Stability and Change in Organizing Knowledge for Information Systems Development

Abstract: Information systems are complex and dynamic artifacts with numerous components, sub-components, and inter-relationships among them. Systems development, maintenance, and evolution are knowledge-intensive activities. The paper outlines a number of schemes for organizing and managing knowledge used in the information systems area, illustrating a trend towards more explicit representations of knowledge to deal with change, and the social context of knowledge.

1. Introduction

Knowledge organization schemes are becoming increasingly important in many areas of endeavour as the volume and complexity of information and knowledge multiply rapidly in these areas. Knowledge organization techniques must be able to deal with change. In the health care sector, for example, new treatments are invented, new cases arise, new findings are discovered in studies. In business enterprises, environmental conditions are causing new products and services to be created, and new practices introduced. Governmental agencies have to deal with changes in the global environment. All these changes present challenges to knowledge organization.

One of the most challenging areas in which knowledge organization can play an important role is the development of computer-based information systems (IEEE, 1992, Mylopoulos, Borgida & Yu, 1997). Building an information system involves understanding the application area (e.g., monitoring patients, manufacturing products, supporting business transactions, servicing aircrafts, etc.), understanding the enabling information technologies (e.g., databases, programming languages, networks and protocols, interfaces, etc.), and selecting and putting together technical components to address application domain problems. Large software systems may have thousands of pages of requirements specifications, volumes of design documents, and hundreds of thousands of lines of computer code or more. While it is already a challenge to produce a working system in the first place, the greater challenge is in being able to maintain the system so that it can evolve to keep up with frequent changes in user needs as well as advances in enabling technologies (De Michelis, 1998). The inability to deal with change has resulted in the conundrum of "legacy systems." These are systems that are heavily relied on for everyday operations in an organization, and yet they present obstacles to change because the documentation is lacking or so poor that designers cannot upgrade them. The "Y2K bug" was a special case that heightened awareness of the more general problem.

While managing the information and knowledge involved in information systems development has a great deal in common with managing knowledge in other areas, it also has its special features. As the concrete product is computerized information and processing, many of the information artifacts produced throughout the systems development process also lend themselves to some degree of computer support. Information can be quite finely divided into small units (e.g., data items, procedure calls, design concepts, requirements statements, etc.), with well-defined relationships to other types of units. In the historical development of the fields of information systems and software engineering, many methods for organizing information and knowledge have been invented. Various specialized languages and models
are used for representing and codifying knowledge about the application domain, requirements, technical system design, and implementation in specific technologies. Many of these representational and organizing techniques are at least partially supportable by computer-based tools – for storage and retrieval (as in computer-aided software engineering – CASE repositories and tools), and for analysis (e.g., to check whether requirements are consistent, and whether a design meets the requirements) (Mylopoulos, Jurisica & Yu 1998, Mylopoulos, 1998).

This paper examines a number of knowledge organization techniques used in the information systems area. From the viewpoint of dealing with stability and change, these techniques may be thought of as falling into four generations of development.

2. Organizing Knowledge for Systems Analysis

Since the early days of computing, standardization of processes and data content has been a fundamental source of power that underlies computerization. When work processes are uniform and their associated data highly structured, they can be easily codified and operated on by automated procedures. To represent knowledge about these processes and data for the purpose of defining and analyzing an information system, modelling techniques such as Structured Analysis and Design Technique (SADT) (Ross & Schoman, 1977) and Data Flow Diagrams (DFD) (De Marco, 1979) and Entity-Relationships (ER) modelling (Chen, 1976) have been developed, and are still widely used.

![Entity Relationship Diagram](image)

**Figure 1 An Entity Relationship Diagram**

Figure 1 shows an ER diagram for an airline reservation system. It specifies the kinds of information that need to be kept in terms of entities, relationships and attributes. Customers have names and addresses. Each customer can be booked on zero or more flights. Each flight flies from and to exactly one city, and so forth. Knowledge represented in this kind of model is crucial for designing databases and supporting queries. Similarly, the activity or process aspects of an information system may be described using DFDs or SADT diagrams.

These modelling techniques are used to represent complex knowledge concisely and unambiguously so as to facilitate communication and analysis, e.g., to check that all the needed relationships are included, or that inputs match up with outputs. Prior to the introduction of these techniques, systems requirements were expressed primarily in textual form, which made them very difficult to analyze, or to check for errors and omissions.

During this first generation, systems were pretty much developed in isolation from each other, and without much concern for future evolution. When changes occur (for example when the airline wants to offer new types of services), there are no systematic ways for revising these models and to support and keep track of changes. Nevertheless, it is recognized that these models capture crucial knowledge about the “business” of the organization and constitute part of its “knowledge assets”.
As experience with these techniques accumulated, it soon became apparent that dealing with change in a large number of models was a problem. The "Information Engineering" approach (Martin & Finkelstein, 1981) for instance, emphasized data modelling in recognition that data schemas tend to be more stable (less susceptible to change) than process models.

3. Organizing Knowledge for Sharing and Reuse

A second generation of techniques focused on recouping the large amounts of investment spent on developing systems that may have a great deal in common. Object-oriented techniques aim to reduce costs and increase robustness of systems by classifying and then sharing or reusing common concepts and behaviours among similar systems (Coad & Yourdon, 1990, Rumbaugh, 1991, Rational, 2000). Related processes and data are grouped into units called "objects". Objects are instances of classes. All instances of a class have the same attributes (but different attribute values). The class definition specifies the structure and behaviour of all its instances. Some languages allow the features and properties of classes to be defined in metaclass definitions, and so on (Mylopoulos et al, 1990). A new class can be defined by specifying what additional features it has in relation to (one or more) already defined class(es). The new class is said to be a specialization, and inherits the features of the more generic class. Figure 2 shows a specialization hierarchy, including an example of inheritance from multiple super-classes. Figure 3 shows some class definitions with attributes.

![Figure 2 - A specialization hierarchy of classes, with example of multiple inheritance](image)

The object-oriented approach thus aims to build a body of relatively stable knowledge (the more generic classes), with changes happening most frequently in the specializations. For example, the generic concept of customer may be quite stable, while mail order customer and web-based customer may depend on how these services are offered.

![Figure 3 - Some class definitions, with attributes](image)
4. Organizing Knowledge for Supporting Design Reasoning and Change

While the object-oriented approach allows more stable and reusable knowledge to be distinguished and treated separately (along various hierarchies) from the more fluid and specific knowledge, it does not explicitly support and manage change. Techniques for explicit change control, version and configuration management have been in use in software engineering, mostly to assure that the right pieces of software code are assembled to build a new release (Berlack, 1992). These techniques can be used to identify which are the right pieces to put together, but they do not provide guidance or rationales for the changes.

A third generation of techniques can thus be identified. They allow design rationales to be captured and later used to explain or guide change (Potts & Bruns, 1988, Lee & Lai, 1991). Traceability links allow design decisions to be traced back to requirements (Jarke, 1998).

A more proactive approach to guiding change is to provide representations for goals and methods for achieving goals (Chung, Nixon & Yu, 1997, Mylopoulos, Chung & Yu, 1999). Figure 4 shows a small example of a goal-oriented technique for supporting design reasoning. In a ticket reservation system, security of the ticket information is treated as a goal to be achieved. The goal is decomposed into sub-goals: availability, integrity, and confidentiality. Meeting all of these sub-goals results in the higher goal being met. (Arrows indicate direction of contribution towards goal achievement.) Design goals are successively refined until concrete measures (shown as heavier circles) can be identified. For example, using fault-tolerant processors or redundant disk arrays will each contribute to addressing the goal of reliability of ticket information processing.

During design, many competing requirements need to be addressed – for example, usability, performance, cost, maintainability, scalability, etc. (e.g., Boehm & In, 1996). Each can be elaborated incrementally in tree-like structures (Figure 4). However, due to the many interactions among requirements and design options, the overall goal contribution graph is a network, with many links crisscrossing a number of predominantly tree-like structures. For example, while fault-tolerant processors may be good for reliability, they may increase costs, delay time-to-market, and make the system harder to scale up.

Managing and maintaining these collections of design knowledge is very important. Information systems must adapt to new environments all the time. A ticketing system designed originally for use by travel agents may need to be extended to serve customers directly via the Internet. Knowing what were the original design goals, what interactions existed, and what tradeoffs were made, will facilitate updating the system to meet new goals.

Knowledge organization schemes are needed to manage the generic know-how (e.g., that the Security goal can be refined into Availability, Integrity, and Confidentiality) and the
actual use of that knowledge to design the system (that the above piece of generic knowledge is judged to be applicable to this ticketing system). The case-specific knowledge involves networks of design goals, alternatives, interactions, decisions, arguments, justifications, assumptions, etc. The generic know-how may be organized as catalogues of means-ends knowledge with applicability conditions, which can be searched using various criteria (Chung, Nixon, Yu & Mylopoulos, 2000).

5. Incorporating the Social Dimension

The above techniques generally assume that the more knowledge one can capture, codify, and explicitly manage, the better one can cope with and control change. They provide mechanisms for linking, classifying, navigating, querying, testing, and analyzing and propagating properties across knowledge representation structures. However, since the processes of knowledge creation and use and evolution are very much human activities (e.g., Nonaka, 1995), there are limits to explicit representation, and indeed to explicit management and control.

A new, fourth generation of modelling techniques is emerging. The agent-oriented approach acknowledges that there will always be incompleteness of knowledge and control (Yu, 1993). Contingencies and uncertainties can be handled by humans with the appropriate knowledge, but that knowledge is not always accessible for examination or codification. The approach incorporates intentional concepts such as goals, beliefs, ability and commitment, but emphasizes the strategic relationships among agents (Yu, 1995, 1997, 1999). The approach brings ideas from the social sciences to enrich the understanding and development of technical systems. Such an approach is necessary as information systems are increasingly being subsumed as one component under the more encompassing notion of knowledge management (Jurisica, Mylopoulos & Yu 1999). The boundaries between system development, system use, and system evolution are being challenged, with the treatment of knowledge being a central issue. The agent-oriented approach provides modelling and analysis support not only for the knowledge contents and processes, but also for the embedding organizational environments.

Figure 5 A strategic dependency model from the $i^*$ framework

The $i^*$ modelling framework (Yu, 1995) treats agents as social actors who are free to act within a network of relationships with other actors. Figure 5 shows a (simplified) configuration of dependency relationships among various parties in airline ticket reservation. Customers depend on travel agents to find seats. Safety, convenience, price, and comfort are accompanying goals to be met by the agent. The travel agent in turn needs prices and
availability information from the airline. These should be accurate and up-to-date (not shown). She also depends on safety information from the regulatory agency. The internal reasoning of each actor is modeled in terms of goal graphs as exemplified in Figure 4. However, the agent-oriented approach acknowledges that the internal models are necessarily incomplete and are subjective from each actor’s viewpoint. Actors are strategic in that they evaluate their opportunities and vulnerabilities within the network of dependencies. Systems design is seen in the context of various stakeholders each seeking to advance their strategic interests by altering the dependency configurations.

Strategic network models offer yet another new way for managing knowledge in the face of change. When customers can directly access airline ticketing systems via the Internet, what are the strategic implications for travel agents? Are customers adequately served by the new configuration of relationships? What new roles can travel agents play as intermediaries? These strategic issues lead directly to the analysis of information needs and system requirements. Having appropriate knowledge representation and organizing schemes are therefore crucial for supporting systematic exploration and analysis of these issues.

The system development process itself is a complex social activity that can benefit from an agent-oriented approach (Yu & Mylopoulos, 1994, Chung, Gross, & Yu, 1999).

6. Conclusions

Information system development is a knowledge-intensive activity. There are numerous facts, issues, options, tradeoffs, decisions, assumptions, etc. that need to be organized and managed systematically if the systems are to be able to cope with change. Earlier schemes for representing knowledge in information system development and software engineering have not treated change as a major consideration. Most systems in use today are not well adapted to change because they lack the knowledge organization to support change. This paper outlined a progression of research results that are leading towards new techniques that put knowledge organization and management at the centre of information system development. Efforts are underway to develop these techniques more fully.

Acknowledgements. This research is supported in part by the Natural Sciences and Engineering Research Council of Canada, Communications and Information Technology of Ontario, and the IRIS Federal Network of Centres of Excellence.

References