Using a Process Handbook to Design Organizational Processes

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

Center for Coordination Science (E40-171)
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
dell@mit.edu, malone@mit.edu

Abstract

The goal of the Process Handbook project is to provide a set of theories, methodologies, and tools, to enable the modeling and redesign of organizations in a more systematic way. A key element of the work is a novel approach to representing processes, which uses ideas from computer science about inheritance, and from coordination theory about managing dependencies. This representation improves understanding of complex processes, assists in the identification of process inefficiencies, and facilitates generation and comparative evaluation of alternative processes. We have built an on-line Process Handbook computer tool based on our approach, to represent, store, classify and manipulate business processes. Using that tool, we have developed the beginnings of a systematic design method for process (re)design.

1. Introduction - Project Overview

This paper provides a summary of the Process Handbook project currently underway at the MIT Center for Coordination Science. It focuses on the aspects of the project that are particularly relevant to the field of Computational Organization Design. Finally, it concludes with a brief report on the status of the project.

The goal of the Process Handbook project [Malone93] is to provide a firmer theoretical and empirical foundation for such tasks as enterprise modeling, enterprise integration [Petrie92], and process re-engineering [Davenport93, Hammer93].

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 function as an Organizational-CAD tool and helps (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 [Malone91, Malone94] 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.

2. Process Handbook and Organization Design

One of the principal goals of the Process Handbook project is to develop a method to help (re)design organizational processes in a way which is more systematic than many current approaches. The following paragraphs present a summary of the project aspects that are particularly relevant to this goal.

2.1 Representation of organizational processes

A key to this project is developing novel techniques for representing organizational processes. Our goal is to use advanced process representation techniques in order to:

- enhance understanding of complex processes

- assist process analysis and identification of inefficiencies

- facilitate generation and comparative evaluation of alternative processes

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 contrast to the traditional notion of inheritance, which is organized around a hierarchy of increasingly specialized *objects* [Stefik86, Wegner87], the Process Handbook develops and exploits hierarchies of increasingly specialized *processes*. Specializations of a process are used to indicate alternative ways of performing an activity. This specialization hierarchy,

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

when combined with the usual decomposition hierarchy of a process, produces a powerful mechanism for systematically exploring new alternatives.

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.

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.

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. In that case, the activity inherits the union of decompositions of all 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.

Advantages of process inheritance:
This method of representing processes using a combination of decomposition and alternative specializations has a number of significant benefits over traditional process representation techniques.

Easier representation of new processes. By simply identifying a more general process that the new process is intended to specialize, much of the information about the new process can be automatically inherited and only the changes need to be explicitly entered.

Simplified maintenance of families of process descriptions. Changes made at a high level can be automatically propagated to more specialized processes, thus allowing easy and consistent maintenance of a large number of related process descriptions.

Facilitated comparison and evaluation of alternatives. By explicitly representing alternative processes and their relative strengths and weaknesses, the task of selecting appropriate processes is facilitated.

Enhanced retrieval, combination, and generation of relevant alternatives. 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 propose 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 is the notion from coordination theory that coordination processes can be thought of as ways of managing dependencies among activities [Malone91, Malone93]. Organizational processes can be viewed as containing both *production* and *coordination* components.

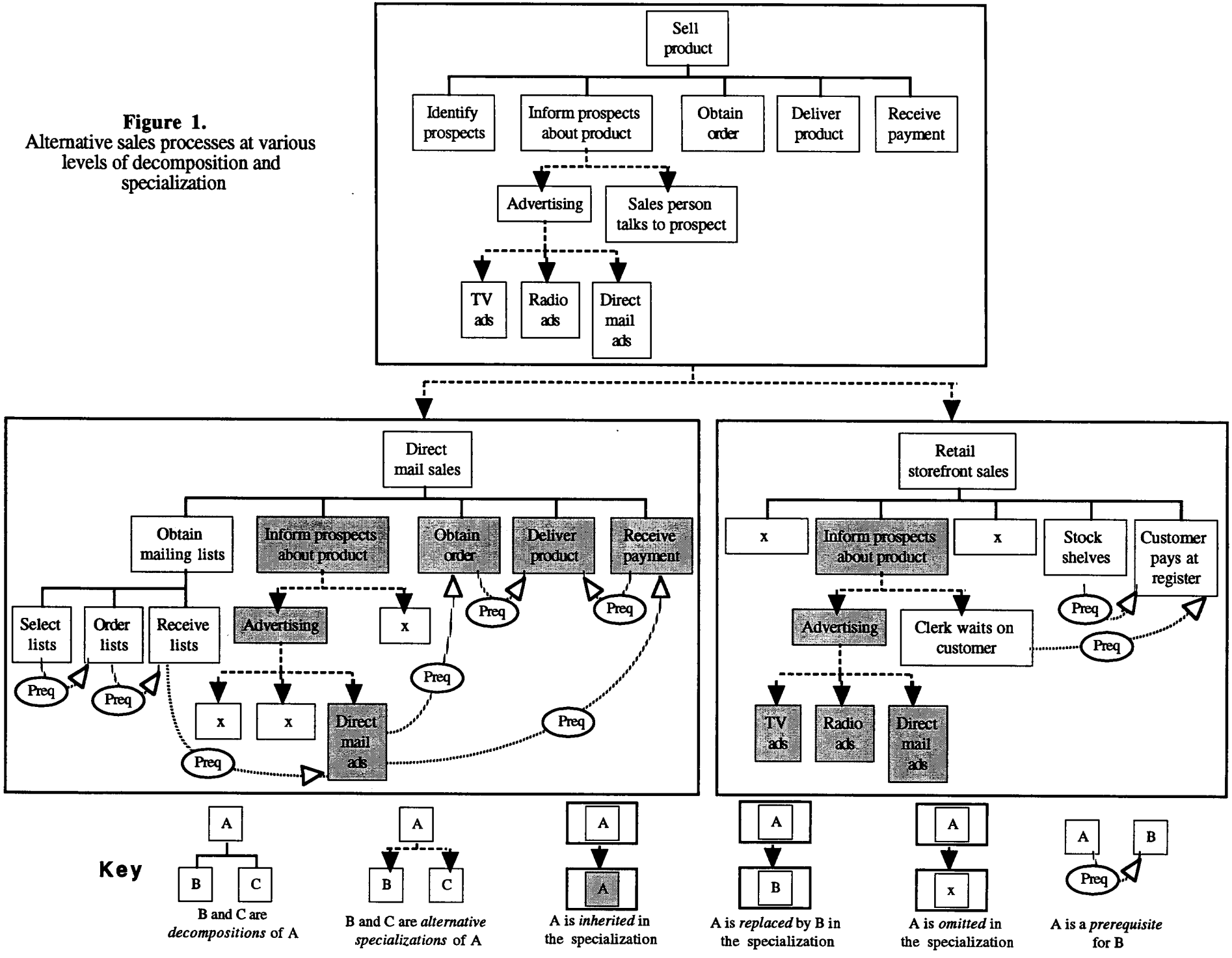
The *production component* includes the process activities that are physically or logically necessary to achieve the stated goals of the process. The *coordination component* consists of the activities necessary to properly manage the *dependencies* among production activities.

Our process representation clearly decouples the two components by identifying the production steps explicitly, and then associating the coordination steps with the underlying dependencies that make them necessary in the first place.

In Figure 1, only the basic production subactivities of the "Sell Product" process appear explicitly. Coordination activities, such as customer order processing, product inspection and shipping, and payment tracking, do not appear in the picture, but are "hidden" behind dependencies that make that coordination necessary. For example order processing, inventory management, order inspection and shipping, are all subactivities of *one way* of managing the Producer/Consumer dependency between "Obtain order" and "Deliver product". Alternative specializations of the Sales process may contain a completely different set of coordination activities to manage the same dependency.

It is possible that a given activity may be viewed as a production and as a coordination component at different

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



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.)

times. For example, writing a manual may be a production component of the documentation process, but a coordination process that manages the usability dependency in the overall context of a product development process. We believe, however, that the distinction between production and coordination is always useful when the context of analysis is fixed.

We are testing the hypothesis that most coordination processes encountered in practice are alternative ways of managing a relatively small set of dependency types. From this perspective, we can build a taxonomy of basic dependency types and associate each dependency type with a specialization hierarchy of alternative coordination processes for managing it. Furthermore, it seems that many coordination processes are applicable across a large range of functional domains. For example, a coordination process used to manage a dependency in a manufacturing process, may be used intact or with minor modifications to manage a similar dependency in a finance process.

Table 1 suggests the beginnings of such an analysis. The table shows a set of common dependencies between activities. together with examples of alternative coordination processes used to manage them.

Note that dependency types themselves form a specialization hierarchy, with more specific types inheriting (and possibly specializing) the set of managing processes of more general dependency types. 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.

Figure 2 shows how a subset of alternative processes for managing a prerequisite dependency could be organized into a specialization hierarchy and integrated in the Process Handbook.

Advantages of activity-dependency representation:
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 benefits are: *enhanced process understanding* and *generativity.*

Abstraction enhances understanding. Coordination activities are often relatively "low-level" (e.g. "Complete requisition form", "Ship packet by courier", etc.), compared to the production activities whose dependency they manage. Traditional process representations, which mix production and coordination activities in the same picture, very fast become cluttered with detail. This hinders understanding the essence of the process, as well as the purpose of coordination activities. Separating coordination activities and "hiding" them behind dependencies, results in simpler representations, which highlight the essence of a process. In addition, it associates coordination activities with their underlying dependencies. Thus, the purpose (or lack thereof) for each coordination activity is made clear.

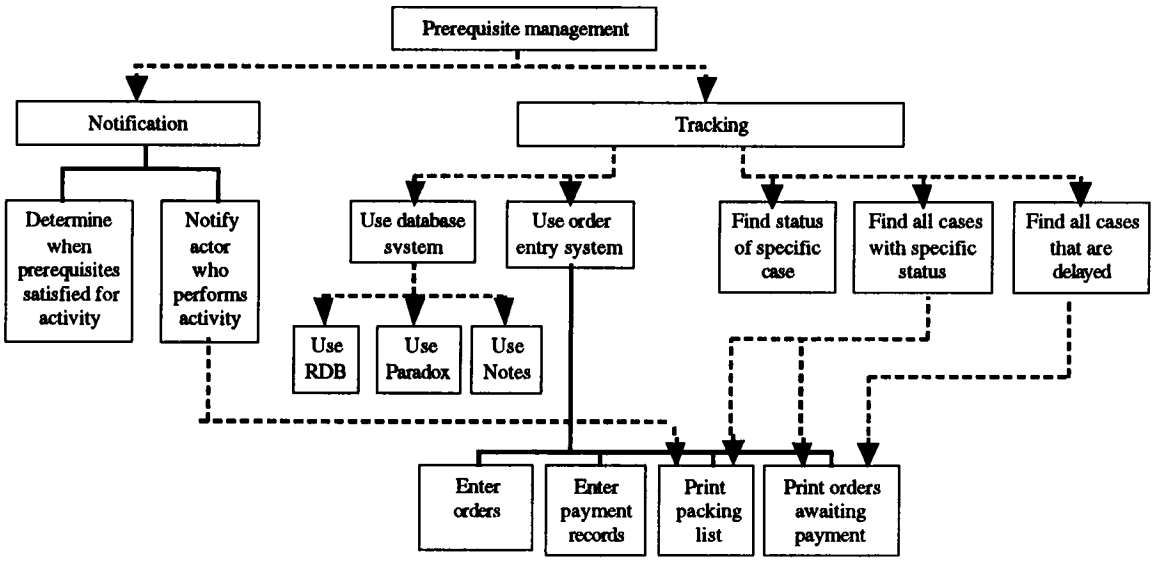


Figure 2. Alternative coordination processes for managing a prerequisite dependency.

Dependencies enhance generativity. As we mentioned earlier, each dependency type is connected to a specialization hierarchy of alternative coordination processes for managing it. The association of an existing coordination process with its underlying dependency, not only improves understanding of the business process under study, but also provides immediate access to a large number of alternative ways of performing that coordination, extracted from business processes in many different domains. This helps produce ideas for generating improved alternatives for the studied process.

2.2 A coordination-theory-based method for process (re)design

The synergy of the two novel aspects of our process representation (process inheritance and coordination theory) is especially important in the capture, analysis and redesign phases of an existing process. Our approach suggests the beginnings of a systematic design method for capturing and analyzing existing business processes and for generating new ones. The method is based on the coordination theory representation of processes and consists of the following steps:

a. Capture existing processes

The first phase in any process redesign project usually consists of capturing the existing situation. Our method of representing processes using a combination of decomposition and alternative specializations can substantially reduce the amount of work necessary for process capture. By identifying a previously stored process that the target process is "similar" to, much of the information about the new process can be automatically inherited. Hence, mapping of new processes can sometimes be an incremental refinement of previously stored models, rather than an exercise that has to begin from scratch each time.

b. Analyze existing processes

- *Separate production activities from coordination activities of existing process.* Given an initial map of a business process, identify which activities capture the "essence" of the process, and which are simply there to manage dependencies among production activities. Although this step requires human judgment and experience, comparison with maps of "similar" processes already stored in the Process Handbook, for which the separation has already been performed, can often provide significant help. We are developing a set of guidelines and heuristics that will help analysts perform this step with accuracy and consistency.

- *Identify underlying dependencies being managed by each coordination activity.* Replace coordination activities with their underlying dependencies. This step produces an activity-dependency view which is uncluttered by coordination details and captures what is, in some sense, the essence of a process. We believe that this representation helps analysts gain insight into the coordination requirements of the process, as well as identify latent dependencies among production activities which may not be addressed in the existing process. Again, comparison with the maps of "similar" processes can assist in performing this step efficiently.

- *Identify unmanaged dependencies, poorly managed dependencies, and unnecessary coordination activities.* Many process inefficiencies can be traced back to missing, poor, or redundant coordination processes. By separating coordination processes and isolating them behind their underlying dependencies, they can be individually examined in a systematic way.

c. Generate alternative processes

- *Redesign production component of existing process.* Use the specialization hierarchy of the process handbook to compare the existing process with possible alternatives and generate ideas about possible improvements.

- *Redesign coordination process for each dependency.* Use the coordination process specialization hierarchy associated with each dependency type to generate ideas for possibly better ways of achieving the desired coordination. Since coordination processes for a given dependency type are collected from many different contexts, this step enables experience gained in possibly very different domains to be used in improving the process under study.

- *Combine production and coordination activities to generate new process.* Replace each dependency with its chosen managing process. Combine production and coordination activities to produce a description of the new process which might be fed to a workflow generator, executed by a computer system, or given to people as a set of job descriptions (We are working towards automating this step).

The above design process reduces the complexity of the analysis and synthesis task by identifying simpler design subtasks in a systematic way. Furthermore, by arranging processes in a specialization hierarchy, the process of finding, comparing, and selecting relevant stored alternatives is greatly enhanced. For example, the specialization hierarchy helps identify, not only alternative processes in similar domains, but also analogous processes in very different domains. Finally, the decomposition structure of processes allows recombination of previously stored process segments to generate entirely new designs.

2.3. A Process Handbook as an organizational-CAD tool

Traditional CAD tools have transformed the process of engineering design, reducing the cycle time of new designs and improving their reliability and accuracy. We believe that the process of organization design can be similarly transformed by the use of *Organizational-CAD tools*. An important goal of this project is to develop a Process Handbook computer tool that supports our representation techniques and the evolving design methodology presented in the previous section. More specifically, our Process Handbook provides (or will provide) support in the following areas:

Process representation and storage. The most basic requirement from a Process Handbook is to support a process ontology based on our representation ideas and to allow easy graphical entry, editing, and viewing of process data, automatic process inheritance, and efficient storage of processes.

Navigation and retrieval. The system should also allow users to navigate within the process specialization hierarchy and easily retrieve stored processes that match, or are "similar" to, a set of criteria using a variety of different ways. The Process Handbook conveniently places "similar" processes close together in the specialization hierarchy and supports navigation using a graphical browser. In addition, desired processes can be located using keyword matching. We are experimenting with other ways of process retrieval, based on comparing structural or behavioral attributes of processes.

Evaluation of alternative processes. Our prototype system allows collecting sets of "similar" processes under bundles and then comparing those processes using tradeoff matrices. Tradeoff matrices display a selected subset of attributes for each process in the bundle. As in the Sibyl system [Lee90], tradeoff matrices can also include detailed justifications for the various settings. In the future, our system will support additional evaluation capabilities, such as dynamic simulation.

Generation of new processes. The Process Handbook can function as a design whiteboard, to allow graphical editing of existing processes, or creation of completely new processes. Processes thus created can be compared and evaluated with other stored processes, and either permanently stored in the specialization hierarchy or discarded.

Translation of designs into enactable forms. The system will provide support for automatically translating a process design from our coordination-theory representation into more directly enactable formats. Such formats may include sets of job descriptions (for manual processes), workflow generator scripts, or maybe even software to drive arbitrary computer systems.

Intelligent assistance. The system will allow users to define various heuristics to be applied to stored or new process designs. The ability to define and execute heuristic algorithms at the user level is very important, since both our design methodology and knowledge about process design are currently evolving. Thus, the Process Handbook becomes a tool, not only for designing new processes, but also for experimenting with new principles of process design. Heuristics could be helpful in locating "similar" processes, identifying process inefficiencies, evaluating candidate alternatives or generating new alternatives.

3. Current Status and Research Directions

Our project is currently progressing on several fronts:

- We are continuing our development of coordination theory, including an evolving taxonomy of dependency types and corresponding coordination processes.

- We have developed a first version of our process taxonomy, which currently includes over 500 processes.

- We have developed several prototype versions of the Process Handbook. Our most recent version is built on top of Kappa-PC and runs on PC-platforms. We are currently experimenting with several alternative environments for our next implementation, including Lotus Notes.

- We are developing a Process Interchange Format (PIF) to facilitate the exchange of process representations among different kinds of systems, such as the Process Handbook, simulation tools, flowcharting tools, and workflow systems.

- Finally, we have initiated a number of field studies to collect process data, test, and refine our methodologies and approach.

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 to enact those processes. It will also contribute to developing a central part of coordination theory: the understanding of generic coordination processes and how they occur in organizations. Finally, 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.

ACKNOWLEDGMENTS: We would like to thank the other members of the Process Handbook team, Fred Luconi, Charlie Osborn, and George Wyner, for their valuable contributions. We would also like to acknowledge the financial support of the MIT Center for Coordination Science and the National Science Foundation (grant #IRI-9224093).

References

[Davenport92] Davenport T.H. *Process Innovation.* Harvard Business School Press: Boston, MA, 1992.

[Hammer93] Hammer M. and Champy J. *Reengineering the Corporation.* Harper Business: New York, NY, 1993.

[Lee90] Lee J. Sibyl: A tool for managing group decision rationale. In *ACM Conference on Computer-Supported Cooperative Work (CSCW '90)*, Los Angeles, CA, 1990.

[Malone91] Malone T.W. and Crowston K.G. *Toward an interdisciplinary theory of coordination,* TR #120, Massachusetts Institute of Technology, Center for Coordination Science: Cambridge, MA, 1991.

[Malone93] Malone T.W., Crowston K., Lee J., Pentland B. Tools for inventing organizations: Toward a handbook of organizational processes, In *Proceedings of the 2nd IEEE Workshop on Enabling Technologies Infrastructure for Collaborative Enterprises,* Morgantown, WV, April 20-22, 1993.

[Malone94] Malone T.W. and Crowston K.G. The interdisciplinary study of coordination, *ACM Computing Surveys*, in print.

[Petrie92] Petrie C., Ed., *Enterprise Integration Modeling: Proceedings of the First International Conference,* MIT Press: Cambridge MA, 1992.

[Stefik86] Stefik M. and Bobrow D.G. Object-oriented programming: Themes and variations. *AI Magazine*, Spring 1986, pp. 40-62.

[Wegner87] Wegner P. Dimensions of object-based language design. In *Proceedings of the Conference on Object-Oriented Systems, Languages and Applications (OOPSLA '87)*, pp. 168-182, Orlando, Fla., 1987: ACM.