SAP DESIGN GUILD

Process Recombination: An Ontology-Based Approach for Business Process Re-Design[1]

By Abraham Bernstein, Department of Information Technology, University of Zürich (note)

Abstract

A critical need for many organizations is the ability to quickly (re-)design their business processes in response to changing needs and capabilities. Current process design tools and methodologies, however, are very resource-intensive and provide little support for generating (as opposed to merely recording) new design alternatives.

This paper describes the 'process recombination,' a novel approach for template-based business process re-design based upon the MIT Process Handbook. This approach allows one to systematically generate different process (re-) designs using the repository of process alternatives stored in the Process Handbook. Our experience to date has shown that this approach can be effective in helping users produce innovative process designs.

Keywords: business process redesign (BPR)

1. The Challenge: Designing Processes

More and more organizations today need to be able to quickly redesign their business processes[2] in response to new needs and capabilities such as rapidly changing markets, new business practices (such as virtual enterprises) and evolving technologies (such as the Internet).

Current BPR methods and tools typically require that one designs these new business processes more or less from scratch, based on an exhaustive analysis of the tasks to be performed. This practice is very resource-intensive, and the support provided by BPR tools is limited to recording in some formal representation (e.g. flow charts, Petri nets) the resulting process diagrams. Little or no support is thus provided for helping users decide what these new process designs actually should be.

The MIT Process Handbook project has taken an alternative "template-based" approach to this challenge (Malone et al. 2003; Malone et al. 1993; Malone et al. 1999). In this approach, one re-designs a process by uncovering its key tasks and task interdependencies, and then systematically exploring the alternatives for these tasks and dependencies stored in the Process Handbook's repository of 'best practices' [3] templates. Our experience to date has shown that this methodology can be effective in helping users produce innovative process designs.

This paper will describe 'Process Recombination', an ontology-based approach that uses this business process redesign methodology on top of the Process Handbook. The remaining sections will give some necessary background on the Process Handbook, describe Recombination itself, evaluate the contributions this work has made, and conclude with a discussion of possible directions for future research.

2. The Process Handbook

The Process Handbook (PH) has been under development at the MIT Center for Coordination Science (CCS) for over ten years, including the contributions of a diverse and highly distributed group of over 40 scientists, students and industrial sponsors (Malone et al. 2003; Malone et al. 1999). The goal of the PH project is to develop a repository and associated conceptual tools that help users effectively retrieve and exploit the process knowledge relevant to their current challenges. The PH is large: it contains, to date, descriptions of over 5000 processes, 12000 entities, and 38000 properties/values. It contains process descriptions ranging from specific examples (e.g. a Mexican beer factory, an automobile parts manufacturer, a university purchasing department) to more generic templates (e.g. for logistics, concurrent design, resource allocation and decision techniques). The PH has been used by a number of practical industrial process designers, partly via a dedicated commercial spinoff (Phios Corp.).

The PH's process ontologies have been shown, at a research level, to be useful in a variety of domains, such as business process reengineering (Bernstein 1998; Malone et al. 1999), e-contracting (Grosof et al. 2002), business process automation (Bernstein 2000a; Bernstein 2000b), software design (Dellarocas 1996), etc. The PH has been influential as well in the arena of industry standards for process representation/modeling: it spawned Process Interchange Format (PIF) (Lee et al. 1996) which in turn spawned Process Specification Language (PSL) (Schlenoff et al. 2000), a draft NIST/ISO standard. In the coming months the process handbook ontology and associated tools will be released to the public as a part of the open process handbook initiative under a novel license agreement that will allow the free use of the software given certain conditions (see http://ccs.mit.edu/phintro.htm for more information). Furthermore a web-based version of the handbook is available for browsing at http://ccs.mit.edu/phintro.htm.

The Process Handbook takes advantage of two simple but powerful concepts to organize process knowledge; *process specialization*, and the notion of *dependencies and their coordination mechanisms*.

2.1. Process Specialization

Practically all process representation techniques (including ours) use the notion of decomposition: that a process can be broken down (or "decomposed") into subactivities. Our representation includes in addition to this the concept of 'specialization'. While a subactivity represents a *part* of a process; a specialization represents a *type* of (or way of doing) the process.

Using this concept, processes can be arranged in a hierarchical structure with very generic processes at one extreme and increasingly specialized processes at the other. As in object-oriented programming, the specialized processes inherit properties of their more generic "parents," except where the specialized processes explicitly add, delete or change a property. Unlike traditional object-oriented programming, however, our inheritance is organized around a hierarchy of increasingly specialized processes (verbs) not objects (nouns). To simplify the representation in this paper we have chosen to show the specialization hierarchy from left to right. Thus, moving left represents moving towards abstract or "deep structure" processes in the specialization tree, moving right represents moving towards more specific "surface structures" in the specialization tree, moving up represents aggregating activities into their parent activities, and moving down represents decomposing activities into their subactivities. In the figures below we will be consistent in displaying activity decomposition on a vertical axis, and specialization on a horizontal axis. As in object-oriented programming, the specialized processes inherit properties of their more generic "parents," except where the specialized processes explicitly *add*, *delete*, or *change* a property. Experience within the process handbook project has reconfirmed findings (MacLean et al. 1990) that people usually tend to have less trouble understanding and using specialization hierarchies that allow this non-monotonic inheritance behavior.

Figure 1 illustrates this special inheritance behavior PH ontology. Here, the generic activity (i.e., process) called "Sell product" is decomposed into sub-activities like "Identify potential customer," "Inform potential customer," and "Deliver Product" (among others). Both "Sell by mail order" and "Sell in retail store," which are specializations of "Sell Product," inherit those decompositions. Some relationships are *modified* in the specialized processes and, therefore, colored gray. In "Sell by mail order," for example "Identify potential customer"is replaced by "Obtain mailing list." For "Sell in retail store" the sub-task "Inform potential customers" was even *deleted* (shown with a red X in Figure).

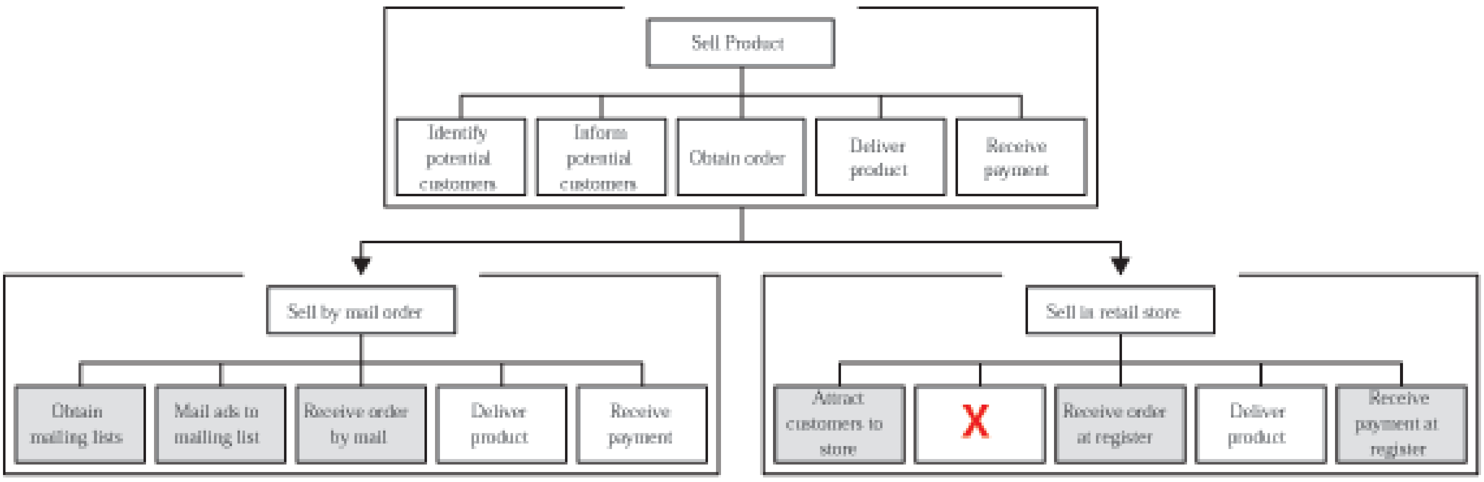


Figure 1: An example of inheritance in the specialization hierarchy (click image for larger version)

2.2. Dependencies and Coordination Mechanisms

The second key concept we use is the notion that coordination can be viewed as the management of task *dependencies* (Malone et al. 1994). Clearly, coordination is only necessary when activities are inter-dependent. Task dependencies can be captured by describing the resource[4] flows between them. Every dependency can include an associated *coordination mechanism*, which is simply a process that manages the resource flow and thereby coordinates the activities connected by the dependency. It is possible to identify a rich library of generic dependency types with associated coordination mechanisms that can be applied to a wide range of business (and other) processes (Crowston 1991; Zlotkin 1995). Multiple candidate coordination mechanisms are typically available for managing any given dependency. For example, there are at least two possible mechanisms to ensure that resources passing through a flow dependency are available at the right time ('prerequisite management'): make-to-order (a variant of which is called just-in-time production) and make-to-inventory (where a stockpile of the product is created in anticipation of future demand). These mechanisms can be applied to almost any domain.[5]

3. Process Recombination

Process Recombination is an approach basing upon a process redesign methodology built on the specialization and dependency concepts described above. This methodology consists of the three key steps (see (Herman et al. 1998) for a more detailed description):

1. *abstract out* the core activities and the key dependencies (i.e. the *deep structure*)of the process you want to (re-) design, using the specialization tree
2. systematically *generate* a set of alternative refinements (i.e. *surface structures*) for the tasks and dependencies in this deep structure model, by 'recombining' existing alternatives for these process components
3. *select* from this set the process(es) that appear to best satisfy your requirements, possibly by using the information stored in tradeoff matrices

We will describe how these steps are accomplished, using the recombination, in the sections below.

3.1. Identifying the Process Deep Structure

The methodology begins by identifying the *deep structure*, i.e. a process model which captures in generalized form the core activities and key dependencies of the process we wish to redesign This maximizes our room for innovation by abstracting away non-essential features. The Handbook supports this via the specialization hierarchy. One can either simply select an existing generic process from the hierarchy if a suitable one already exists, or else one can generate a new generic process, using the hierarchy to identify generalizations for the subactivities in the process you wish to redesign

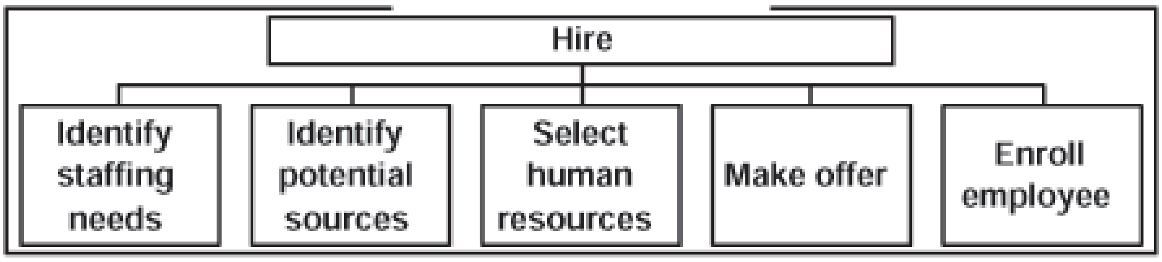


Figure 2: Example Deep Structure for a 'Sell Something' Process

Figure 2 above gives an example of a deep structure model for a 'Hire' process. While the process we wish to redesign may have specified particular techniques (e.g. 'identify potential sources on the Internet') in its subactivities, they have been generalized into more generic activities (e.g. 'identify potential sources') so we can explore alternative ways of achieving those functions. A similar process can be used to generalize activity dependencies and their associated coordination mechanisms.

3.2. Process Recombination

The next step is to find alternative ways (i.e. different *surface structures*) for achieving the generic activities and coordination mechanisms identified in the deep structure model. This is achieved by Process Recombination itself. Recombination consists of two approaches, both of which work by allowing systematic exploration along different process design dimensions; they differ only in which dimensions they address. We will look first at *subactivity recombination*, which treats subactivity alternatives as the dimensions of possible variation. Then, we briefly consider the *bundle recombination* which helps identify process alternatives from within the design space defined by bundles.

Subactivity Recombination: Subactivity Recombination treats alternative specializations for process subactivities as the dimensions of potential variation. We all encounter this type of design in our everyday life. When we make a sandwich, for example, we choose the bread (french, pita, multi-grain, …), filling (cheese, meat, lettuce tomato, sprouts, …) and spreads (margarine, mayonnaise, mustard, …) to compose the snack of our choice. Different selections for each element result in different sandwiches.

In a similar way, the subactivities in a process comprise its basic elements. Since the deep structure model's subactivities have been generalized, we can find alternative ways of performing these subactivities simply by looking at the specializations of these subactivities in the specialization hierarchy. The 'Identify staffing needs' subactivity, for example, has specializations such as 'Identify staffing needs by manager,' 'Identify staffing needs by committee', 'Identify staffing needs by forecasting' and so on. To create a new process design, we need only select one specialization for each subactivity. We can do this multiple times to generate any number of 'combinations' of the process subactivity specializations.

Subactivity Recombination supports this by exploring the subactivities of the deep structure model together with alternatives (i.e. specializations) or each subactivity (Figure 3):

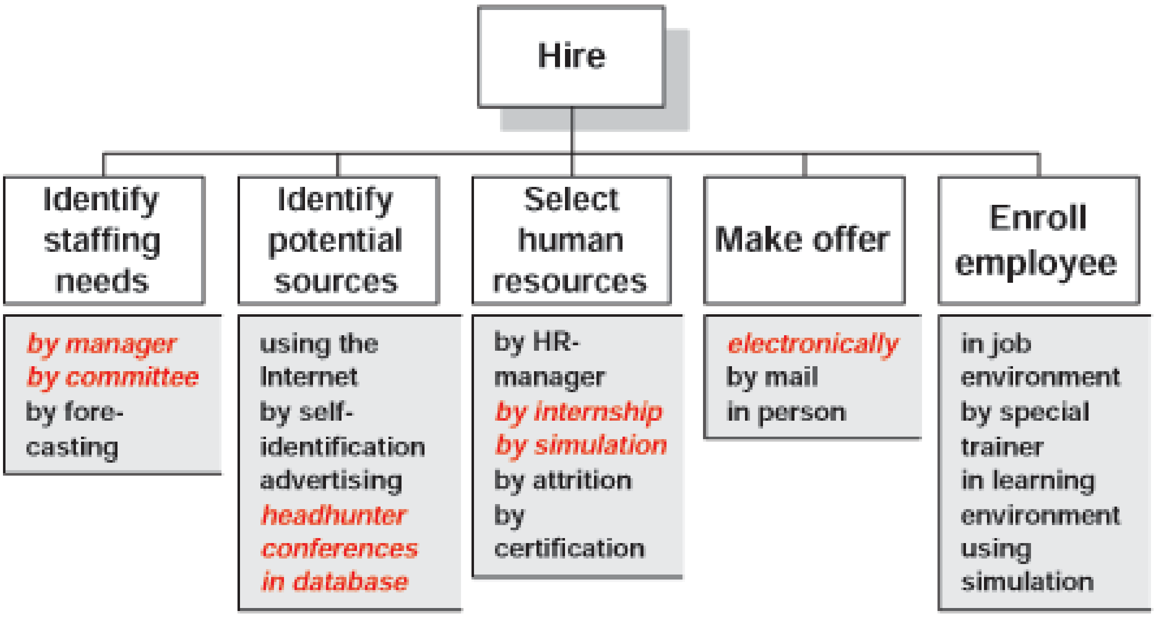


Figure 3: Subactivity Recombination

Every leaf subactivity has an associated list of candidate specializations for that subactivity selected from its specialization hierarchy branch. Each of the candidates can be chosen to replace a subactivity in the overall process. Thus, for the process shown in Figure 3, "Make offer" could be replaced by "Make offer electronically," "Make offer by mail," or "Make offer in person." Generating all the possible alternatives when choosing the highlighted options (in red and italicized) in Figure 3 leads to the twelve different hiring processes shown in Figure 4.

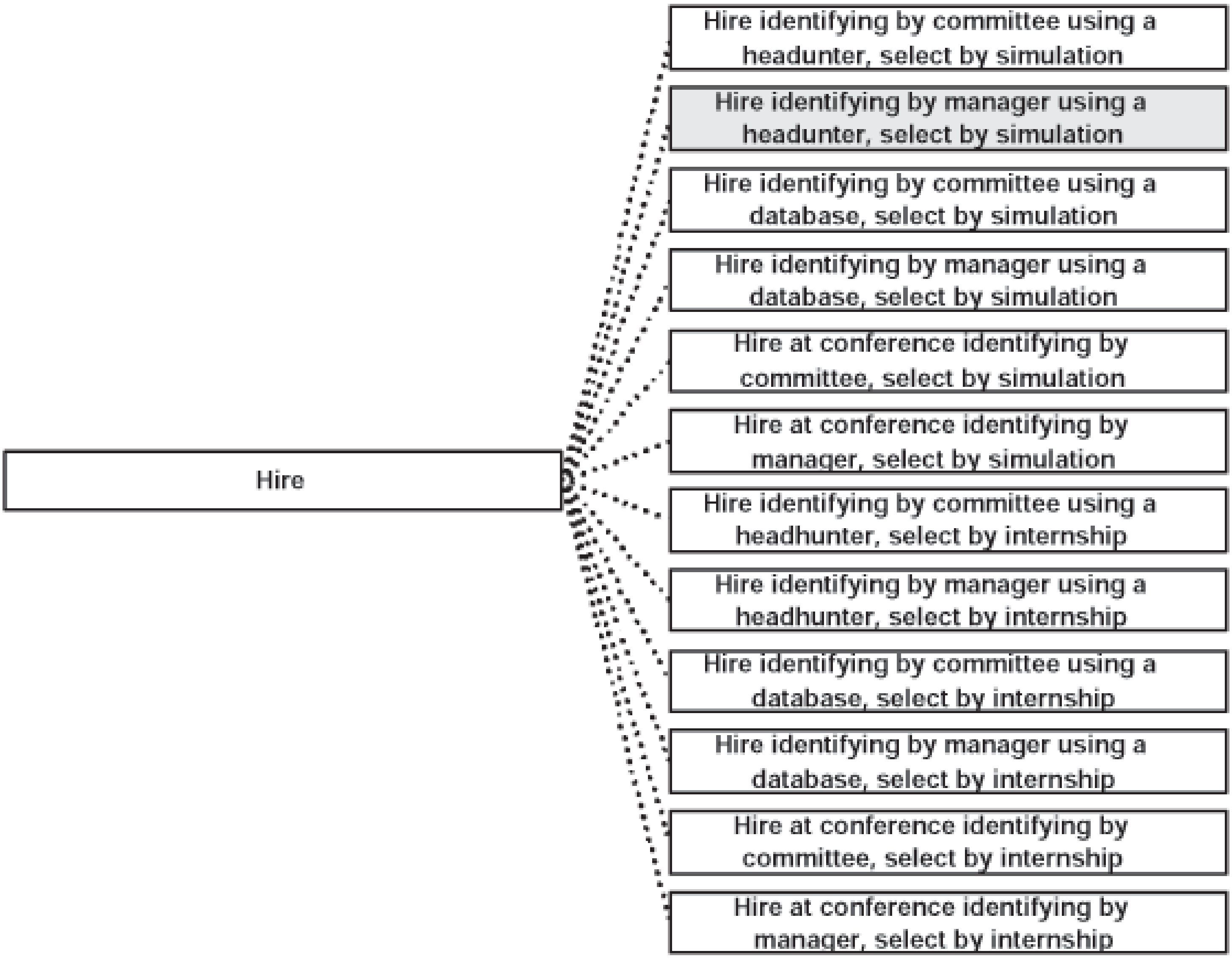


Figure 4: Twelve resulting processes of subactivity recombination

Exploring the shaded process in Figure 4 we find that it combines a managerial identification of staffing needs with a headhunter finding potential job applicants (see Figure 5 for details). The third subactivity "Select human resource" was replaced with "select human resource by simulation". This interesting approach was used within a variety of firms where they let job applicants "act" in a simulation of their job in order to assess their suitability. Apparently, Cessna uses this approach by simulating an assembly line and some executives at BMW were subjected to a simulated office environment for selection (see (Herman et al. 1998) published (partially) in (Malone et al. 2003)).

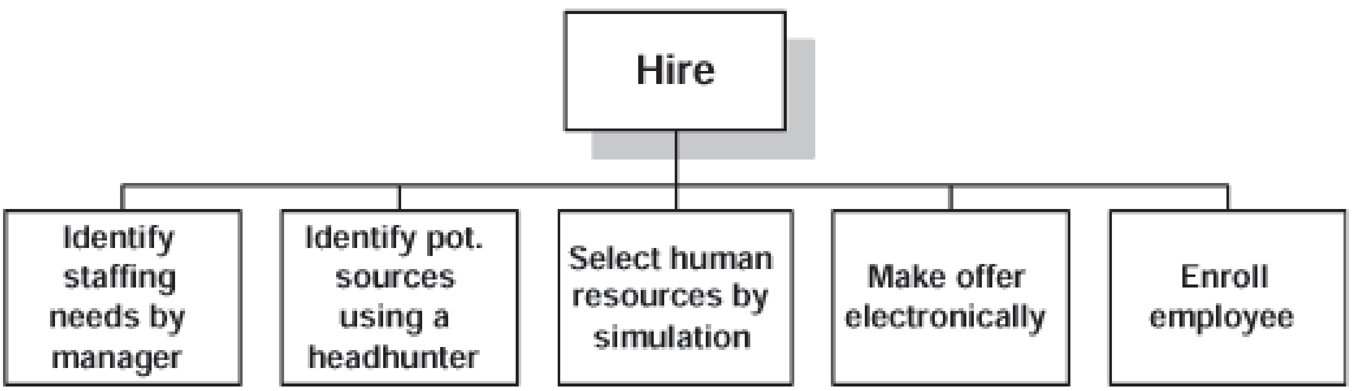


Figure 5: Details of "Hire identifying by manager using a headhunter, select by simulation"

Bundle Recombination: We have found it useful to combine specializations into what we call "bundles" of related alternatives. Figure 6 gives two examples of such bundles in the specialization hierarchy for the "Enroll employee" process, which entails all the details of getting the employee set up in the firm (such as issuing all the benefits, etc.). One bundle ("Enroll when?") collects alternatives for when enrollment is made, while the other ("Enroll by whom?') lists the alternatives of who is helping the new employee to get set up within the firm. Generally speaking, bundles represent different dimensions along which process dimensions can be distinguished.

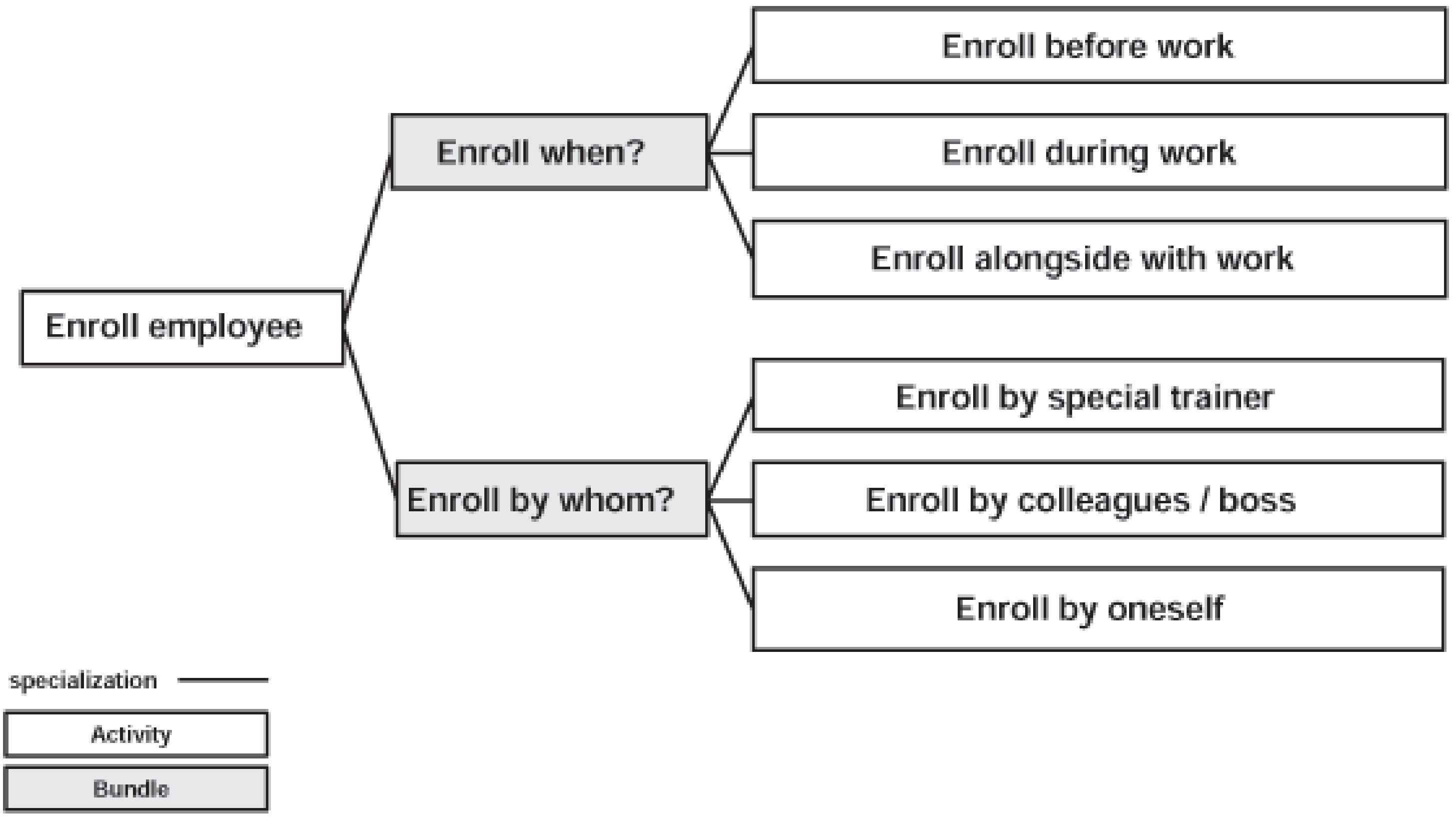


Figure 6: Bundles under "Enroll employee"

Bundle Recombination supports the generation of new design alternatives via exploration of the multiple possibilities defined by the supposedly orthogonal bundles in the specialization hierarchy. The two bundles under 'Enroll employee' therefore define a two dimensional space of possible combinations (Table 1):

		Enroll when?		
		Enroll before work	Enroll during work	Enroll alongside with work
By whom?	Enroll by special trainer	+		
	Enroll by colleagues/boss			
	Enroll by oneself		*	

Table 1: The design space under 'Sell Something'

Each cell in this space represents a possible specialization formed by making one selection from each bundle dimension. The subactivities in the newly created process are derived by multiple inheritance (Lee & Wyner 1995) from the 'parent' processes. This inheritance works as follows: All subactivities that appear in one or both specialization parents are inherited as is. If one parent process has a more specialized form of a subactivity than the other then the more specialized version of that subactivity is inherited. If one parent process has deleted a subactivity that appears in another, or if a subactivity is specialized in different ways in the different parent specializations, then the system asks the user what to do. This all leads to such possibilities as 'Enroll employee during work by special trainer" or "Enroll employee alongside work by colleague/boss." Some options might look unfortunate at a first glance. The recombination marked with the plus sign ("+"), for example, is the standard approach employed by the army (called "boot camp"). The recombination marked with a star is really interesting. It results in a process in which a new employee would enroll himself during work. This is not necessarily stupid. In a Harvard case about the Sun Hydraulics in Florida a new hire went off to learn about the competition for about half a year on full pay to come back and be an extremely productive member of the company (Kaftan et al. 1991).

The Roles of the Different Recombinations: As we have seen, Subactivity Recombination and Bundle Recombination provide orthogonal approach to generate new process ideas. Both approaches can be used in a fully integrated way. Subactivity Recombination is useful when you wish to focus on alternatives for the core activities in your process, and makes it convenient to do so for subactivities at any level in the process decomposition hierarchy. Bundle Recombination allows you to create new process specializations by using bundles as design dimensions. The new specializations created by any Recombination approach can then, of course, be used as alternatives within the other ones.

3.3. Selecting the Re-Designed Process

Once we have used the different components of Process Recombination to produce a number of candidate process redesigns, then the user can make use of a tradeoff matrix to help assess each redesign from the perspective of the criteria that are meaningful to her.

	Cost of recruiting	Strain on Organization	Match	Complexity of selection
Hire using headhunter, select by certification	High	Low	Low	Low
Hire using headhunter, select by simulation	High	Medium	High	High
Hire using manager, select by certification	Medium	High	Low	Low
Hire using manager, select by simulation	Medium	Very High	Low	High

Table 2: Tradeoff Matrix for new process redesigns

The Handbook specialization hierarchy can include, for each process, attributes and associated values that describe the process, in addition to the process subactivities and dependencies. Attributes that are potentially appropriate for comparing the process alternatives are thus inherited by the new combinations just as subactivities are. It is up to the user, however, to determine what the values of these attributes are for each alternative. Once this is done, the new processes and associated tradeoff values are maintained in the Handbook repository as a source of ideas for future users.

4. Contributions

We believe that the Process Recombination fills an important gap in existing BPR techniques and tools. Current process redesign techniques (Davenport 1993; Grover et al. 1995; Hammer et al. 1993; Harrington 1991; Kettinger et al. 1995a; Kettinger et al. 1995b) offer little support for identifying new processes (Kettinger et al. 1997). They suggest how organizations can *organize* their process definition efforts (e.g. using brainstorming, visioning, meeting facilitators) as well as *record* the resulting process designs (e.g. using IDEF or Petri nets), but do not help us actually *generate* new process alternative ideas.

Others have explored the use of re-usable process templates (AT&T Quality Steering Committee 1992; CIO Magazine 1992; McNair et al. 1992; Mi et al. 1993; Schank et al. 1977), abstract process models (Nau 1987; Sacerdoti 1974) and systematic process alternative generation (Salancik et al. 1988). One example notable for its large scale is Andersen Consulting's Global Best Practices Database. Our work is unique, however, in how it combines and extends these ideas, for example by capturing domain-independent coordination mechanisms separately from the more domain-specific core activities.

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 (Baligh et al. 1990; Gasser 1992; Majchrzak et al. 1992). Baligh et al. for example, 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.

The methodology described above has been used to (re-)design several processes; the most substantive example to date has been a collaborative effort with A.T. Kearney Consulting to redesign the hiring processes for a large financial services firm ('Firm A') (Herman et al. 1998). This example was selected because 'Firm A' recognized the need for substantive breakthroughs in the design of its hiring processes, and was willing to devote the time and other resources to explore a new methodology for this purpose. This recognition was driven by business realities that include a shift from low volume to high volumes hires, a shrinking supply of qualified candidates, and a corporate culture of competitive business units that often engage in 'hoarding' as well as 'bidding wars' over good candidates. Our experience is that the methodology was very effective in generating a wide range of novel and promising process design alternatives, including for example such ideas as using 'options markets' approaches to distribute job requisitions to recruiters. Process Recombination was refined after these redesign engagements in order to provide design guidelines for what we understood to be the key components of the methodology.

5. Future Work

The generative strength of our approach is a two-edged sword in the sense that it is often easy to create an overwhelming number of process alternatives. The procedures are not fully formalized, so human judgment is often needed, for example, to select, from the candidates generated by Recombination, the process design appropriate for one's needs. While we do not expect to obviate the need for human judgment, we do plan to explore how tools can further reduce the burden of exploring a large process design space. In the meantime, we believe that it is often preferable to have too many options rather than to few.

6. Acknowledgments

This is a shortened and renewed version of: Bernstein, A., Klein, M., and Malone, T.W. "The Process Recombinator: A Tool for Generating New Business Process Ideas," In the Proceedings of the Informational Conference on Information Systems (ICIS), Charlotte, NC, 1999.

Portions of this paper appeared previously in (Malone et al. 1997).

We would like to thank George Herman and the other members of MIT Center for Coordination Science, as well as Elisa O'Donnell of A.T. Kearney Consulting, for their invaluable contributions to the ideas underlying this paper.

7. References

AT&T Quality Steering Committee "Benchmarking: Focus on World-Class Practices," AT&T Bell Laboratories, Indianapolis, IN USA.

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

Bernstein, A. "The Product Workbench: An Environment for the Mass-Customization of Production-Processes," Workshop on Information Technology and Systems (WITS), Helsinki, Finland, 1998.

Bernstein, A. "How can cooperative work tools support dynamic group processes? Bridging the specificity frontier," Computer Supported Cooperative Work, ACM Press, Philadelphia, PA, 2000a.

Bernstein, A. "Populating the Specificity Frontier: IT-Support for Dynamic Organizational Processes," Ph.D.-Thesis, *Sloan School of Management*, Massachusetts Institute of Technology, Cambridge, MA, 2000b, p. 330.

Bernstein, A., Klein, M., and Malone, T.W. "The Process Recombinator: A Tool for Generating New Business Process Ideas," ICIS, Charlotte, NC, 1999.

CIO Magazine "Back Support for Benchmarkers," *CIO Magazine* (June) 1992, p 16.

Crowston, K.G. "Towards a Coordination Cookbook: Recipes for Multi-Agent Action," PhD Thesis, *Sloan School of Management*, MIT, 1991.

Davenport, T. *Process Innovation: Reengineering Work through Information Technology* Harvard Business School Press, Boston MA USA, 1993.

Dellarocas, C. "A coordination Perspective on Software Architecture: Towards a design Handbook for Integrating Software Components," Ph.D. Thesis, *Department of Electrical Engineering and Computer Science*, Massachusetts Institute of Technology, Cambridge, MA, 1996, p. 288.

Gasser, L. "HITOP-A: Coordination, Infrastructure and Enterprise Integration," Proceedings of the First International Conference on Enterprise Integration Modeling, MIT Press; Cambridge, MA, USA, Hilton Head, SC, USA, 1992, pp. 373-378.

Grosof, B., and Poon, T.C. "Representing Agent Contracts with Exceptions using XML Rules, Ontologies, and Process Descriptions," Submitted Working Paper, MIT - Sloan School of Management, Cambridge, MA, p. 15.

Grover, V., and Kettinger, W.J. (eds.) *Business Process Change: Concepts, Methodologies and Technologies*. Idea Group, Harrisburg, 1995.

Hammer, M., and Champy, J. *Reengineering the Corporation* Harper Business, New York, 1993.

Harrington, H.J. *Business Process Improvement: The Breakthrough Strategy for Total Quality, Productivity, and Competetiveness* McGraw-Hill, New York NY USA, 1991.

Herman, G., Klein, M., Malone, T.W., and O'Donnel, E. "A Template-Based Process Redesign Methodology Based on the Process Handbook," Unpublished Discussion Paper, Center for Coordination Science, Sloan School of Management, Massachusetts Institute of Technology, Cambridge MA.

Kaftan, C., and Barnes, L. "Sun Hydraulics Corporation," 491-119, Case, Harvard Business School, Boston, MA.

Kettinger, W.J., and Grover, V. "Toward a Theory of Business Process Change Management," *Journal of Management Information Science* (12:1) 1995a, pp 9-30.

Kettinger, W.J., Guha, S., and Teng, J.T.C. "The process reengineering life cycle methodology: A case study," in: *Business Process Change: Concepts, Methodologies and Technologies*, V. Grover and W.J. Kettinger (eds.), Idea Group, 1995b, pp. 211-244.

Kettinger, W.J., Teng, J.T.C., and Guha, S. "Business Process Change: A Study of Methodologies, Techniques, and Tools," *Management Information Science Quarterly* (21:1) 1997, pp 55-80.

Lee, J., Grunninger, M., Jin, Y., Malone, T., Tate, A., and Yost, G. "The PIF Process Interchange Format and Framework Version 1.1," Workshop on Ontological Engineering, ECAI '96, Budapest, Hungary, 1996.

MacLean, A., Carter, K., Lövstrand, L., and Moran, T. "User-tailorable Systems: Pressing the Issues with Buttons," Human Factors in Computing Systems, ACM-SIGCHI, Seattle, Washington, 1990.

Majchrzak, A., and Gasser, L. "HITOP-A: A tool to facilitate interdisciplinary manufacturing systems design," *International Journal of Human Factors in Manufacturing* (2:3) 1992, pp 255-276.

Malone, T.W., and Crowston, K. "The Interdisciplinary Study of Coordination," *ACM Computing Surveys* (26:1) 1994.

Malone, T.W., Crowston, K., and Herman, G.A. (eds.) *Organizing Business Knowledge: The MIT Process Handbook*. MIT Press, Cambridge, MA, 2003.

Malone, T.W., Crowston, K., Lee, J., and Pentland, B. "Tools for inventing organizations: Toward a handbook of organizational processes," Proceedings of the 2nd IEEE Workshop on Enabling Technologies Infrastructure for Collaborative Enterprises (WET ICE), Morgantown, WV, USA, 1993.

Malone, T.W., Crowston, K., Lee, J., Pentland, B., Dellarocas, C., Wyner, G., Quimby, J., Osborn, C., Bernstein, A., Herman, G., Klein, M., and O'Donnell, E. "Tools for inventing organizations: Toward a handbook of organizational processes," *Management Science* (45:3) 1999, pp 425-443.

Malone, T.W., Crowston, K., Lee, J., Pentland, B., Dellarocas, C., Wyner, G., Quimby, J., Osborne, C., and Bernstein, A. "Tools for inventing organizations: Toward a handbook of organizational processes," 198, CCS-Working Paper, Massachusetts Institute of Technology - Center for Coordination Science, Cambridge, MA.

McNair, C.J., and Leibfried, K.H.J. *Benchmarking: A Tool for Continuous Improvement* Harper Business, New York NY USA, 1992.

Mi, P., and Scacchi, W. "Articulation: An Integrated Approach to the Diagnosis, Replanning and Rescheduling of Software Process Failures," Proceedings of 8th Knowledge-Based Software Engineering Conference, IEEE Comput. Soc. Press; Los Alamitos, CA, USA, Chicago, IL, USA, 1993, pp. 77-84.

Nau, D.S. "Automated Process Planning Using Hierarchical Abstraction," *TI Technical Journal* 1987, pp 39-46.

Sacerdoti, E.D. "Planning In A Hierarchy Of Abstraction Spaces," *Artificial Intelligence* (5) 1974, pp 115-135.

Salancik, G.R., and Leblebici, H. "Variety and Form in Organizing Transactions: A Generative Grammar of Organization," *Research in the Sociology of Organizations* (6) 1988, pp pp. 1-31.

Schank, R.C., and Abelson, R.P. *Scripts, Plans, Goals And Understanding* Lawrence Erlbaum Associates, 1977.

Schlenoff, C., M., G., Tissot, F., Valois, J., Lubell, J., and Lee, J. "The Process Specification Language (PSL): Overview and Version 1.0 Specification," NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD, p. 73.

Zlotkin, G. "Coordinating Resource Based Dependencies," Working Paper, MIT - Center for Coordination Science, Cambridge, MA.

Notes

Author address: Abraham Bernstein, Department of Information Technology, University of Zürich, Winterthurerstrasse 190, 8057 Zürich, Switzerland

1) This article bases on Bernstein, A., Klein, M., and Malone, T.W. "The Process Recombinator: A Tool for Generating New Business Process Ideas," ICIS, Charlotte, NC, 1999.

2) Here we use the term 'business process' generically, to refer to coordinated activity in any kind of enterprise (government, educational etc.), not just commercial enterprises.

3) It is actually more accurate to call this a database of 'interesting' practices, since typically what is best in one situation is not best in another. The ability to consider trade-offs among various alternatives, culled from a wide variety of industries and functions, can encourage consideration of distant analogies not often considered when looking at typical 'best practice' databases.

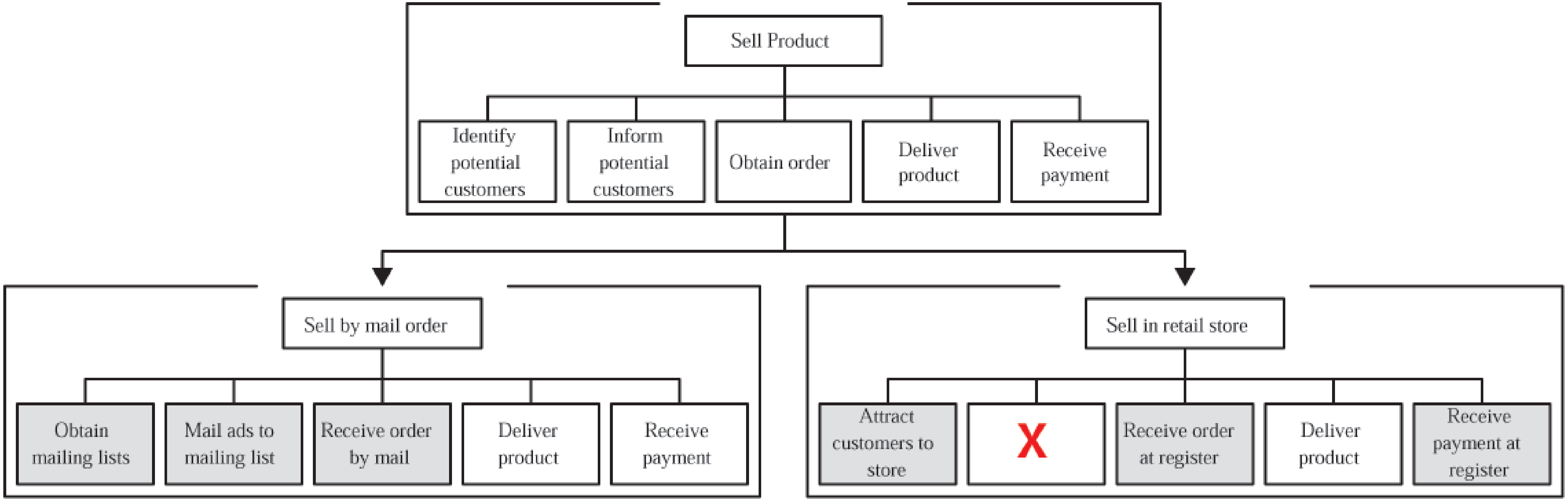
4) A resource can include physical entities (e.g. oil, parts) as well as information (e.g. signals, designs).

5) A detailed discussion of coordination mechanisms is beyond the scope of this paper. Further details can be found in Malone, T.W., and Crowston, K. "The Interdisciplinary Study of Coordination," *ACM Computing Surveys* (26:1) 1994, Malone, T.W.,

Crowston, K., and Herman, G.A. (eds.) *Organizing Business Knowledge: The MIT Process Handbook*. MIT Press, Cambridge, MA, 2003, Zlotkin, G. "Coordinating Resource Based Dependencies," Working Paper, MIT - Center for Coordination Science, Cambridge, MA.

6) In the interest of succinctness this paper omits the discussion of *dependency* recombination, which can be found in the original publication.

top


Sell Product
Identify potential customers
Inform potential customers
Obtain order
Deliver product
Receive payment
Sell by mail order
Obtain mailing lists
Mail ads to mailing list
Receive order by mail
Deliver product
Receive payment
Sell in retail store
Attract customers to store
X
Receive order at register
Deliver product
Receive payment at register