Tools for inventing organizations: Toward a handbook of organizational processes

Thomas W. Malone
Kevin Crowston[1]
Jintae Lee[2]
Brian Pentland[3]

CCS WP #141, Sloan School WP # 3562-93

May 1993

In *Proceedings of the 2nd IEEE Workshop on Enabling Technologies Infrastructure for Collaborative Enterprises*, Morgantown, WV, April 20-22, 1993.

[1] University of Michigan
[2] University of Hawaii
[3] University of California at Los Angeles

Tools for inventing organizations:

Toward a handbook of organizational processes

Thomas W. Malone
Kevin Crowston[1]
Jintae Lee[2]
Brian Pentland[3]

Working Paper

Center for Coordination Science
Massachusetts Institute of Technology

May 1993

In *Proceedings of the 2nd IEEE Workshop on Enabling Technologies Infrastructure for Collaborative Enterprises*, Morgantown, WV, April 20-22, 1993.

[1] University of Michigan
[2] University of Hawaii
[3] University of California at Los Angeles

ABSTRACT

This paper describes a new project intended to provide a firmer theoretical and empirical foundation for such tasks as enterprise modeling, enterprise integration, and process re-engineering.

The project includes (1) collecting examples of how different organizations perform similar processes, and (2) representing these examples in an on-line "process handbook" which includes the relative advantages of the alternatives. The handbook is intended to help (a) redesign existing organizational processes, (b) invent new organizational processes that take advantage of information technology, and perhaps (c) automatically generate software to support organizational processes.

A key element of the work is a novel approach to representing processes at various levels of abstraction. This approach uses ideas from computer science about inheritance and from coordination theory about managing dependencies. Its primary advantage is that it allows users to explicitly represent the similarities (and differences) among related processes and to easily find or generate sensible alternatives for how a given process could be performed.

Tools for inventing organizations: Toward a handbook of organizational processes

There is much talk today about "new organizations" and new managerial tools like "total quality management," "process re-engineering," and "information-based organizations." Even though there is real value in the ideas summarized by these slogans, much of the talk about them relies on vague descriptions of the new organizations and provides little guidance about how we might go beyond the innovations already being advocated.

For instance, for a long time, most American manufacturing firms organized their design efforts as a kind of pipeline with the results of one step in the design process being "thrown over the wall" to the next step. Marketing specialists gave customer requirements to designers who created a product design. Then the product design was given to process engineers who specified the manufacturing process and eventually to purchasing specialists who selected vendors for the parts. Recently, most manufacturing managers have come to believe that there is a much better way to organize this process: as "concurrent engineering" where all these functions are performed concurrently and iteratively by design teams with specialists from all the relevant functional areas (e.g., Carter & Baker, 1991).

Similarly, many American firms are now trying to adopt "Just In Time" (JIT) inventory control methods that are already widely used in Japan. In this approach, parts are delivered just before they are needed, and the customer avoids having to store and pay for significant inventories of the parts.

How can we move beyond these best practices of today (borrowed from Japan or elsewhere) to "invent" the next set of organizational innovations? And where will we keep getting new ideas for new organizational processes to adapt to a continually changing world? For instance, how can we understand and exploit the new organizational possibilities enabled by the continuing, dramatic improvements in information technology.

We believe our best hope for progress on these problems is to develop a much more systematic theoretical and empirical foundation. For instance, if we are ever to understand successful organizational practices, we must be able to recognize and represent the organizational practices we see. And in order to improve organizational practice in a particular situation, we must also be able

to imagine alternative ways of accomplishing the same things. Finally, we need some way of judging which alternatives are likely to be useful in which situations.

This paper describes a new project at the MIT Center for Coordination Science to address these problems by (1) developing methodologies and software support for representing and codifying organizational processes and structures at varying levels of abstraction and (2) collecting, organizing, and analyzing numerous examples of how different groups and companies perform similar functions.

We expect the result of this work to be a kind of on-line "process handbook" which organizations can consult to find a variety of alternative ways for performing particular processes, along with experiences and guidelines about which alternatives work best in which situations. We hope this handbook will be useful in (a) inventing new organizational processes, (b) redesigning existing organizations, (c) educating students about organizational practices, and possibly (d) automatically generating software to support organizations.

Goals

Even though we hope our process handbook will eventually have many different uses, this project initially focuses primarily on (1) helping theoreticians imagine new organizations and (2) helping consultants, managers, and others redesign existing organizations.

Primary goal: Theoretical tools to help "invent" new organizational processes, especially those enabled by information technology

Our primary goal is to develop a representation technique, database, and methods that will help theoreticians make systematic, empirically-grounded suggestions about possible new organizational processes.

One of the most dramatic changes visible in the business world today is the increasing pervasiveness of information technology in more and more business processes. Therefore, we will focus especially on using our conceptual tools to predict new organizational processes that will be enabled by pervasive information technology.

For example, we hope to systematically generalize the kinds of arguments we made in previous work about how decreasing coordination costs caused by information technology leads to new

coordination-intensive organizational structures (Malone, et al., 1987; Malone & Rockart, 1991). Our previous work, for instance, predicts that information technology will favor external coordination through markets rather than internal coordination within firms and thus will lead companies to "buy" rather than "make" more of the goods and services they use (Malone, Yates and Benjamin, 1987). Our preliminary econometric results are consistent with this prediction (Brynjolfsson, et al., 1989), and we hope that the work proposed here will allow us to make similar predictions about other coordination structures.

In addition to analyzing organizational uses of information technology, we expect that the same conceptual tools will also be useful for suggesting responses to other environmental changes such as those in employees' skills, legal constraints, or production technologies.

Secondary goal: Practical tools to help (re)design organizations rapidly and effectively

Many observers of modern organizations have noted that organizations appear to be changing more and more rapidly, and if anything, this trend appears to be accelerating (Toffler, 1970; Toffler, 1990). Reorganizations are becoming a common fact of life rather than rare events involving primarily senior managers and outside consultants. If this is true, it will become increasingly important for organizations to have better tools for suggesting and keeping track of their numerous reorganizations and changes.

We hope that the kinds of insights and examples our handbook contains will be useful in many of these organizational redesign efforts. Possible users for the handbook include: (a) consultants (either internal or external) who are helping redesign organizations (e.g., re-engineering consultants, "quality" consultants, etc.), (b) managers designing the processes they supervise, and (c) employees at all levels who are designing their own work processes.

Of these primary kinds of possible users, our initial target audience is organizational redesign consultants. Since they specialize in this topic, we believe they will be willing to invest more time in learning to use the concepts and the system than managers or employees, who have other responsibilities as well. Therefore, we believe consultants are an "easier" target. Our hope, however, is to develop a system that is easy enough to use that it can be used by the other groups as well.

Other goals

In addition to the above goals, two other uses for our handbook seem particularly attractive:

Teaching students about organizations and organizational design. The handbook has the potential to be useful to students at various levels. For undergraduates unfamiliar with basic organizational functions (e.g., marketing, personnel, accounting, purchasing, sales, manufacturing), the process handbook would provide an interactive overview. For more advanced students, it would provide a way of learning about and comparing alternative designs for various organizational functions in different industries. Because it contains a database of processes, and an analytical framework with which to compare those processes, the handbook would provide a valuable resource for creating and analyzing so-called "best practices."

Automatically generating software to support the processes. One of the most ambitious (and most interesting) possible uses of the kind of knowledge we hope to capture in our handbook is automatically generating software to support the processes represented (see Marques, et al., in press; Dallemagne, et al., 1991). For instance, we can imagine that when a group of employees recognizes that they all need to share use of the same machine, they might consult the handbook for a variety of alternative processes for scheduling shared resources (e.g., "first come/first serve", "priority order", "bidding", and "managerial decision"). After the group selects one of these processes and specializes it for their own particular situation, it might be possible to automatically generate a customized scheduling application specifically tailored to the needs of this group. Software to support many other workflow processes (such as approval processes, hiring procedures, and equipment ordering methods) might also be easily generated using this approach.

Example

To see how a handbook like this might work, consider the following hypothetical scenario. Imagine that you are a consultant to the general manager of a new division of a large computer hardware vendor. This vendor has traditionally used a highly skilled direct sales force, but the mission of the new division is to sell personal computers and software by direct mail. A small publishing group has existed in this company for years, distributing documentation and other technical reference material by mail order. This publishing group will also be incorporated into the new division.

Your job is to help the new manager decide how to staff and structure this new division. Should you simply adopt and scale up the processes already in use in the mail order publishing group? What new problems might arise when you try to scale up these processes to sales volumes 50 to 100 times what they have been? Are there other processes that might be better suited for the volumes, products, and customers you are targeting? What is the "best practice" among mail order vendors in other industries? Can you exploit advanced information technology to organize a highly efficient mail order service? For instance, would it be possible to guarantee customers overnight delivery of the products they order?

In order to investigate these questions using the handbook, you would take the following steps:

(1) First, you specify the general situation from a list of "generic" processes that are available in the handbook. You can select at a rather high level of abstraction (e.g., "Selling a product") or a more specific level ("Direct mail sales"). This provides an "anchor" or point of entry into the space of possible processes catalogued in the handbook. Choosing the more generic level will open up a wider range of comparative alternatives, which may or may not be relevant to your objectives.

(2) Within that general situation, you may select specific features or goals of interest to you. For example, in this case, you are already constrained to sell three classes of products: computer hardware, software, and reference material. Two of these products (hardware and software) require a relatively large amount of customer service. Furthermore, you would like a very short response time (even overnight delivery, if possible).

(3) At this point, the handbook retrieves (or generates) a set of alternative organizational forms and displays them in a matrix with each alternative rated in terms of goals such as "cost of sales", "response time", and "quality of customer service". In cases where it is difficult to evaluate an alternative reliably, the range of possibilities or the degree of uncertainty is indicated.

(4) The handbook also provides a variety of interactive ways to explore, compare, combine, and redesign the alternatives. For example, you can: (a) view a flow diagram for each process, (b) examine the basis for the evaluation on each goal, (c) see "tips" for success in using the processes, and (d) find examples of specific companies that use the processes. For instance, the processes used by Lands End, a widely admired mail order clothing company, might be described along with references to documents and other sources of more information.

(5) Eventually, you may want to relax some of the initial constraints and let the system suggest more radical innovations. For instance, what if you distributed through your own new chain of retail stores? What if you let customers place orders through PC-based programs or through touch-tone telephone systems? Would it even be possible for all employees of the division to work at home, or for all the "employees" to work as independent contractors?

Of course, the desirability of these alternative processes will often depend on factors in the actual situation that are not represented in the handbook, and we will rely on intelligent users to take these other factors into account. The handbook, however, can help these users systematically examine many possibilities they might never otherwise have considered.

The key intellectual challenge: How to represent organizational processes?

A key to this project is developing techniques for representing processes. Fortunately, the last several decades of research in computer science and other disciplines has resulted in a number of well-developed approaches to representing processes (see section below titled "Process models"), and we expect to draw extensively upon this work. Several of these approaches have been applied to representing specific processes in particular organizations. It is clear, therefore, that we could use these techniques to assemble a large set of individual process descriptions, collected from many different organizations. Such a "library" of process descriptions would be of immediate use to managers and consultants, and provides a kind of "baseline" for the contribution this project could make.

We hope, however, to make an additional and much more significant intellectual contribution. To do this, we are developing a language for describing organizational processes that explicitly represents the similarities (and the differences) among a collection of related processes. For example, with this approach, we can represent a generic "sales process" and then represent variations from this process by simply indicating how they differ from the generic process. Then, instead of having to analyze each new organizational situation "from scratch," we should be able to go into a new situation, and by noting only a few features of the situation (such as the goals to be achieved and the type of organization), we could immediately bring to bear a great deal of related knowledge about alternative processes that might be used in this situation.

This goal is similar, in some ways, to that of the few previous organizational analysts who have developed representations of "generic" organizational processes (e.g., Winograd & Flores, 1986; Carlson, 1979). In most cases, however, these previous approaches have focused on representing

a single high-level abstraction of a common process. By representing processes at many different levels of abstraction, and by representing alternative processes for achieving specific goals, we hope to provide more useful conceptual tools than this previous work.

In order to do this, we are exploiting two key sources of intellectual leverage: (1) notions of *inheritance* from knowledge representation, and (2) concepts about managing dependencies from *coordination theory*.

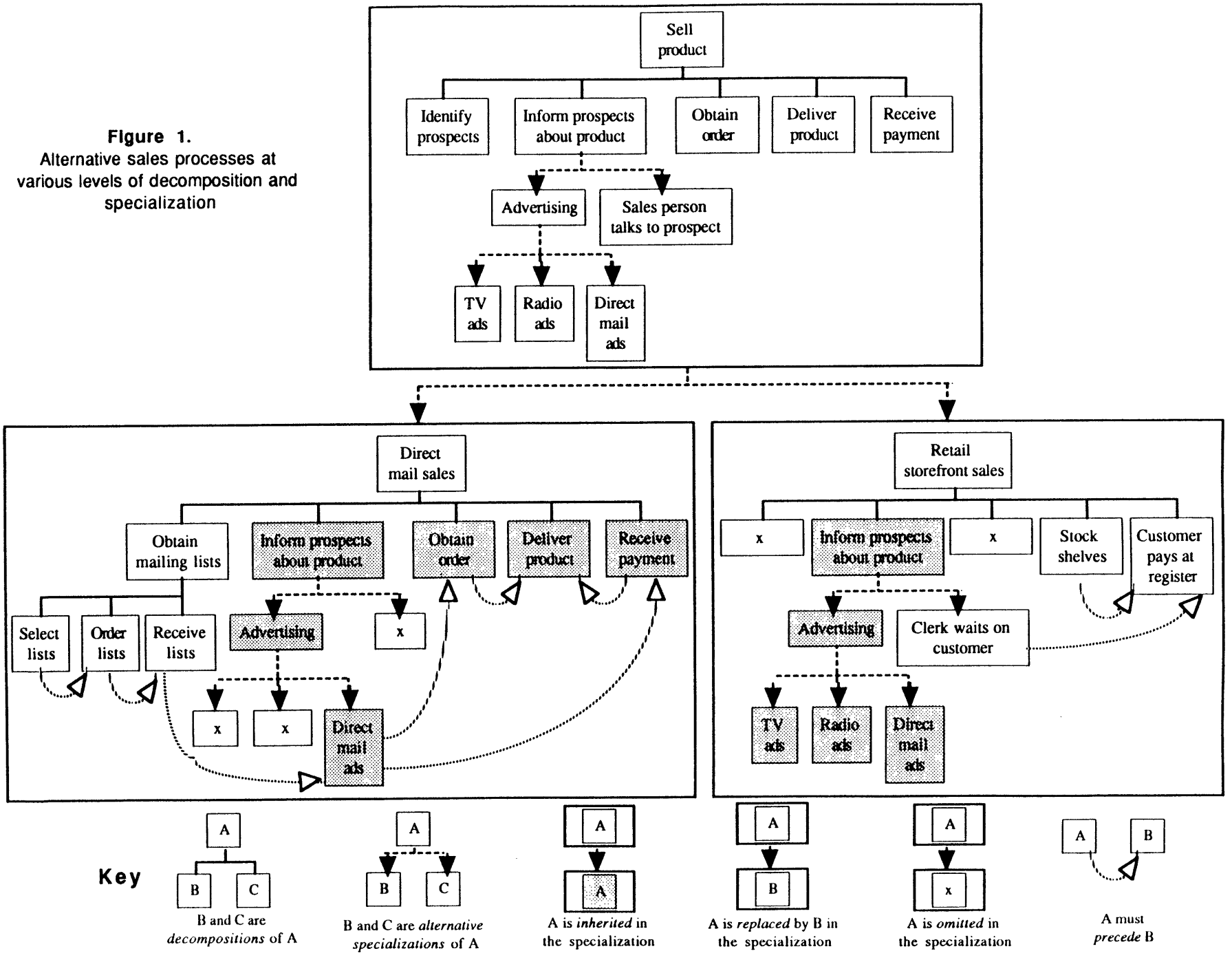
Inheritance

In the traditional notion of inheritance, as used in object-oriented programming and knowledge representation (e.g., Stefik & Bobrow, 1986; Wegner, 1987; Brachman & Levesque, 1985), objects in the world are arranged in a hierarchical network with general categories at the top and increasingly specialized kinds of objects at lower levels. For example, "Products" might be specialized into categories like "Goods" and "Services", and "Goods" might be specialized into categories like "Automobiles" and "Furniture". At each of these levels, objects may "inherit" characteristics from higher levels, and add or change characteristics of their own. For instance, all "Goods" might have a "Weight" and "Size", and "Automobiles" might also have a "Miles per gallon" characteristic.

In contrast to this traditional notion of inheritance, which is organized around a hierarchy of increasingly specialized *objects*, we propose to develop a hierarchy of increasingly specialized *processes*. It is important to note that this notion of process *specialization* is different from (but complementary to) the conventional notion of process *decomposition*. Figure 1 shows an example of how decomposition and specialization can work together using the preliminary representational scheme we have developed.

In this figure, the generic activity of "Selling a product" is *decomposed* into subactivities like "Identify prospects" and "Inform prospects about product". The generic activity is also *specialized* into more focused activities like "Direct mail sales" and "Retail storefront sales". These specialized activities automatically inherit the subactivities and other characteristics of their "parent" process. In some cases, however, the specialized processes add to or change the characteristics they inherit. For instance, in direct mail selling the subactivities of obtaining an order and delivering a product are inherited without modification. But identifying prospects is replaced by the more specialized activity of obtaining mailing lists, and the sales person talking to prospects is omitted altogether.

Figure 1.
Alternative sales processes at various levels of decomposition and specialization



In general, specialization can be used to indicate *alternative* ways of performing an activity. For instance, Figure 1 shows that "Selling a product" can be specialized into "Direct mail sales" or "Retail storefront sales" with a different set of modifications to the inherited subactivities in each case. In cases like this, where there are multiple alternative specializations for an activity, our handbook will be able to include a *tradeoff matrix* that compares the different alternatives in terms of their ratings on various *goals*. As in the Sibyl system (Lee, 1990), this tradeoff matrix can also include detailed justifications for the various ratings. In some cases, these tradeoff matrices may be the result of systematic studies; in others, they may be simply rough guesses by knowledgeable managers or consultants (with appropriate indications of their preliminary nature); and, of course, in some cases, there may not be enough information to include any tradeoff matrices at all.

These techniques of decomposition and alternative specialization can, of course, be used for activities at any level. For instance, Figure 1 shows that "Obtain mailing lists" can be further decomposed and "Inform prospects about product" can be specialized into "Advertising" or "Sales person talks to prospects". In general, we use decomposition to indicate "and" relationships, and specialization to indicate "or" relationships. This seems sensible, intutitively, because the notion of specialization implies that the specialized thing is "complete in itself", not just a part of something else.

Even though the examples in Figure 1 only show one "parent" for the activities that are specializations, it is also often useful to have *multiple inheritance* in which a single activity is a specialization of several parents. For example, "TV ads" might be a specialization, not only of "Advertising", but also of "TV broadcasts", and it might, therefore, inherit subactivities or other characteristics from both of these parents.

A refinement: "Bundles" of alternatives. Our preliminary work with this representation scheme has suggested that it is also useful to combine groups of alternative specializations into "bundles" of related alternatives. For instance, one bundle of specializations for "Sell product" is related to *how* the sale is made: direct mail, retail storefront, or telemarketing. Another bundle of specializations has to do with *what* is being sold: shampoo, computers, newspapers, or consulting services. And another bundle has to do with *who* is being sold to: consumers, hotels, manufacturing companies, and so forth. These bundles are used in two ways:

(1) Comparing alternatives in a tradeoff matrix is appropriate only within a bundle of related alternatives. For example, comparing "retail storefront sales" to "selling shampoo" doesn't make much sense.

(2) Alternatives in a bundle should automatically inherit alternatives from the other bundles but not the other alternatives in their own bundle. For instance, it makes sense for someone selling shampoo to be automatically presented with alternatives for direct mail, retail storefront, and telemarketing, but it does not make much sense for this person to be automatically presented with alternatives of selling computers, newspapers, and consulting. Users who identify their interest as selling shampoo could also always move up to the more generic activity of "Selling a product" to see other possible products.

Advantages of this approach. This method of representing processes using a combination of decomposition and alternative specializations has a number of significant benefits over previous process representation techniques. First, it can substantially reduce the amount of work necessary to represent a new process. By simply identifying a more general process that the new process is intended to specialize, most of the information about the new process can be automatically inherited and only the changes need to be explicitly entered. Second, changes made at a high level can be automatically inherited by more specialized processes, thus greatly simplifying the process of maintaining the process descriptions. Third, by explicitly representing alternative processes and their relative advantages and disadvantages, the task of selecting appropriate processes is facilitated.

Fourth, and perhaps most importantly, by arranging the alternative processes in a specialization hierarchy, the process of finding, combining, and generating relevant alternatives is greatly enhanced. Depending on their goals, users of the system can browse at various levels of abstraction, finding alternatives that are related according to the principles embodied in the specialization structure. For instance, merely collecting descriptions of how different companies sell consulting services would probably identify numerous examples of direct sales and perhaps mail advertising techniques. But the specialization hierarchy we have proposed would quickly lead users who were interested in more radical alternatives to consider options like retail storefront selling, even if no cases of consulting firms using this method had been observed. Thus, the system helps users generate new alternatives by creating new specializations of alternatives at higher levels of abstraction.

Coordination theory

The second key concept we are using is the notion from coordination theory (e.g., Malone & Crowston, 1991) that coordination processes can be thought of as ways of *managing dependencies*

Dependency	*Examples of coordination processes for managing dependency*
Shared resources	"First come/first serve", priority order, budgets, managerial decision, market-like bidding
Task assignments	(same as for "Shared resources")
Producer / consumer relationships	
Prerequisite constraints	Notification, sequencing, tracking
Inventory	Inventory management (e.g., "Just In Time", "Economic Order Quantity")
Usability	Standardization, ask users, participatory design
Design for manufacturability	Concurrent engineering
Simultaneity constraints	Scheduling, synchronization
Task / subtask	Goal selection, task decomposition

Table 1.
Examples of common dependencies between activities and alternative coordination processes for managing them. (Indentations in the left column indicate more specialized versions of general dependency types.)

between activities. We assume that all processes can be thought of as a set of *activities* (e.g., "steps", "tasks", or "subprocesses"). From this perspective, further progress should be possible by characterizing different kinds of dependencies and identifying the coordination processes that can be used to manage them.

Table 1 suggests the beginnings of such an analysis. For example, the table shows that shared resource constraints can be managed by a variety of coordination processes such as "first come / first serve", priority order, budgets, managerial decision, and market-like bidding. If three job shop workers need to use the same machine, for instance, they could use a simple "first come / first serve" mechanism. Alternatively, they could use a form of budgeting with each worker having pre-assigned time slots, or a manager could explicitly decide what to do whenever two workers wanted to use the machine at the same time. In some cases, they might even want to "bid" for use of the machine and the person willing to pay the most would get it.

As Table 1 suggests, some dependencies can be viewed as specializations of others. For instance, *task assignment* can be seen as a special case of allocating shared resources. In this case, the "resource" being allocated is the time of people who can do the tasks. This implies that the

coordination processes for allocating resources in general can be specialized to apply to task assignment.

It is important to note that the lists of dependencies and coordination processes shown in Table 1 are by no means intended to be exhaustive. However, many specific processes that arise in particular kinds of systems (such as "design for manufacturability") can be seen as instances of more generic processes (such as managing "usability" constraints between adjacent steps in a process).

By identifying various types of dependencies possible between activities and the associated coordination processes for managing them, we believe we will obtain several representational benefits in the process handbook. Two of the most important potential benefits are: *conciseness* and *generativity*.

Concise representations. Coordination theory suggests a new set of abstractions that can be used (together with inheritance) to provide a more *concise* representation of processes. Instead of having to explicitly list all the coordination activities separately in each different process, we will be able to simply indicate that "the dependency between activities A and B is managed by an instance of coordination process X".

For example, Figure 1 shows one very important kind of dependency between activities: *prerequisite* constraints. Note that no prerequisites are shown at the generic level of "Sell product", suggesting that the generic activities can, in principle, be performed in any order. The specializations of "Direct mail sales" and "Retail storefront sales", however, both include prerequisite constraints between activities. For instance, in direct mail, we assume that the order and the payment must both be received before the product is delivered.

Referring to Table 1, we see that these prerequisite dependencies can be managed, in part, by *notification* and *tracking* processes. Figure 2 suggests further decompositions and specializations of these processes. A typical order entry system, for instance, specializes both a notification process and a tracking process. When an order entry system prints a packing list of orders ready to be shipped, it *notifies* the packers that the prerequisites for shipping have been fulfilled and it helps managers *track* the orders for which payments have been received but that have not yet been packed.

By developing generic process representations for each of the coordination processes listed in Table 1, and by extending Table 1 to include many more dependencies and coordination processes, it should be possible to concisely represent much of the coordination activity that occurs in organizations as specializations of these generic processes. In addition to its contribution to the process handbook, therefore, this work will also provide an empirically driven opportunity to advance a central theoretical task of coordination theory: identifying and analyzing generic coordination processes.

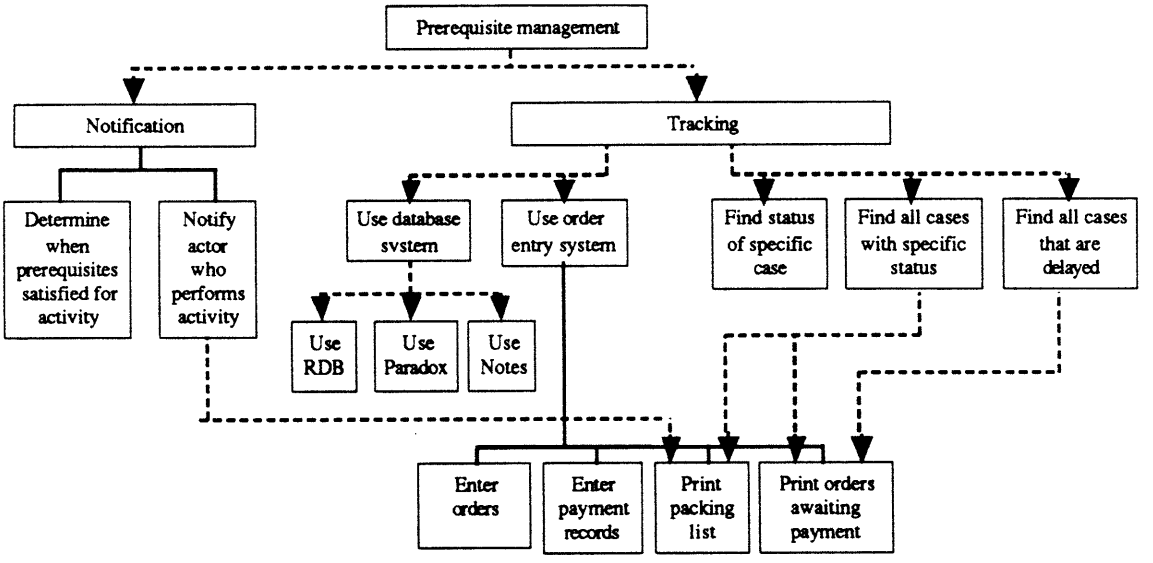


Figure 2. Alternative coordination processes for managing a prerequisite dependency.

Generative representations. Since coordination activities are often those most susceptible to being changed by information technology, a second, and potentially more important benefit of this approach is that it can help us *generate* new possibilities for coordination processes. If we know that, in general, there are several possible coordination processes for managing a given dependency, then we can automatically generate all of them as possibilities for managing that dependency in any new process we analyze. Some of these possibilities may be new or non-obvious, and their generation requires no specific knowledge of the process other than the type of dependencies it involves.

For example, Figure 1 shows prerequisite relationships among the sub-activities of obtaining mailing lists: selecting, ordering, and receiving the lists. Based only upon the existence of these prerequisite relationships, Figure 2 sugests that the designers of this process should consider how to track the status of various mailing lists that have been ordered. Figure 2 also suggests alternatives for how to do this tracking, including various kinds of database systems.

Note that our system would generate only alternatives that were "sensible" according to the constraints reflected in the system, but it could not rule out alternatives that are inappropriate because of other factors. Instead, we hope to organize knowledge so that human users can quickly scan numerous alternatives, all of which have some relationship to the situation being considered, but many of which can be quickly eliminated. By systematically presenting related alternatives, we expect that the system will often surprise its users with possible, but non-obvious, alternatives they might not have considered.

Related work

The project described in this proposal draws upon and has analogies with many strands of work from many different disciplines. In this section, we briefly review a few of these areas.

Process models

One of the simplest and most common ways of representing processes in computer programs and elsewhere is with *flow charts*. Flow charts represent a process as a series of steps with arrows between them representing the order in which the steps can be performed. Some of the steps are decision points, so depending on the circumstances, different sets of steps might be performed. Similarly, a *data-flow diagram* represents the steps of a process but focuses on the ordering relationships imposed by the fact that data produced by some steps must be used by others (e.g., Yourdon, 1989).

A *state transition diagram* or *finite state machine* represents a process in terms of the possible states of the system; the steps taken in the process are the transitions that move the system from one state to another (e.g., Lewis & Papadimitriou, 1981). For example, Winograd and Flores (1986) analyzed the patterns in people's "Conversations for Action" in terms of the states and transitions involved.

Another approach to representing processes is as steps arranged in "strings" like those in *languages* of various sorts. Computer science and linguistics have developed a number of formalisms for representing languages, ranging in power (i.e., the number of lanuages that can be represented) from finite state machines (or regular expressions) to *push-down machines* (or *phrase-structure* or *context-free grammars*) to the most most powerful representation, *Turing machines* (or *context-*

sensitive grammars) (e.g., Lewis and Papadimitriou, 1981). Turing machines are equivalent in power to a program written in any computer programming language.

Multiple actors. To represent processes involving multiple actors, we may want to focus on the interactions between the actors. One approach to doing this is suggested by *Petri nets* (Peterson, 1977) and various representations derived from them (e.g., Holt, 1988; Singh & Rein, 1992). A Petri net is similar to a finite state machine, but allows multiple states to be "marked" simultaneously. Transitions between states may be synchrononized, since multiple states may have to be marked at the same time for a particular transition to occur.

A second approach to representing multiple actors is to represent the process followed by each individual separately, using any of the techniques described above, and explicitly modeling the exchange of information or objects between them. For example, the modelling technique developed by (Crowston, 1991) represents individual actors by programs written in logic. These actors can perform a variety of actions to achieve their goals, including speech actions to change the states of other actors.

Specialization. We also expect to use ideas from Tenenberg's (1986) technique for "planning with abstraction." This technique automatically generates plans (sequences of actions) using information about preconditions and effects of actions at different levels of abstraction in a specialization hierarchy.

Generic taxonomies

Identifying families of processes and their similarities and differences is analogous to developing taxonomies of species in biology. McKelvey (1982) argues that the study of organizations is at a "pre-Linnaean" stage, awaiting a more systematic taxonomy (like the one we propose to develop) to enable further scientific progress.

For instance, researchers in artificial intelligence have begun to identify "generic processes" for tasks commonly done by artificial intelligence programs such as diagnosis, design, and planning (Chandrasekaran, 1983; Clancey, 1983; Marques, et al., in press). An essential idea here is that, if we recognize the commonalities in a whole class of systems, we can understand and develop such systems much more effectively. In particular, the "task-structure analysis" described in (Chandrasekaran, et al., 1992) is very similar in spirit to the approach we are taking. Earlier work by Schank and colleagues also used a similar kind of generic process descriptions (called "scripts"

and "MOPs") to explain how people remember events and how computer programs might do so, too (Schank & Abelson, 1977; Schank, 1982).

Our work is also similar, in some ways, to Lenat's attempt (Lenat & Guha, 1990) to formally represent a great deal of "common sense" knowledge about the world. In our case, of course, the domain of interest is not all of "common sense", but only organizational processes, and we expect to represent much more than just "common sense" about this domain.

Organization theorists have developed a number of taxonomies for organizations. For example, Mintzberg (1979) identifies five basic kinds of organizations: simple hierarchies, machine bureaucracies, functional hierarchies, multidivisional hierarchies, and adhocracies. Others have used technology (Woodward, 1965), formalization (Burns & Stalker, 1961), or other features to create a taxonomy. Following the methodological lead of biologists, McKelvey (1982) proposed a more general approach to the classification of organizations which takes into account not only the current features of an organizational form, but also its historical genesis. However, these taxonomies classify organizations according to very high level characteristics of the organization as a whole, rather than classifying specific processes within an organization. Even those theorists who have classified lower level processes have only done so at an extremely general level. For example, Thompson (1967) identifies standardization, planning, and mutual adjustment as three kinds of processes which deal with varying degrees of interdependence between organizational subunits. While these categories provide some theoretical insight, they offer little guidance as to how one might standardize, plan, or adjust in a particular situation.

More closely related to our approach is the work of Salancik and Leblebici (1988) on generating organizational processes using "grammars" that describe the possible ways a set of tasks can be sequenced. Using restaurants as their example, Salancik and Leblebici examine all of the possible sequences of the basic tasks to be performed (order, cook, serve, eat, and pay), and the combinations of persons who might perform these tasks. Each possible sequence of tasks corresponds to an organizational form (e.g., many cafeterias use the sequence: cook, order, pay, eat). We also plan to create models of real organizational processes, and to suggest alternative processes to accomplish a given objective. We expect our work to extend Salancik and Leblebici's (1988) basic approach by representing processes at varying levels of abstraction, and by identifying generic coordination processes that manage interdependencies between other tasks.

Design tools

Our "handbook" is analogous to design tools in a number of different disciplines. For example, engineers often have engineering handbooks that describe and evaluate different alternative design components for a specific goal. A mechanical engineer, for instance, might consult a handbook to find the density, tensile strength, and other properties of different materials to use in a new device.

Of particular relevance to our work are systems that automatically generate organizational designs or recommendations based on descriptions of the organizational tasks and other factors (e.g., Baligh, et al., 1990; Majchrzak & Gasser, in press). For instance, Baligh, Burton, and Obel (1990) are codifying "textbook" knowledge about organizational design in an "expert system" that will make recommendations based on rules like "If the environment is stable, then a formal organization is appropriate."

Our work differs from these approaches in at least two ways: (1) We are interested not only in providing "conventional" guidance for "traditional" organizations, but also in providing tools to help "invent" new organizations. (2) We are not attempting to provide completely automated advice based on simple input parameters (the traditional "expert systems" approach). Instead, we are attempting to provide conceptual frameworks and partly automated tools to help intelligent people organize and use a large amount of information. That is, we want to provide a "handbook" for use by human experts, not an "automated expert" that tells humans what to do.

Methodology

In the first phase of the project, we are focusing on refining the representational methodology and software support. To do this, we are collecting and representing a relatively small number of process descriptions. Some of these processes are relatively circumscribed and specific (e.g., how software support "hot lines" are organized). Other processes will be broader and more abstract processes as well (e.g., how customer requirements are communicated from marketing to engineering).

As we become increasingly confident that our representational methodology is satisfactory, we expect to collect increasing numbers of examples from different organizations and for different processes. We also hope to involve many more people and organizations in contributing and sharing knowledge through the handbook. Part of our eventual goal will be to "institutionalize" the collection and maintenance of this database so that other organizations can take it over. A possible

analogy for this process, for instance, is the maintenance of the Human Genome Bank (e.g., Frenkel, 1991).

As shown in Figure 5, the primary approach we are using is one of iterative development. That is, we will repeatedly cycle between the following four activities: (a) developing representational techniques, (b) collecting examples of organizational processes, (c) representing these example processes in the handbook, and (d) trial use of the handbook. In some cases, merely trying to represent a process in our handbook will suggest new processes to include or changes needed in the representation. In other cases, trying to use the handbook will also suggest changes or additions needed.

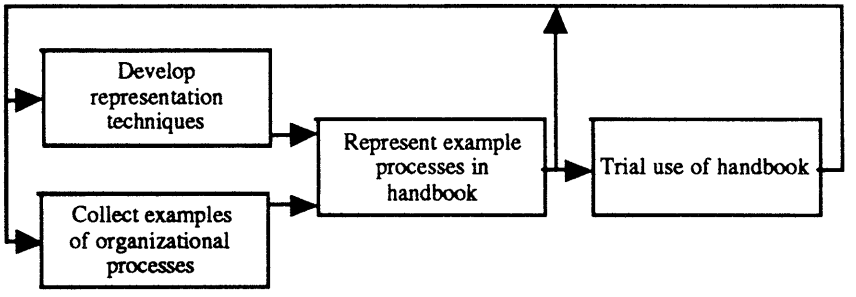


Figure 5. Iterative development process for the handbook.

Current status

Software. The first primitive prototype of the handbook software was developed using the Oval system (Malone, et al., 1992) (a "radically tailorable" collaborative work tool). Later, rudimentary data collection environments were developed using commercial outliners and flow-charting packages on a Macintosh. We are currently implementing a more extensive prototype using the Kappa-PC commercial knowledge-base tool on a personal computer.

Process descriptions. To date, we have collected descriptions of over 150 processes at varying levels of detail. Some of these process descriptions are based on extensive previous fieldwork (e.g., Crowston, 1991; Pentland, 1991, 1992). Others are based on our personal knowledge of other organizations and on several published process descriptions (e.g., Swanson & Beath, 1990).

Conclusion

If this research is successful, it will provide a set of powerful intellectual tools and an extensive database to help invent new kinds of organizations, improve existing organizational processes, and, perhaps, automatically generate software. It will also contribute to developing a central part of coordination theory: the understanding of generic coordination processes and how they occur in organizations. Perhaps most importantly, we hope it will help us understand the possibilities that information technology provides for creating organizations that are not only more effective, but also more fulfilling for their members.

Acknowledgements

Chris Dellarocas is the primary implementor of the current prototype of the handbook software. This work was supported, in part, by Digital Equipment Corporation, the National Science Foundation (Grant No. IRI-8903034), and other sponsors of the MIT Center for Coordination Science.

REFERENCES

Baligh, H. H., Burton, R. M. and Obel, B. (1990). Devising expert systems in organization theory: The Organizational Consultant. In M. Masuch (Ed.), *Organization, Management, and Expert Systems* (pp. 35-57). Berlin: Walter de Gruyter.

Brachman, R. J. and Levesque, H. J. (Eds.). (1985). *Readings in Knowledge Representation* . Los Altos, CA: Morgan Kaufmann.

Brynjolfsson, E., Malone, T., Gurbaxani, J. and Kambil, A. (1989). *Does Information Technology Lead to Smaller Firms?* (Technical report 106). Center for Coordination Science, MIT.

Burns, T. and Stalker, G. M. (1961). *The Management of Innovation.* London: Tavistock.

Carlson, W. M. (1979). Business Information Analysis and Integration Technique (BIAIT) -- The new horizon. *Database, Spring*, 3-9.

Carter, D. E. and Baker, B. S. (1991). *Concurrent Engineering: The Product Development Environment for the 1990's.* Reading, MA: Addison-Wesley.

Chandrasekaran, B. (1983). Towards a taxonomy of problem solving types. *AI Magazine, 4*(1), 9-17.

Chandrasekaran, B., Johnson, T. R. and Smith, J. W. (1992). Task-structure analysis for knowledge modeling. *Communications of the ACM, 35*(9), 124-137.

Clancey, W. J. (1983). The epistemology of a rule-based expert system -- A framework for explanation. *Artificial Intelligence, 20*(3), 215-251.

Crowston, K. (1991). *Towards a Coordination Cookbook: Recipes for Multi-Agent Action.* Ph.D. Dissertation, MIT Sloan School of Management, Cambridge, MA.

Dallemagne, G., Klinker, G., Marques, D., McDermott, J. and Tung, D. (1991). Making application programming more worthwhile. In F. Schmalhofer and G. Strube (Ed.), *Contemporary Knowledge Engineering and Cognition.* Heidelberg: Springer-Verlag.

Frenkel, K. A. (1991). The Human Genome Project and informatics. *Communications of the ACM, 34*(11), 40-51.

Holt, A. W. (1988). Diplans: A new language for the study and implementation of coordination. *ACM Transactions on Office Information Systems, 6*(2), 109–125.

Lee, J. (1990). Sibyl: A tool for managing group decision rationale. In *ACM Conference on Computer-Supported Cooperative Work (CSCW '90).* Los Angeles, CA.

Lenat, D. B. and Guha, R. V. (1990). *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project.* Reading, Mass.: Addision-Wesley.

Lewis, H. R. and Papadimitriou, C. H. (1981). *Elements of the Theory of Computation.* New York: Prentice-Hall.

Majchrzak, A. and Gasser, L. (in press). HITOP-A: A tool to facilitate interdisciplinary manufacturing systems design. *International Journal of Human Factors in Manufacturing,* .

Malone, T. W. and Crowston, K. G. (1991). *Toward an interdisciplinary theory of coordination* (Technical report #120). Cambridge, MA: Massachusetts Institute of Technology, Center for Coordination Science.

Malone, T. W., Lai, K.-Y. and Fry, C. (1992). Experiments with Oval: A radically tailorable tool for cooperative work. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW '92).* Toronto, Ontario.

Malone, T. W. and Rockart, J. F. (1991). Computers, networks, and the corporation. *Scientific American, 265*(3 (Sept.)), 128-136.

Malone, T. W., Yates, J. and Benjamin, R. I. (1987). Electronic markets and electronic hierarchies. *Communications of the ACM, 30,* 484–497.

Marques, D., Dallemagne, G., Klinker, G., McDermott, J. and Tung, D. (in press). Easy programming: Empowering people to build their own applications. *IEEE Expert,* .

McKelvey, B. (1982). *Organizational Systematics--Taxonomy, Evolution, Classification.* Berkeley: University of California Press.

Mintzberg, H. (1979). *The Structuring of Organizations.* Englewood Cliffs, NJ: Prentice-Hall.

Pentland, B. (1991). *Making the right moves: Toward a social grammar of software support hot lines.* Ph.D. thesis, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA.