Using Goals, Rules, and Methods to Support Reasoning in Business Process Reengineering*

Eric S. K. Yu and John Mylopoulos
Department of Computer Science, University of Toronto
Toronto, Ontario, Canada M5S 1A4

Abstract

One step towards a more systematic approach to the design of business processes is to develop models that provide appropriate representations of the knowledge that is needed for understanding and for reasoning about business processes. We present a modelling framework which uses *goals, rules,* and *methods* to support the systematic analysis and design of business processes. The framework consists of two main components — a *Strategic Dependency* model that describes a process organization in terms of intentional dependencies among actors, and a *Strategic Rationale* model that supports reasoning during process redesign. Formal representation of these models allows computer-based tools to be developed as extensions to, and eventually integrated with, other tools for supporting information systems development.

1 Introduction

In recent years, the business community has become increasingly aware of the important role that information technology can play in efforts to improve organizational effectiveness. Businesses are beginning to realize that new technology should not be used merely to automate existing processes, but should be used as a basis for reshaping these processes to meet new business realities [14, 9, 34]. Unfortunately, it is generally acknowledged that the practice of "business process reengineering" is still more art than science, and results are often unpredictable. Although some principles and guidelines have been proposed, a systematic framework or methodology has yet to be developed.

One step towards a more systematic approach to the design of business processes is to develop models that provide appropriate *representations of the knowledge* that is needed for understanding and for reasoning about business processes. In the software engineering and information system development areas, researchers have been applying knowledge representation techniques to model information systems and their environments (e.g., [4, 2, 10]). The premise is that, by representing and systematically utilizing the types of knowledge pertinent to each phase of the development process — requirements, design, and implementation — one could achieve a more effective link between rapidly changing organizational needs and the technologies for meeting those needs.

In this paper, we show how a knowledge representation approach can be useful in the context

*Submitted to the *International Journal of Intelligent Systems in Accounting, Finance and Management*, special issue on Artificial Intelligence for Business Process Reengineering. An early version of this paper appeared as [38].

of business process redesign, and in particular, how knowledge represented in terms of *goals, rules,* and *methods* can make the reengineering task more systematic. The presentation in this paper is informal, using examples from the reengineering literature to illustrate the main concepts of the framework.

The framework consists of two main components. The *Strategic Dependency* model describes an organization as a network of interdependencies among actors. Organizational actors depend on each other for goals to be achieved, tasks to be performed, and resources to be furnished. The *Strategic Rationale* model captures the reasoning used to come up with process design alternatives, and arguments about the relative merits of alternative designs with respect to various issues of concern. Goals, rules, and methods provide representations for generic means-ends relationships in the framework.

The technical concepts in the framework adapts and combines ideas from a number of areas: requirements modelling in information systems development (e.g., [13]), agent modelling in artificial intelligence (e.g., [6]), organizational task modelling and work support (e.g., [7]), and frameworks for argumentation, decision and design rationales support, and qualitative reasoning (e.g., [18, 24]).

The paper is organized as follows. Section 2 describes how the intentional structure of a business process is expressed in a Strategic Dependency model, Section 3 describes the use of rules in the Strategic Rationale model to discover process alternatives. Section 4 describes the qualitative reasoning aspect of the Strategic Rationale model, and how methods and correlation rules lead to an argumentation structure. Section 5 explains in greater detail the motivations for the modelling framework, and the types of support tools we expect to be derived from it. Section 6 puts this work in the context of related work. We conclude in section 7 by identifying some future work.

2 Modelling the intentional structure of a business process

Workflow models, which show the flow of work products from one work unit (e.g. a department or a person) to another, are commonly used to describe business processes and for discussing their redesign. Because of their close correspondence to observable entities and activities, workflow models are intuitive and easy to understand. Unfortunately, these models only provide a superficial picture of how an organization operates. The *reasons* that underlie work activities and products are absent in a workflow model. The "Why?" and "What-If?" questions that are central to reengineering [14] are hard to answer with workflow models.

Consider a typical goods acquisition process in a business organization, as represented in the workflow model of Figure 1. This model focuses on capturing the "whats" — what process steps or activities are performed; what work products are produced. Suppose an invoice is missing in accounts payable. Who would be concerned, and why? Who is expected to track down a missing invoice? And why is there a need for invoices in the first place? What if we eliminate ("obliterate") invoices all together, can that be done? And what if we do away with purchasing and let clients order directly from vendors, what issues are involved? Workflow models cannot answer these questions because they do not capture the "whys" — the intentions, motivations, and rationales that underlie the "whats".

To have a deeper understanding of how a business process operates in an organizational context, we need models that capture the *intentional* dimension of organizational work. Organizational members need to be viewed as autonomous actors with desires and abilities. Instead of simply

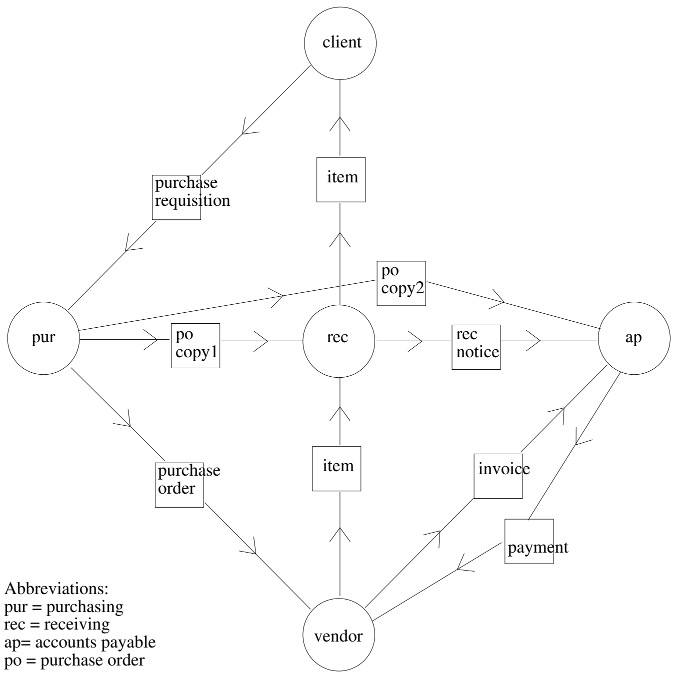

Figure 1: Workflow model of a goods acquisition process

carrying out process steps, they use knowhow and resources to pursue goals. Business processes are accomplished by way of a network of interdependencies among actors: actors depend on each other for *goals* to be achieved, *tasks* to be performed, and *resources* to be furnished.

A **Strategic Dependency** model is a graph, where each node represents an "actor", and each link between two actors indicates that one actor depends on the other for some "object" in order that the former may attain some goal. We call the depending actor the **depender**, and the actor who is depended upon the **dependee**. The object around which the dependency relationship centres is called the **dependum**. By depending on another actor for a dependum, an actor is *able* to achieve goals that it was not able to achieve without the dependency, or not as easily or as well. At the same time, the depender becomes *vulnerable* [21]. If the dependee fails to deliver the dependum, the depender would be adversely affected in its ability to achieve its goals. The strategic implications of an actor's dependencies can be analyzed in terms of the opportunities that they bring, as well as the risks that are associated.

Figure 2 presents a Strategic Dependency model of a goods acquisition process. We distinguish among four types of dependencies. In a **goal-dependency**, the depender depends on the dependee to bring about a certain state in the world. The dependee is free to, and is expected to, make whatever decisions are necessary to achieve the goal (the dependum). The depender does not care how the dependee goes about achieving the goal. The client depends on purchasing for meeting the goal of having an item. Purchasing in turn depends on the vendor to meet the goal of having the item delivered.

In a **task-dependency**, the depender depends on the dependee to carry out an activity. A task-dependency specifies *how* the task is to be performed, but not *why*. The depender makes the decisions. The depender's goals are not given to the dependee. Purchasing depends on Receiving to perform the task of receiving the item. The client's dependency on Accounts Payable to pay

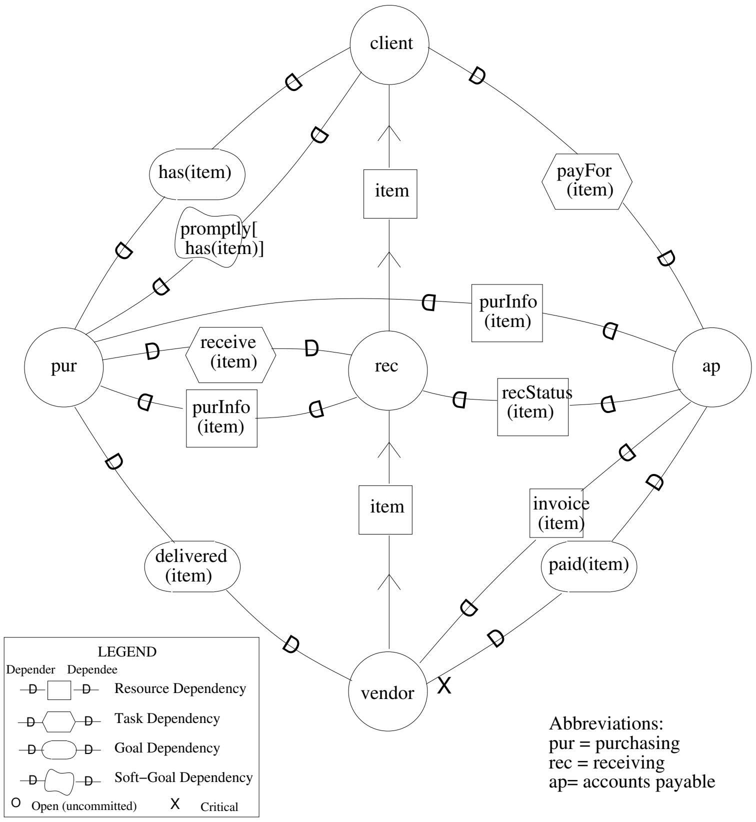

Figure 2: A Strategic Dependency model of a goods acquisition process

for the item is also a task dependency. In each case, the dependee is supposed to follow specific procedures.

In a **resource-dependency**, the depender depends on the dependee for the availability of an entity (physical or informational). Under resource-dependency, the issue of decisions does not come up. A resource is usually the finished product of some deliberation-action process. It is assumed that there are no open issues or decisions to be addressed. The production process is not viewed as problematic by the depender in a resource dependency. There is no decision to be made. For example, purchase information is supposed to be available to receiving and to accounts payable unproblematically.

A **soft-goal-dependency** is similar to a goal dependency, except that the condition to be attained is not sharply defined, but requires clarification between depender and dependee. Typically, the meaning of the goal is specified in terms of the methods that are chosen in the course of pursuing the goal. The depender makes the final decision, but does with so with the benefit of the dependee's knowhow. In our example, the client wants to have the goal "has(item)" met "promptly" by purchasing. The client may decide what choice of vendor and method of delivery would be "promptly" enough in consultation with purchasing.

The model also allows dependencies to have different degrees of strength. Three levels of dependencies are distinguished: Open (uncommitted), Committed, and Critical. These are detailed in [37, 41].

This set of modelling concepts allows us to have a deeper understanding of a business process and the way it is embedded in an organization because at each point in a chain of dependencies, one can infer from the model how the actor's goal-seeking behaviour may be enhanced or restricted, based on the type and strength of the dependencies that it has. It is this deeper knowledge that is necessary to help judge potential targets for obliteration. To answer the question: "What if the Purchasing department is removed from the goods acquisition process?", we observe from the model that the client depends on Purchasing to achieve the goal of having an item, and is therefore vulnerable with respect to this same goal. The client's ability is enhanced through this dependency because the goal can be achieved even if the client does not have the knowhow or the resources to pursue it on his own. To bypass Purchasing, the client would have to acquire the knowhow and have the needed resources (e.g., time and effort) to do purchasing on his own.

3 Discovering Process Alternatives

Since a Strategic Dependency model has explicit representation of goals, one is led to realize that there is more than one way to do things, and to look for alternatives. Hammer has pointed out that organizations often follow rules that are outdated, e.g. pay for an item only when an invoice is received. In this context, a rule has the connotation that it is the only way to do things. One is not aware there are other ways to do things because the goal is not explicit. But if the goals are explicit, then a rule could easily be seen as *one* way of achieving the goal. A rule is a generic link from means to ends. There can be other rules that provide other means to lead to the same ends.

In the Strategic Rationale model, goals and rules are used to express and support explicit means-ends reasoning. A **rule** embodies a generic means-ends relationships, and is expressed by augmenting the attributes in an activity description. In conceptual modelling, activities can be described by using attributes such as input, output, subactivity (decomposition), pre- and post-conditions, and activation and stop conditions [13]. However, this notion of activity is not adequate for means-ends reasoning. We add a **goal** attribute to activity. This attribute will allow the activity to be selected as a candidate for anyone who wants to have this goal met. There can be more than one activity that matches the goal. A means-ends hierarchy results from the decomposition of activities. Instead of decomposing only into subactivities, we allow **subgoals** as well as **subtasks** (following [7]). A subgoal will lead to a search for an activity that matches, whereas a subtask names a particular activity without involving a search. Other attributes include **resource** and **side-effect**, as well as attributes from RML [13]. Side-effects are post-conditions that are not goals.

Figure 3 shows some of the rules that pertain to the buying of an item in the goods acquisition example. The subgoal "ordered(item)" in "buy" matches the goal attribute in "OrderByPhone" as well as that in "OrderByMail", so either of these activities can be applicable. In contrast, the other subpart of the "buy" activity — "payWhenInvoiced(item)" — being a subtask, refers to a particular way of paying.

From the viewpoint of organizing a business process (as opposed to the problem of planning in classical AI, e.g., [25]), the objective is not to decompose or reduce goals into detailed subgoals and subtasks so that we will know how each actor will pursue these goals. Rather, the objective is to find a workable division of labour among organizational members, i.e. a network of intentional dependencies. The kinds of rules that are most relevant for organization design and business process reengineering are therefore those that lead to dependency relationships among actors. For

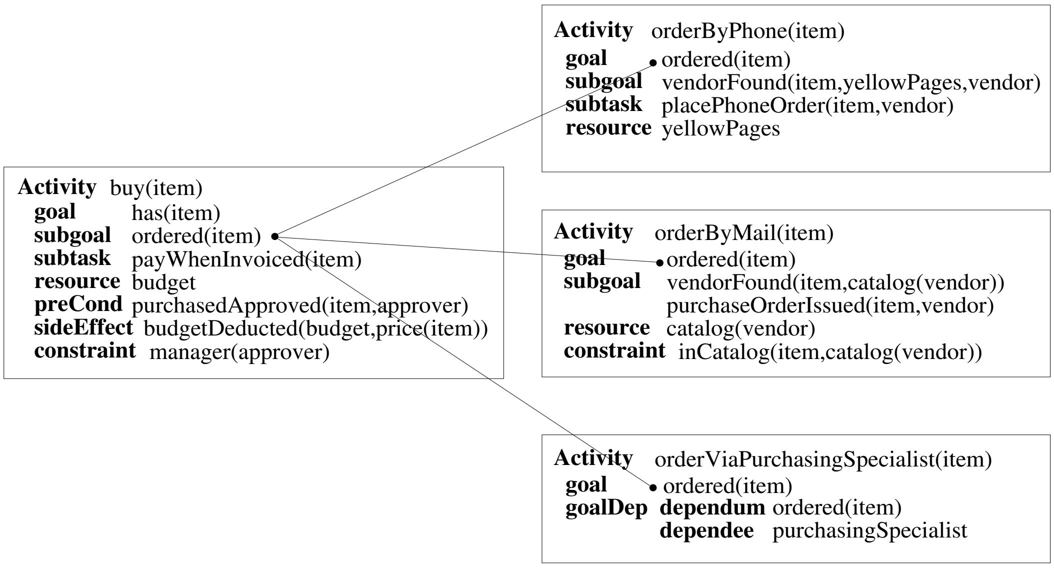

Figure 3: Rules expressed as activity descriptions

example, the third option for achieving the goal "ordered(item)" in Figure 3 is by way of a goal dependency. The activities "orderByPhone" and "orderByMail" are of interest as alternatives to this dependency, suggesting that the actor (the client) might be able to achieve this goal on its own. Further decompositions of these activities are not of interest unless they lead to dependencies, i.e. if the actor (the client, in this example) is not able to perform these activities. For instance, if the client needs to borrow catalogs from the purchasing department, then this should appear as a resource dependency rather than as a resource.

When one has an explicit representation of the intentional dimension of a business process in terms of goals and rules, one could explore new possibilities more systematically, by generating and evaluating alternatives, as well as by coming up with new rules. One way in which new rules can arise is when new technology becomes available offering new abilities.

For example, a new rule might indicate that expert systems technology has the capability to provide the knowhow and resource for simple purchases. This would appear as a fourth option in the example of Figure 3. One company has indeed reengineered its purchasing process in this way [14]. The traditional purchasing process was full of paperwork, errors, and delays. For small purchases, it was not uncommon for the purchasing process to cost more than the item. Now, except for large or strategic orders, company employees would order most items directly from pre-approved vendors through the system without the help of human purchasing agents. If this modelling framework is used, one can systematically explore opportunities for redesigning work using new technology.

The search for new rules can be done systematically along an "IS-A" hierarchy. For example, the approach taken by the Ford Motor Company in its effort to reengineer its goods acquisition process (as described in [14]) may be represented as in Figure 4. A more general version of the rule is first sought, then another specialization is found which achieves the same goal. The current rule for payment is recognized as one way to achieve the goal of "paid(item)". A new rule which eliminates invoices is found. The new rule eliminates the resource dependency on invoice, and simplifies the subgoal (from matching three items to matching two items). It turns out that this is a much simpler operation and can be accomplished mostly by computer.

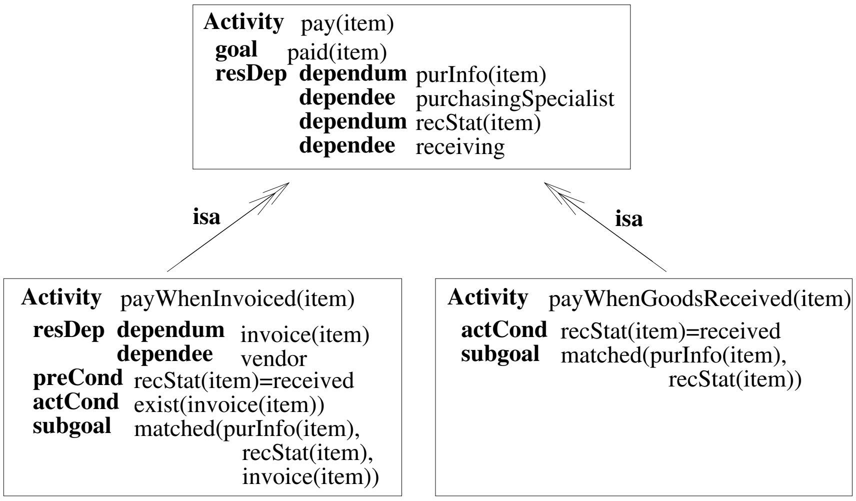

Figure 4: Two payment rules as specializations of a generic payment activity

4 Qualitative reasoning about process redesign

The type of rules presented in the preceding section generates alternatives based on whether a goal can be achieved at all, but there is no representation of how well the goal might be achieved. In designing a business process or organization, there is usually a host of issues and concerns that need to be addressed when choosing among alternatives, e.g. costs, speed, manpower requirements, likelihood of error, quality of service to customers, etc. During the early stages of design, these issues are often handled qualitatively. The ability to represent these design issues and to systematically use them to guide and evaluate design alternatives would greatly facilitate a reengineering effort. We therefore augment the representation of means-ends reasoning in the Strategic Rationale model with qualitative reasoning concepts. We adopt a framework for qualitative reasoning originally developed for dealing with non-functional requirements in software engineering [24, 5].

A Strategic Rationale model is a network of means-ends relationships, representing process design goals and their relationships to (potential) solutions. In the context of organization design, the solutions are organizational configurations, i.e., Strategic Dependency structures. We use the term *issue* to refer to process design goals that are actively being pursued.

Issues are treated as *contributing factors* toward other issues. Contributions can be **positive** or **negative**. The term *satisficed* is used to indicate that a contribution is strong enough to address an issue[1] Figure 5 shows the issues involved when considering replacing the rule "PayWhenInvoiced" with "PayWhenGoodsReceived". An intermediate, "automation" option is represented by the rule associated with the activity "matchByComputer".

The Strategic Rationale model provides a number of link types for differentiating among the different types of relationships that might exist between issues (nodes), so as to support qualitative reasoning. A link may carry four kinds of information:

- the direction of the contribution — from which node to which node. In the graphical notation,

[1] The term *satisficing* was originally used by Herbert Simon [30] to refer to the search for a satisfactory solution rather than an optimal one.

we use arrowheads that point from the solution towards the goal.

- the sense of the contribution — whether the contribution is a positive or a negative one. Graphically, we use a "+" or a "–" annotation adjacent to the arrowhead.
- the extent or degree of the contribution — Following [5], we only distinguish between enough contribution versus not enough (partial) contribution. These are called **sup** and **sub** links in [5], connoting above and below. Graphically we use "Δ" and "∇" (for sup and sub respectively) in conjunction with the "+" and "–" signs.
- how multiple links contribute towards a node — either as a combined effect, called **AND** (graphically a single line arc across the links), or as separate effects, called **OR** (graphically a double line arc).

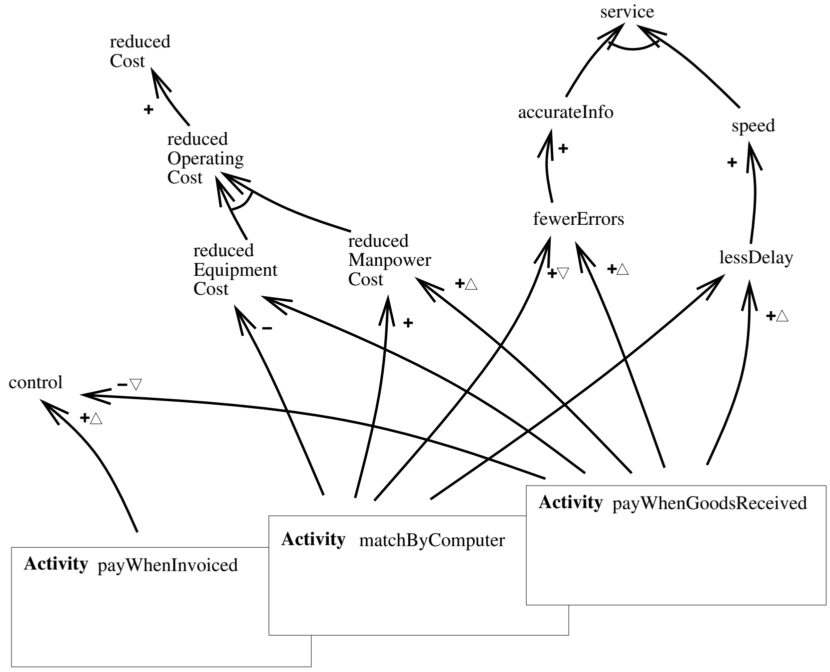

Figure 5: Qualitative reasoning supporting the redesign of a goods acquisition process

A **positive** link indicates that one issue contributes positively to another, but the extent is not known. The example of Figure 5 indicates that using a computer to match invoice to purchase order and receiving document would reduce manpower costs, but whether this reduction is adequate is unknown. A **negative** link represents a negative contribution of unknown extent. Matching by computer would make a negative contribution to the goal of reducing equipment cost.

A **+sub** link represents a positive contribution that is partial in extent, i.e. not enough to say that the goal has been met. Matching by computer is likely to produce fewer errors, but not few enough. This assessment might be supported by the argument that: even though matching by computer is less error-prone, the sources of external error still remain. A **+sup** link indicates a positive contribution that is enough to address the goal. The new payment rule "payWhenGoodsReceived" is judged to adequately address the design issues of delays, errors, and manpower cost. The **–sub** and **–sup** links are defined analogously. For example, invoiceless processing is judged to have a negative impact on control, but not unacceptably so, e.g., regarding auditability.

The **AND** link indicates that several contributions together make enough of a contribution to address a goal. For example, the goal of better service is a combination of faster processing and more accurate information. The **OR** link indicates that each of the individual contributions among several is enough to address a goal. The **und** link type (for undetermined) is used to indicate that one issue affects another in some way, but the sense of the contribution (whether positive or negative) is not known. For example, it is expected that invoiceless processing would affect equipment cost, but it is not known whether there will be an increase or a decrease. Similarly, matching by computer will impact delay, but it is not known whether it will reduce or worsen the delay.

The Strategic Rationale model of Figure 5 is shown in simplified form. In general, each node will have parameters that express further structure, e.g., "accurateInfo" might refer to particular information items or aggregates of items, control can refer to control over particular portions of the organization structure. Some of these parameters will refer to elements in a Strategic Dependency model.

During a design process, the network of arguments will be dynamically constructed. Nodes will have one of four possible values — *satisficed, denied, conflict,* or *unknown* — that indicates the decision status for that node. These values are propagated over the network through a labelling procedure [5].

The links in Figure 5 represent arguments (or means-ends relationships) pertaining to that particular example. Arguments used in a design process, however, often come from generic principles. For example: to reduce manpower costs, try using computers to do the processing; to improve speed, try to reduce or eliminate delays along the process; to achieve better accuracy, try to identify sources of error. The arguments that appear in Figure 5 could be seen as *applications* of these generic principles.

These principles are similar to the rules of section 2, except that (a) they apply to design goals that are about the quality of the process being designed, and (b) they contribute towards, but do not constitute complete solutions for design goals, since design goals are often soft, non-functional goals. Nevertheless these principles can be codified — in the form of **methods** — and used systematically to guide the selection of design alternatives.

For example, in Figure 6, the issue (attribute **softGoal**) "reducedOperatingCost" is decomposed, via a **decompositon method**, into two sub-issues. For each sub-issue, methods are sought whose softGoal attribute matches the sub-issue. In the example, a match is found with the method "reducingManpowerCostByAutomating". This is a **satisficing method** in that it does not simply decompose an issue further into sub-issues, but identifies a structure in a Strategic Dependency model for adoption, i.e. a solution in terms of organizational structure, a component in the business process. The type of link generated by the method is indicated as one of the attribute categories associated with the soft goal (**pos** in this case).

Adopting the activity "matchByComputer", however, has a negative contribution to the goal of reducing equipment cost, according to Figure 5. This kind of correlation from design goal or solution to another design goal is an important type of generic knowledge. These can be encoded as **correlation rules** (Figure 6). A collection of correlation rules forms a matrix that tabulates the cross-impacts among design goals, as done in the Quality Function Deployment technique for quality-driven design [15].

Using specialization ("IS-A") hierarchies, large numbers of methods and rules can be organized

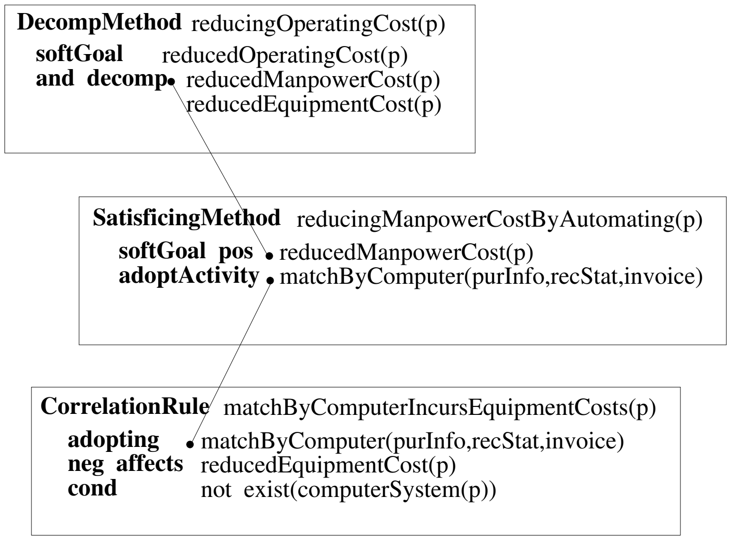

Figure 6: Methods and Correlation Rules

and applied systematically. Additional attributes can further restrict the applicability of rules and methods. For example, the **cond** attribute in the correlation rule in Figure 6 indicates that it is applicable only if there is no computer already in place to support the process p. Other **argumentation methods** are also possible. For example, quantitative or other types of evidence could be included informally to support an argument.

5 Discussion

We have presented the main elements of a framework for modelling organizations in the context of information system development, and have shown its relevance to business process reengineering by using examples from the reengineering literature. In this section, we discuss in more detail the rationales for our approach.

In the early stages of considering information systems for development, there is need to reason about the organization of work, before one develops the technical systems that support the work. This is necessary because there are many ways to provide information technology support. The organization of the near future will likely be populated by a range of technology elements — from passive information repositories to active information agents — complementing the human capabilities of the organization. During this early stage, one needs to be able to describe an organization design, in terms of the relationships among the many human and computer-based elements of the organization. This description, or model, must be able to make the appropriate kinds of conceptual distinctions that are necessary for reasoning about and deciding among alternate designs.

Workflow models present a simplified, and sanitized, procedural view of a business process. They do not reflect the problematic, contingent nature of organizational work [11, 31]. The details of a process — the actual sequence of events as they unfold under a particular set of circumstances — result from the reactions of intentional actors to highly contingent external circumstances, and are therefore ultimately open-ended and unpredictable. Models which attempt to specify a process in terms of detailed steps are unlikely to be faithfully adhered to in reality, and are therefore inappropriate as the basis for analyzing and designing the process.

Nevertheless, even in the face of uncertainty and open-endedness, organizational actors need to be able to predict each others' behaviour to some degree in order to collaborate effectively. In our approach, intentional concepts such as goal, ability, and commitment are used to characterize the *expectations* that actors have of each others' behaviour. By using the concept of intentional dependency, one is able to take a structural view of a business process, without assuming that actors behave deterministically.

In relation to the classical AI problem-solving and planning paradigm (e.g., [25]), we can view a Strategic Dependency structure as a partial solution to business process goals. It specifies a distribution of work among actors, but does not fully reduce the goals, i.e. it may be viewed as a distributed, partial plan. The distribution of work does not dictate specific actions for actors to perform, but involves a network of intentional dependencies among actors. Actors can act freely within the confines of these dependencies. The dependency types in the Strategic Dependency model indicate the various types of freedom and constraints that exist in the relationships among actors. Actors may violate expectations and commitments. The need to infer the consequences of such violations is part of the reason for having an intentional, strategic model of organizational structure.

Rules in the Strategic Rationale model provide a systematic approach for arriving at the partial solution — the Strategic Dependency structure. This type of reasoning focuses on the structural elements needed to achieve a process goal. Finer differentiations among alternatives based on additional quality dimensions that are typical of design reasoning are dealt with by incorporating a qualitative reasoning component. A qualitative, argumentation framework is used because of the typically "non-functional" or "soft" nature of these design goals. The distinctive characteristics of this level of reasoning is the use of methods and correlation rules, which produce contributions towards goals, instead of clear-cut success or failure.

The development of this framework has concentrated so far on the representational aspects. Although the presentation in this paper is informal, a more formal, axiomatic characterization of the framework concepts is provided in [41]. Although we have used a simplified notation for concise presentation in this paper, the knowledge representation language Telos [22] is used as the underlying representation in the framework [41].

A formal approach using knowledge representation concepts opens up opportunities for computer support of the reasoning used in reengineering. Many AI techniques can be adapted for use with this framework. However, computer-based tools will play a primarily supporting role rather than the full automation aimed for in traditional AI. Knowledge about organizations and organization design is typically incomplete, and so will require much human input and judgement at various points. A reengineering effort will involve many interleaved steps of analysis, design, as well as knowledge acquisition and management. As well, general inference procedures based on the proposed models will likely be intractable due to their expressiveness. Specialized algorithms will need to be developed for selected practical reasoning tasks.

Analysis tools that could be developed for the Strategic Dependency model include identification of opportunities and vulnerabilities as implied by a dependency network. Certain properties of interest can be detected, e.g., loops in a chain of intentional dependencies; conflicting goals, opposing tasks, and contention of resources. With additional assumptions about actor behaviour, scenario simulations can be generated.

Rules in a Strategic Rationale model will allow means-ends reasoning to generate alternatives, subject to incompleteness in rule chaining. Conversely, plan recognition techniques can be used to "reverse engineer" from a given structure to uncover possible underlying intentions. Plan critics can be used to recognize common pitfalls in organization design, and suggest alternatives through methods and rules.

Computer support of the qualitative reasoning will allow generation and management of decision graphs, propagating decision statuses across link types and combining them at node junctures. "What-if" scenarios can be explored using design replay support tools. Non-functional design goals and their associated methods will provide a more focused search for alternatives. References from the Strategic Rationale model to the Strategic Dependency model help delineate the scope of relevance of design issues.

Knowledge structuring and management techniques such as classification, generalization, and aggregation hierarchies will facilitate the acquisition, maintenance, and evolution of libraries of cases and generic knowledge, and will encourage reuse.

6 Related work

The organization modelling framework presented in this paper follows a conceptual modelling approach to software engineering and information system development, which emphasizes the need to represent and utilize pertinent knowledge to support each phase of development and on-going evolution [23]. We aim to add to this line of research ([13, 22, 16, 5, 26]) by elaborating on the link between organization redesign and technical system development. Earlier versions of this framework has been presented in [35, 36, 37, 38, 39, 40].

By being part of this larger framework, we take advantage of concepts and tools that have already been developed and implemented (e.g. [16]). A prototype of the NFR assistant for qualitative reasoning has also been implemented [5, 26].

The Strategic Dependency model draws on concepts of dependency from organization theory (e.g. [33, 27]). The formal characterization of dependency types in terms of modal operators for belief, goal, ability, and commitment (as presented in [41]) is an adaptation of intentional models of agents developed in AI (e.g., [6, 32, 19]).

Our concept of a rule is an extension of that of an activity in RML [13], which in turn elaborates on the SADT notion of activity [29]. Activity descriptions are used for means-ends reasoning and planning in [7], by allowing activities to have goals and to decompose into subgoals and/or subtasks.

The Strategic Rationale model adopts the NFR framework of [24] and [5] for dealing with non-functional requirements, which is influenced by Lee's goal-oriented decision support framework [18], which in turn extends previous frameworks for argumentation and for design rationale support (e.g., [28, 20]). This present paper combines functional goals and rules with non-functional goals, methods, and correlation rules into a single framework. The formal relationship between the design level model (primarily non-functional) and the process organization structure level (primarily functional) is inspired by the area of research in AI dealing with the meta-level control of reasoning (e.g., [12, 17]). However, these meta-level computational architectures do not make use of non-functional goals or qualitative reasoning.

In the Requirements Engineering area, the KAOS framework [8] also takes goals of agents in the environment into account, and uses them to lead to what amounts to an organization design and a set of system requirements. However, it does not distinguish process goals (as in our Strategic Dependency model) from process design goals (as in our Strategic Rationale model). It assumes a top-down, design-from-scratch, goal-driven design philosophy, rather than a redesign philosophy. The final design is a set of activities assigned to agents, rather than a network of intentional dependencies among organizational actors.

Goals and rules have also been used in a number of system architectures and models to provide more flexible capabilities in an information system (e.g., [1, 7]). Since these are intended to be models for software systems, their focus is on eventual execution on some machine, rather than on supporting reasoning about the design of organizations and the associated requirements for various types of information system capabilities.

7 Conclusion

In this paper, we have presented some basic elements of a framework for modelling organizations and for reasoning about organization redesign in the context of information system development, and have demonstrated the applicability of the framework to business process reengineering by using examples from the reengineering literature. In particular, we have illustrated how the use of goals, rules, and methods can help in making explicit the reasons that underlie business processes and their design, thus providing a more systematic approach to understanding and designing these processes. It is hoped that the practice of business process reengineering, with the help of the type of framework presented here, will progress more towards an engineering discipline, from its current state as an art.

The development of this framework have so far concentrated on the representational aspects. Practical application of the framework will require the development of algorithms for use with the framework, and further clarification of its semantics. The adequacy of the set of modelling concepts will also need to be tested against a broader range of organizational settings. One possible area for exploration, for example, would be software development organizations and software processes (see, e.g., [39]), where a great deal of interest exists for process modelling and improvement, and where there is active experimentation with a wide variety of new process support technologies. Briand et al. [3] reports on the experiences in applying the Strategic Dependency model to an actual software organization.

References

[1] G. R. Barber, *Office Semantics*, Ph.D. Dissertation, Dept. of Elec. Eng. and Comp. Sci., M.I.T., 1982.

[2] A. Borgida, S. Greenspan, J. Mylopoulos, Knowledge Representation as the Basis for Requirements Specifications, *IEEE Computer*, April 1985, pp. 82-91.

[3] L. Briand, W. L. Melo, C. Seaman, and V. Basili, *Characterizing and Assessing a Large-Scale Software Maintenance Organization,* Technical Report CS-TR-3354, Computer Science Department, Institute for Advanced Computer Studies, University of Maryland, College Park, Maryland, 1994.

[4] J. A. Bubenko, Information Modeling in the Context of System Development, *Proc. IFIP,* pp. 395-411, 1980.

[5] 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.

[6] P. R. Cohen and H. J. Levesque, Intention is Choice with Commitment, *Artif. Intell.,* 42 (3), 1990.

[7] W. B. Croft and L. S. Lefkowitz, A Goal-Based Representation of Office Work, *Office Knowledge: Representation, Management, and Utilization,* W. Lamersdorf (ed.), Elsevier, 1988, pp. 99-124.

[8] A. Dardenne, A. van Lamsweerde, S. Fickas, Goal-directed requirements acquisition, *Science of Computer Programming*, 20, pp. 3-50, 1993.

[9] T. Davenport, J. Short, The New Industrial Engineering: Information Technology and Business Process Redesign, *Sloan Management Review*, Summer 1990, pp. 11-28.

[10] E. Dubois, J. Hagelstein, E. Lahou, F. Ponsaert and A. Rifaut, A Knowledge Representation Language for Requirements Engineering, *Proc. IEEE*, 74 (10), pp. 1431 –1444, Oct. 1986.

[11] 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.

[12] M. R. Genesereth, An Overview of Meta-Level Architecture, *Proc. Third National Conference on Artificial Intelligence, AAAI-83*, Washington, D.C., pp. 119-124, 1983.

[13] S. J. Greenspan, *Requirements Modelling: A Knowledge Representation Approach to Software Requirements Definition,* Ph. D. Thesis, Dept. of Comp. Sci., Univ. of Toronto, 1984.

[14] M. Hammer, Reengineering Work: Don't Automate, Obliterate, *Harvard Business Review*, July-August 1990, pp. 104-112.

[15] J. R. Hauser, D. Clausing, The House of Quality, *Harvard Business Review*, May-June 1988, pp. 63-73.

[16] 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.

[17] B. Kramer, *Control of Reasoning in Knowledge-Based Systems*, Ph. D. thesis, Dept. of Comp. Sci., Univ. of Toronto, 1985.

[18] J. Lee, SIBYL: A Qualitative Decision Management System, *Artificial Intelligence at MIT: Expanding Frontiers*, P. Winston, ed., with S. Shellard, Cambridge, MA: MIT Press, June 1990.

[19] Y. Lesperance, *A Formal Theory of Indexical Knowledge and Action,* Ph.D. Thesis, Dept. of Comp. Sci., Univ. of Toronto, also, Tech. Rept. CSRI-248, Comp. Sys. Res. Inst., Univ. of Toronto, Feb. 1991.

[20] 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.

[21] T. W. Malone, Modeling Coordination in Organizations and Markets, *Management Science*, vol. 33, 1987, pp. 1317-1332.

[22] J. Mylopoulos, A. Borgida, M. Jarke, M. Koubarakis, Telos: Representing Knowledge about Information Systems, *ACM Trans. Info. Sys.*, 8 (4), 1991.

[23] J. Mylopoulos, Representing Knowledge About Information Systems, *Intl. Workshop on Development of Intelligent Information Systems*, Niagara-on-the-Lake, Ontario, Canada, April 21-23, 1991, pp. 94-96.

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

[25] N. Nilsson, *Principles of Artificial Intelligence*, Tioga Press, 1980.

[26] B. Nixon, Dealing with Performance Requirements During the Development of Information Systems, *Proceedings of First IEEE Symposium on Requirements Engineering*, San Diego, Calif., 1993, pp. 42-49.

[27] J. Pfeffer, G. Salancik, *The External Control of Organizations: A Resource Dependence Perspective*, Harper and Row, 1978.

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

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

[30] H. A. Simon, *The Sciences of the Artificial*, 2nd ed., MIT Press, 1981.

[31] L. Suchman, Office Procedures as Practical Action: Models of Work and System Design, *ACM Trans. Office Information Systems*, vol. 1, no. 4, October 1983, pp. 320-328.

[32] B. Thomas, Y. Shoham, A. Schwartz, and S. Kraus, Preliminary Thoughts on an Agent Description Language, *Intl. J. Intell. Sys.*, Vol. 6, 1991, pp. 498-508.

[33] J. D. Thompson, *Organizations in Action,* McGraw-Hill, 1967.

[34] N. Venkatraman, IT-Induced Business Reconfiguration, *The Corporation of the 1990's — Information Technology and Organizational Transformation*, M. Scott Morton, ed., 1991, pp. 122-158.

[35] E. Yu, Modelling Organizations for Information Systems Requirements Engineering, *Proceedings of First IEEE Symposium on Requirements Engineering*, San Diego, Calif., 1993, pp. 34-41.

[36] E. Yu, An Organization Modelling Framework for Multi-Perspective Information System Design, *Requirements Engineering 1993 — Selected Papers*, J. Mylopoulos et al., eds., Tech. Rpt. DKBS-TR-93-2, Dept. of Comp. Sci., Univ. of Toronto, July 1993.

[37] E. Yu, J. Mylopoulos, An Actor Dependency Model of Organizational Work — With Application to Business Process Reengineering, *Proc. Conference on Organizational Computing Systems (COOCS 93)*, Milpitas, Calif., Nov. 1-4, 1993, to appear.

[38] E. Yu, J. Mylopoulos, Using Goals, Rules, and Methods To Support Reasoning in Business Process Reengineering, *Proc. 27th Hawaii Int. Conf. System Sciences*, Maui, Hawaii, Jan. 4-7, 1994, vol. IV, pp. 234-243.

[39] 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.

[40] E. Yu, J. Mylopoulos, From E-R to "A-R" — Modelling Strategic Actor Relationships for Business Process Reengineering, *Proc. 13th Int. Conf. Entity-Relationship Approach*, December 13-16 1994, Manchester, U.K., to appear.

[41] E. Yu, *Modelling Strategic Relationships for Process Reengineering*, Ph.D. Thesis, Dept. of Computer Science, Univ. of Toronto, forthcoming.