1

An Organization Modelling Framework for Information Systems Requirements Engineering

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

Extended Abstract

Information technology is becoming increasingly crucial to the viability and competitiveness of business and other organizations (e.g., [Keen91] [Davenport93]). Yet the development and evolution of information systems remain difficult challenges. High costs, long lead times, and the inflexibility of systems are still problems for technical system developers.

A major source of these problems is the difficulty in understanding the business and organizational environment in such a way as to make it possible to systematically relate that understanding to decisions in technical system development.

In the emerging field of *requirements engineering*, researchers have proposed models based on formal concepts to describe not just the technical system, but the environment within which the system operates (e.g., [Zave81] [Borgida85] [Dubois86] [Dardenne93]). These models can be used to obtain a more precise and deeper understanding of the environment and to relate them to technical system development.

One deficiency of most requirements models to date is that they do not deal adequately with the *intentional* dimension, i.e., the motivations, intents, or rationales that underlie the way an organization operates. The need to ask "Why?" questions has been well argued in the recent literature on business process redesign (e.g. [Hammer90]). Without a clear understanding of the rationales behind existing practices and structures, one could not easily decide what changes could be made to business processes. By discovering underlying reasons, one can more readily identify outdated practices, and replace them with information technology systems and work arrangements that reflect new realities.

In this paper, we outline a modelling framework which deals explicitly with the intentional dimension in order to help understand and redesign organizations and business processes in the context of IS development. The basic framework consists of a series of four models, progressively revealing the intentional content behind a work organization.

The **Workflow** (WF) model describes what *actors* take what *entities* as *input* and produce what entities as *output*. Figure 1 shows a Workflow model of a (simplified) example process from the health insurance domain, in which treatment must be pre-approved by the insurer in order for the physician to receive reimbursement. The claims manager issues approval upon verifying

*to appear in *Proceedings of the Third Workshop on Information Technologies and Systems (WITS'93)*, Orlando, Florida, U.S.A., December 4–5, 1993.

the applicability of the patient's policy to his medical condition, and upon confirming that the treatment plan is appropriate according to medical opinion.

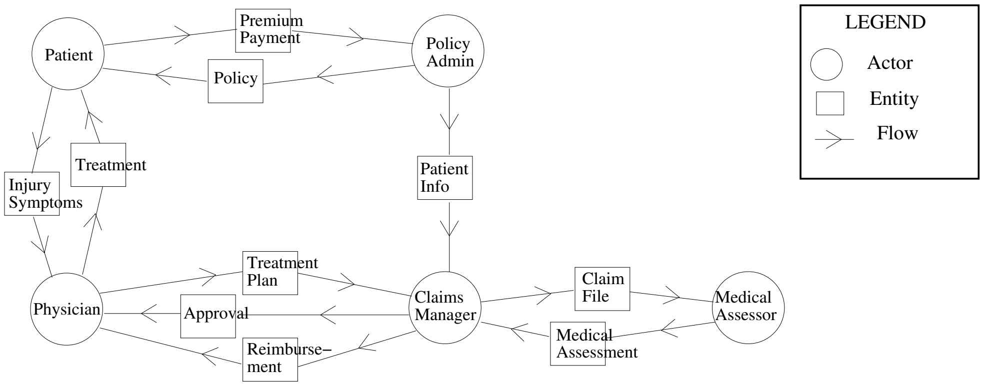

Figure 1: Workflow Model (WF).

The graphical representation of workflow in Figure 1 highlights the flow of entities from actor to actor. In the formal representation, each actor may perform multiple *activities*. Entities flow from activity to activity. Further conditions and properties, such as *pre-conditions* and temporal or other *constraints*, can be attached to activity and entity descriptions. The features of the Workflow model are based on RML [Greenspan84] (which is based in turn on SADT [Ross77]). A WF model can be analyzed for completeness and consistency of input/output and other properties. Violations of temporal precedence and other constraints can also be detected.

This level of modelling presents a non-intentional description of a work process – it can answer "what" questions, but not "why" questions. Nevertheless, a WF model provides an important reference point for asking "why" questions.

The **Actor Dependency** (AD) model provides a first step for making the intentional content behind the practices of an organization explicit. Actors are taken to be goal-oriented. Instead of rigidly following pre-determined courses of action, actors use knowhow and resources to pursue goals. Since actors in organizations often do not have all the knowhow and resources needed for achieving many of their goals, they depend on each other, resulting in a network of dependencies that might be called the *intentional structure* of the organization. Four types of dependencies are used to differentiate the relationships between depender and dependee. In a *goal dependency*, the depender depends on the dependee to bring about a certain state or condition in the world, without constraining how the dependee would go about achieving it. In a *task dependency*, the depender specifies how the task is to be performed by the dependee, without necessarily making the goal explicit. In a *resource dependency*, the depender depends on the dependee for the availability of an entity (physical or informational). 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 in terms of the methods that are used in the course of pursuing the goal.

In the insurance example (Figure 2), the patient depends on the physician for his injury to be treated (a goal dependency). The physician depends on a laboratory to perform tests (a task dependency), and on pre-approval from the insurer in order to be reimbursed subsequently (a resource dependency). The patient's desire for good service, and the policy administration department's desire for cost-effective treatment and fast claims approval are modelled as soft-goal dependencies. The notion of soft-goal is based on an approach for dealing with non-functional requirements in software engineering, which is also the basis for the NFR model, to be introduced later in the paper.

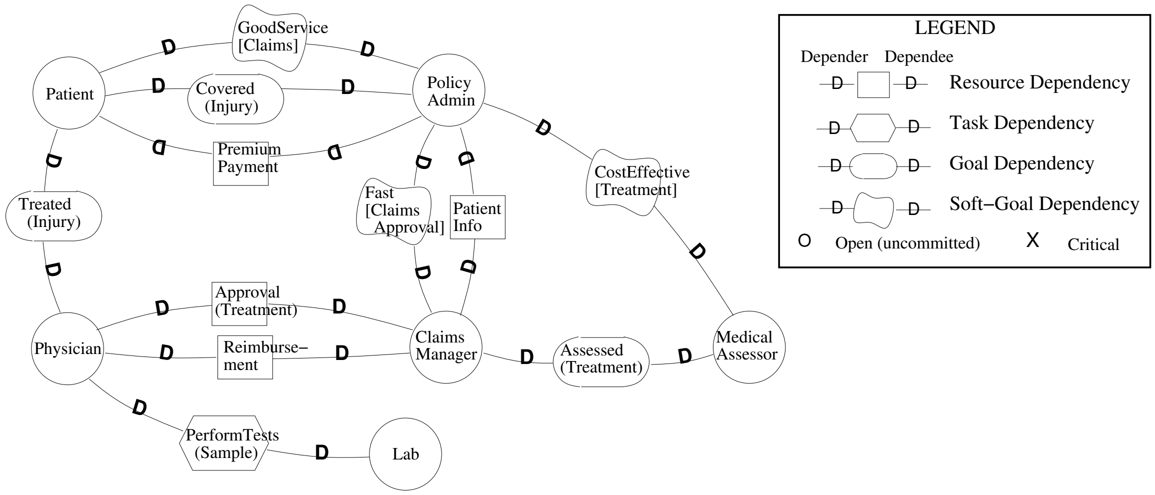

Figure 2: Actor Dependency Model (AD).

Dependency relationships offer insights into the motivations and rationales behind the superficial, non-intentional structure of an organization or process. For instance, why would a patient seek treatment from a physician? – because he wants to have his injury treated, but does not have the medical knowhow to treat himself. Why does a physician depend on a lab to perform tests? – because he needs to treat patients, but lack the resources (e.g., time, specialized equipment) to perform lab tests.

An Actor Dependency model permits different types of analyses than in a Workflow model. A dependency expands an actor's capabilities, so that it is able to achieve what it is otherwise unable to achieve. At the same time, an actor becomes vulnerable when it depends on others [Malone87]. By following the chains of dependencies in an AD model, one could obtain an understanding of the range of possibilities that are open to an actor, as well as the potential for failure arising from these dependencies. Actors attempt to mitigate their vulnerabilities by establishing mechanisms for enforcement (e.g., withdrawal of medical services if premium is not paid), assurance (e.g., having a second medical opinion in assessing large claims), and/or insurance (e.g., the patient's purchase of an insurance policy to insure against sudden financial burdens). These mechanisms can be identified by recognizing patterns in the dependency network, such as reciprocal links or loops, parallel paths, and redundant dependencies.

The AD model provides for three degrees of strength in a dependency. In an *open dependency*, the depender would be affected to some extent by a failure of the dependency, but would not suffer seriously. In a *committed dependency*, some course of action in the depender would fail. In a *critical dependency*, all known courses of action would fail [Yu93a]. A depender would go to different lengths to establish mechanisms for mitigating vulnerability according to the degree of dependency.

The **Functional Rationales** (FR) model makes explicit the means-ends relationships that underlie the structure of the business process. While the AD model describes only the external relationships among actors, the FR model captures the rationales that are internal to an actor. The network of means-ends relationships can be viewed as constituting a *plan* for meeting some goal. A plan is made up of *subgoals*, *subtasks*, and *resources*, connected by *means-ends links*. Unlike robot planning in AI, plans in the FR model typically do not include all details, but only those elements that are considered important to the success of the plan, and in particular, those elements connected to dependencies with other actors. An important role of the FR model is to show the structure of a plan in terms of decompositions and alternatives (in the form of an AND/OR graph), thus making explicit the choices that are available for achieving a goal.

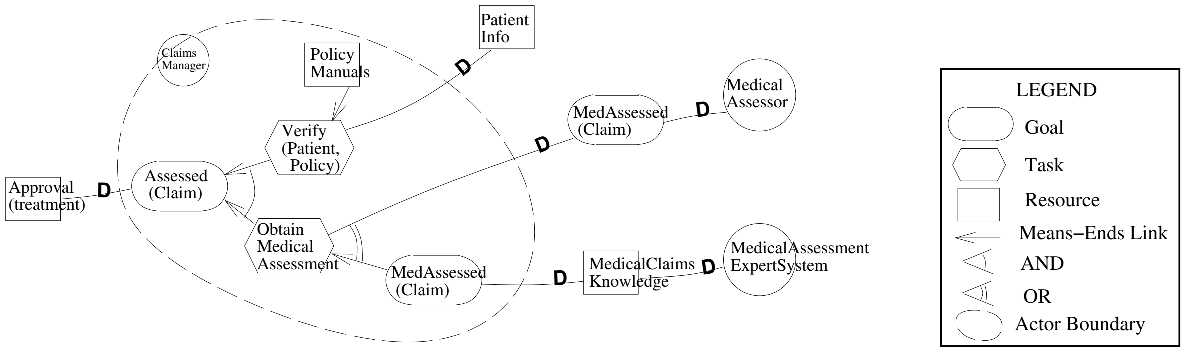

Figure 3: Functional Rationales Model (FR).

Figure 3 shows an FR model for the claims manager, pertaining to the goal of assessing a claim. The goal is decomposed into two subtasks: verifying coverage of the patient's policy, and obtaining a medical assessment. The latter can be met by one of two alternatives: obtaining the assessment from a medical assessor, or doing the assessment herself, with the help of (a dependency on) an expert system containing appropriate knowledge.

The means-ends links may be seen as instantiations of *rules* encoding generic knowhow for achieving (functional) goals [Yu93d]. For instance, there may be a rule which says that an insurance claim can be assessed using an expert system if a body of expert knowledge exists for that insurance domain. An FR model can be analyzed for consistency with a library of such rules.

The **Non-Functional Rationales** (NFR) model captures the decision rationales used in selecting among alternatives in the FR model. An NFR model is a network structure whose nodes are *non-functional goals* (soft-goals as introduced in the AD model). The links in a NFR model are also means-ends links, as in an FR model. However, a link in the NFR model is interpreted as indicating a *contribution* from one node (the subgoal) to another (the goal), which may or may not meet the goal fully. Contributions can be positive or negative.

For example, in deciding between a human medical assessor or an expert system assistant, non-functional criteria may include turnaround time and cost for processing a claim, and the quality of the medical opinion rendered (Figure 4). Contribution links are seen as invocations of *methods* [Yu93d]. For example, using an expert system is one method for achieving faster turnaround. Links that indicate the cross-impact among non-functional goals are instances of *correlation rules*. For example, improving turnaround tends also to improve operating costs. An NFR model can be analyzed with respect to how well the non-functional goals are met, using a qualitative reasoning scheme. The NFR model is based on a framework for dealing with non-functional requirements in software engineering [Chung93].

The series of four models – WF, AD, FR, and NFR – represents an increasing depth of understanding into the organizational environment. Separate models are used in order to support the different forms of knowledge involved and to permit different styles of reasoning. These four models together serve to help in the analysis and re-design of organizations and business processes, with

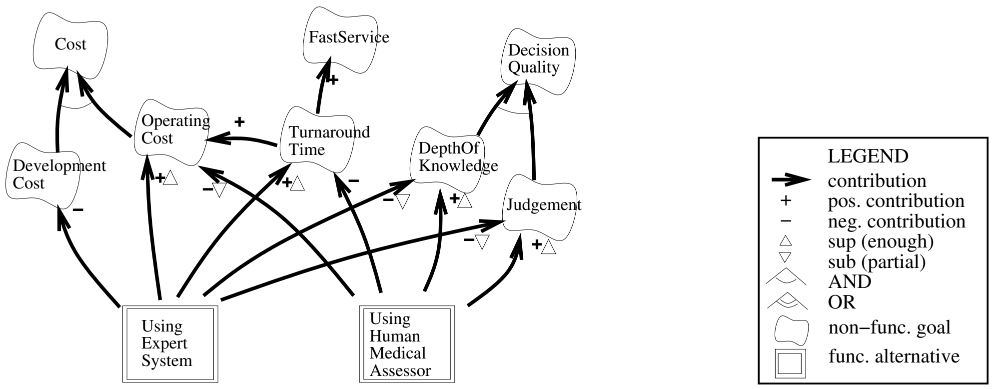

Figure 4: Non-Functional Rationales Model (NFR).

information systems as integral, embedded elements.

Figure 5 shows, in the format of an SADT activity diagram [Ross77], how the four models are inter-related. A requirements engineering process may be initiated by some process design objectives, such as improving service or reducing costs. These non-functional design goals would lead to considerations of functional alternatives (e.g. expert system vs. human expert), with the help of methods and rules. Organizational members, being stakeholders in the organization design effort, would try to obtain work relationships (dependencies) best suited to their work and personal goals. Routine, normal work patterns would emerge as the dominant workflows. Those fragments of the AD and WF models which refer to information items and their processing would then be translated into information system development requirements.

Conversely, and more typically of a reengineering effort, motivations, intents, and rationales might be discovered by asking "why" questions about the existing workflow, gradually revealing stakeholder interests and alternatives. In practice, a requirements engineering process is unlikely to be purely top-down or bottom-up; incremental adjustments in all directions are likely to be required as new issues surface and are negotiated upon.

By explicitly representing and utilizing four types of knowledge, the proposed framework makes the requirements engineering process more tangible and systematic. General principles as well as case experiences can be encoded into libraries to assist in analysis and design. The formal representation of the framework builds on the Telos language for object-oriented conceptual modelling [Mylopoulos91a], thereby gaining structuring mechanisms such as classification, generalization, aggregation, and time. For modelling complex organizational relationships, actors are differentiated into *roles*, *positions*, and *agents* [Yu93e].

The framework follows an approach to software engineering and information system development based on *conceptual modelling* [Mylopoulos91b] [Jarke92]. The premise is that, in order to organize and manage the large amounts of knowledge involved at each phase in software development, appropriate schemes for representing and using the concepts relevant to each phase are essential. Different conceptual models are needed because each phase addresses different subject matters, involves different types of knowledge, has different built-in assumptions, and may require different styles of reasoning.

The framework described in this paper extends previous work that follow this approach by proposing a set of knowledge representation constructs for the earliest phases of system development, namely, understanding and redesigning organizations, leading to technical system require-

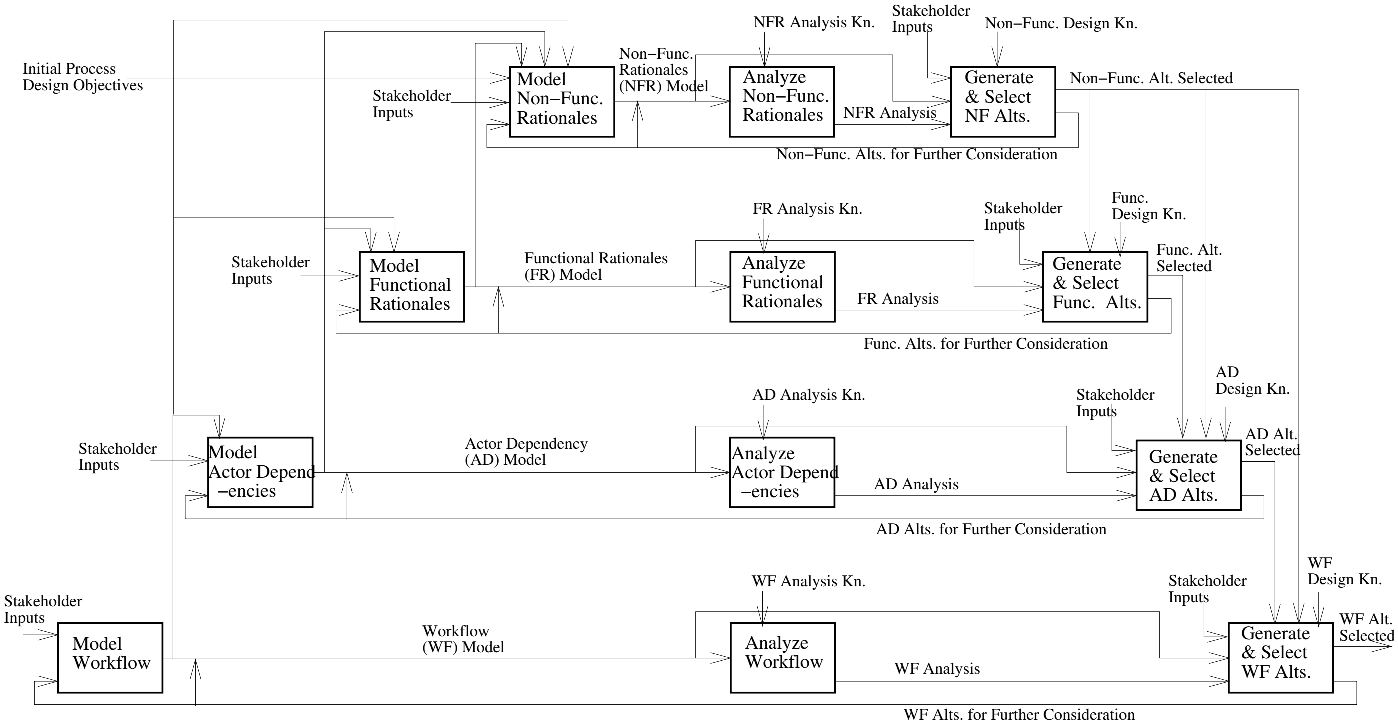

Figure 5: Understanding and Designing Organizations and their Information Systems Using the Proposed Framework.

ments. The use of knowledge representation concepts with formal semantics encourages greater precision in modelling and helps avoid ambiguities. Computer-based tools can be developed to support analysis and design. However, as we progress upstream in the development phases to address the organizational domain, the subject matter becomes less mechanistic and more open-ended. The models tend to be more incomplete. The kinds of support provided will emphasize a high degree of interactive exchange, since analysts' insights and designers' expertise will still be required.

The common underlying mechanisms in a conceptual modelling approach make it possible to link organizational structures and rationales to technical system requirements, and to the design and implementation decisions that follow from them. These linkages permit a high degree of traceability, and can be used to identify potential impact, as well as to propagate changes, in both directions.

In the recent requirements engineering literature, goal-based approaches have been proposed, e.g., [Dardenne93]. Goals are reduced during the requirements process until actions (assigned to actors) are arrived at. The resulting requirements model corresponds to the WF level of modelling. An AD model can be seen as a representation of a multi-agent organization in which the goals are partially reduced and distributed among the actors. Actors have freedom to choose their actions so that they can meet goals under varying circumstances. Actors' freedom, however, are limited because of their dependencies on each other. An AD model is a structural description of an organization that avoids a deterministic view of actors by modelling the expectations that actors have about each others' behaviour, rather than pre-determined courses of action. Concepts of dependency have been used by a number of organization theorists (e.g., [Emerson62] [Thompson67] [Pfeffer78]). The formal semantics of the AD model are based on agent modelling concepts developed in AI [Cohen90] [Thomas91].

The FR model is based on planning concepts in AI, especially as applied to organizational work, e.g., [Croft88]. The NFR model is an application of a framework originally designed for dealing with non-functional requirements in software engineering [Chung93], which is based on ideas from argumentation, design rationale, and decision support, e.g., [Lee90].

Earlier versions of portions of this framework have been presented in the context of requirements engineering [Yu93a], business process reengineering [Yu93c] [Yu93d], and understanding the organizational impact of information systems [Yu93b]. The framework has also been applied to software process modelling, analysis, and design [Yu93e]. The modelling concepts from this framework has also contributed to conceptions for information systems of the future, which need to take into account the broad spectrum of emerging technologies, as well as the new, more interdependent, and faster-paced realities of organizational environments [Mylopoulos93a] [Mylopoulos93b]. On-going work on the framework include further axiomatic characterization of the modelling features, and identification of computational algorithms [Yu93f].

Acknowledgements

The author gratefully acknowledges the encouragement and guidance of Prof. John Mylopoulos in the development of this framework, and the many useful comments from Lawrence Chung, Brian Nixon, and Prof. Carson Woo.

References

[Borgida85] A. Borgida, S. Greenspan, J. Mylopoulos, Knowledge Representation as the Basis for Requirements Specifications, *IEEE Computer*, April 1985, pp. 82-91.

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

[Cohen90] P. R. Cohen and H. J. Levesque, Intention is Choice with Commitment, *Artif. Intell.,* 42 (3), 1990.

[Croft88] 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.

[Dardenne93] A. Dardenne, A. van Lamsweerde, S. Fickas, Goal-directed requirements acquisition, *Science of Computer Programming*, v. 20, 1993, pp. 3-50.

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

[Dubois86] E. Dubois, J. Hagelstein, E. Lahou, F. Ponsaert and A. Rifaut, A Knowledge Representation Language for Requirements Engineering, *Proc. IEEE*, 74 (10), Oct. 1986, pp. 1431-1444.

[Emerson62] R. M. Emerson, Power-Dependence Relations, *American Sociological Review*, v. 27, February 1962, pp. 31-40.

[Greenspan84] S. J. Greenspan, *Requirements Modelling: A Knowledge Representation Approach to Software Requirements Definition,* Ph. D. Thesis, Dept. of Comp. Sci., Univ. of Toronto, 1984.

[Hammer90] M. Hammer, Reengineering Work: Don't Automate, Obliterate, *Harvard Business Review*, July-August 1990, pp. 104-112.

[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.

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

[Lee90] 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.

[Malone87] T. W. Malone, Modeling Coordination in Organizations and Markets, *Management Science*, vol. 33, 1987, pp. 1317-1332.

[Mylopoulos91a] J. Mylopoulos, A. Borgida, M. Jarke, M. Koubarakis, Telos: Representing Knowledge about Information Systems, *ACM Trans. Info. Sys.*, 8 (4), 1991.

[Mylopoulos91b] 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.

[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.

[Mylopoulos93a] J. Mylopoulos, E. Yu, Information Repositories and Information Agents, *First Intentional Conference on Intelligent and Cooperative Information Systems*, keynote address, May 12-14, 1993, Rotterdam, Netherlands; paper to appear.

[Mylopoulos93b] J. Mylopoulos, E. Yu, Aligning Information System Strategy with Business Strategy: A Technical Perspective, *International Workshop on Next Generation Information Technology and Systems*, keynote address, June 28-30, 1993, Technion, Israel.

[Pfeffer78] J. Pfeffer, G. Salancik, *The External Control of Organizations: A Resource Dependence Perspective*, Harper and Row, 1978.

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

[Thomas91] 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.

[Thompson67] J. D. Thompson, *Organizations in Action,* McGraw-Hill, 1967.

[Yu93a] E. Yu, Modelling Organizations for Information Systems Requirements Engineering, *Proceedings of First IEEE Symposium on Requirements Engineering*, San Diego, Calif., 1993, pp. 34-41.

[Yu93b] 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, pp. 66-86.

[Yu93c] 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.

[Yu93d] E. Yu, J. Mylopoulos, Using Goals, Rules, and Methods To Support Reasoning in Business Process Reengineering, *Proc. 27th Hawaii International Conference on System Sciences*, Maui, Hawaii, Jan. 4-7, 1994, to appear.

[Yu93e] E. Yu, J. Mylopoulos, Understanding "Why" in Software Process Modelling, Analysis, and Design, submitted for publication.

[Yu93f] E. Yu, *An Organization Modelling Framework for Information Systems Requirements Engineering*, Ph.D. Thesis, Dept. of Computer Science, Univ. of Toronto, forthcoming.

[Zave81] P. Zave, R. T. Yeh, Executable Requirements for Embedded Systems, *Proc. 5th Intl. Conf. Soft. Eng.*, 1981, pp. 295-304.