Chung93]) for dealing with many important domain concepts that would otherwise have to be treated informally.

5 Conclusions

In this paper, we have argued for the importance of models that support the redesign of organizational work. These models play a different role than models that are used directly in work support systems during their execution.

We have identified the need for two major types of models: models that describe work organizations, and models that support the redesign process. We compared several approaches to each of the two types of models, using a common example. We discussed the importance of the choice of the modelling concepts, the significance of reconciling a work design model and its associated work description model, and the benefits of using formal representations in the modelling.

We also highlighted the potential benefits of models that employ intentional concepts. These intentional models, however, are at an early stage of development. Much further work is need to refine and test these models in practice. Tools and usage methodologies also need to be further developed. Work is also needed to show how these models can be used in the overall context of system development [Jarke92] [Yu95], and for supporting change [Chung95].

Acknowledgements. The author is grateful to Sylvie Nigot for assistance in working out a related banking example, and to Alessandra Agostini and Prof. Giorgio

[6] Hence the name i^*, which stands for *distributed intentionality*.

De Michelis for providing additional information on banking case studies. Prof. J. Mylopoulos provided valuable on-going guidance, support, and encouragement.

Bibliography

[Briand95] L. Briand, W. L. Melo, C. Seaman, and V. Basili, Characterizing and Assessing a Large-Scale Software Maintenance Organization, *Proc. 17th Int. Conf. Software Engineering*, April 1995, Seattle, Washington, USA, pp. 133-143.

[Brodie84] M. L. Brodie, J. Mylopoulos, J. W. Schmidt, eds., *On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases, and Programming Languages*, Springer-Verlag, 1984.

[Bubenko93] J. A. Bubenko, Extending the Scope of Information Modeling, *Proc. 4th Int. Workshop on the Deductive Approach to Information Systems and Databases,* Lloret-Costa Brava, Catalonia, Sept. 20-22, 1993, pp. 73-98.

[Chung93] K. L. Chung, *Representing and Using Non-Functional Requirements for Information System Development: A Process-Oriented Approach*, Ph.D. Thesis, also Tech. Rpt. DKBS-TR-93-1, Dept. of Comp. Sci., Univ. of Toronto, June 1993.

[Chung95] K. L. Chung, B. A. Nixon, E. Yu, Using Non-Functional Requirements to Systematically Support Change, *Second IEEE International Symposium on Requirements Engineering,* York, England, March 1995.

[Conklin88] J. Conklin and M. L. Begeman, gIBIS: A Hypertext Tool for Explanatory Policy Discussions, *ACM Transactions on Office Information Systems*, 6(4), pp. 303-331, 1988.

[Conklin91] E. J. Conklin and K. C. Burgess-Yakemovic, A process-oriented approach to design rationale, *Human-Computer Interaction*, 6(3-4), 1991.

[Curtis92] W. Curtis, M. I. Kellner and J. Over, Process Modelling, *Comm. ACM*, 35 (9), 1992, pp. 75-90.

[Dardenne93] A. Dardenne, A. van Lamsweerde and S. Fickas, Goal-Directed Requirements Acquisition, *Science of Computer Programming*, 20, pp. 3-50, 1993.

[Davenport93] T. H. Davenport, *Process Innovation: Reengineering Work Through Information Technology*, Harvard Business School Press, Boston, Mass., 1993.

[Dubois94] E. Dubois, Ph. Du Bois, F. Dubru, and M. Petit, Agent-Oriented Requirements Engineering – A Case Study Using the ALBERT Language, *Proc. 4th International Working Conference on Dynamic Modelling and Information Systems – DYNMOD'94*, Noordwijkerhout, the Netherlands, Sept. 1994.

[Feather87] M. S. Feather, Language Support for the Specification and Development of Composite Systems, *ACM Trans. Prog. Lang. and Sys.* 9, 2, April 1987, pp. 198-234.

[Fickas92] S. Fickas and R. Helm, Knowledge Representation and Reasoning in the Design of Composite Systems, *IEEE Trans. Soft. Eng.*, 18, 6, June 1992, pp. 470-482.

[Fikes80] R. E. Fikes and D. A. Henderson, On Supporting the Use of Procedures in Office Work, *Proceedings of American Association for Artificial Intelligence*, pp. 202-207, 1980.

[Greenspan94] S. J. Greenspan, J. Mylopoulos, A. Borgida, On Formal Requirements Modeling Languages: RML Revisited, (invited plenary talk), *Proc. 16th Int. Conf. Software Engineering,* May 16-21 1994, Sorrento, Italy, pp. 135-147.

[Hammer93] M. Hammer and J. Champy, *Reengineering the Corporation: A Manifesto for Business Revolution*, HarperBusiness, 1993.

[Jarke92] M. Jarke, J. Mylopoulos, J. W. Schmidt, Y. Vassiliou, DAIDA: An Environment for Evolving Information Systems, *ACM Trans. Information Systems*, vol. 10, no. 1, Jan 1992, pp. 1-50.

[Jarke94] M. Jarke and K. Pohl, Requirements Engineering in the Year 2001: On (Virtually) Managing a Changing Reality, *Workshop on System Requirements: Analysis, Management, and Exploitation*, Schloß Dagstuhl, Saarland, Germany, October 4–7, 1994.

[Keen91] P. Keen, *Shaping the Future: Business Design Through Information Technology*, Harvard Business School Press, Boston, Mass., 1991.

[Kling82] R. Kling, W. Scacchi, The Web of Computing: Computer Technology as Social Organization *Advances in Computing,* vol. 21, 1982, pp. 1-90.

[Lee92] J. Lee, *A Decision Rationale Management System: Capturing, Reusing, and Managing the Reasons for Decisions*, Ph.D. thesis, MIT, 1992.

[MacLean91] A. MacLean, R. Young, V. Bellotti, T. Moran, Questions, Options, and Criteria: Elements of Design Space Analysis, *Human-Computer Interaction*, vol. 6, 1991, pp. 201-250.

[Medina-Mora92] R. Medina-Mora, T. Winograd, R. Flores, and F. Flores, The Action Workflow Approach to Workflow Management Technology, *Conference on Computer Supported Cooperative Work*, Nov. 1992, pp.281-288

[Mylopoulos90] J. Mylopoulos, A. Borgida, M. Jarke, M. Koubarakis, Telos: Representing Knowledge about Information Systems, *ACM Trans. Info. Sys.*, 8 (4), pp. 325-362, October 1990.

[Mylopoulos92] J. Mylopoulos, L. Chung, B. Nixon, Representing and Using Non-Functional Requirements: A Process-Oriented Approach, *IEEE Trans. Soft. Eng.*, 18 (6), June 1992.

[Potts88] C. Potts, G. Bruns, Recording the Reasons for Design Decisions, *Proc. 10th International Conference on Software Engineering* , 1988, pp. 418-427.

[Ross77] D. T. Ross and K. E. Shoman, Structured Analysis for Requirements Definition, *IEEE Trans. Soft. Eng.*, Vol. SE-3, No. 1, Jan. 1977.

[Schael93] T. Schael, B. Zeller, Workflow Management Systems for Financial Services, *Proc. Conf. Organizational Computing Systems (COOCS 93)*, Milpitas, Calif., Nov. 1-4, 1993, pp. 142–153.

[Tapscott93] D. Tapscott, A. Caston, *Paradigm Shift – The New Promise of Information Technology*, McGraw Hill, 1993.

[Turner84] J. Turner, Computer mediated work: the interplay between technology and structured jobs, *Comm. ACM* Vol. 27, Dec. 1984, pp. 1210-17.

[Yu93] E. Yu, J. Mylopoulos, An Actor Dependency Model of Organizational Work – With Application to Business Process Reengineering, *Proc. Conf. Organizational Computing Systems (COOCS 93)*, Milpitas, Calif., Nov. 1-4, 1993, pp. 258-268.

[Yu94a] E. Yu, J. Mylopoulos, Understanding "Why" in Software Process Modelling, Analysis, and Design, *Proc. 16th Int. Conf. Software Engineering*, May 16-21 1994, Sorrento, Italy, pp. 159-168.

[Yu94b] E. Yu, *Modelling Strategic Relationships for Process Reengineering*, Ph.D. Thesis, (also appears as Tech. Report DKBS-TR-94-6, December 1994).

[Yu94c] E. Yu, J. Mylopoulos, From E–R to "A–R" — Modelling Strategic Actor Relationships for Business Process Reengineering, *Entity-Relationship Approach (ER'94) — Business Modelling and Re-Engineering, (Proc. 13th Int. Conf. on the Entity-Relationship Approach, Manchester, U.K., December 1994)*, Springer-Verlag, LNCS-889, pp. 548–565.

[Yu95] E. Yu, P. Du Bois, E. Dubois, J. Mylopoulos, From Organization Models to System Requirements — A "Cooperating Agents" Approach, submitted to the *3rd Int. Conf. on Cooperative Information Systems, CoopIS-95*, Vienna, Austria, May 9–12 1995.