Grammatical Approach to Organizational Design

Jintae Lee
College of Business Administration.
University of Colorado
Boulder CO 80301
jintae@colorado.edu
(ph) 303-492-4149
(fax) 303-492-5962

Brian T. Pentland
Eli Broad Graduate School of Management
Michigan State University
N259 Business Complex
East Lansing, MI 48824-1122
pentlan2@msu.edu
(ph) 517-432-2927

Acknowledgment: We would like to thank Thomas Malone for providing a spark that started this work as well as for his later comments. We also thank George Herman, Mark Klein, and Elisa O'Donnell, whose earlier work provided a starting ground for this work.

Exploring the Process Space:
A Grammatical Approach to Process Design/Redesign

Abstract

This paper introduces a grammatical approach to the design and redesign of work processes. Process grammar allows an explicit representation of the space of possible processes within a given domain, making it possible to search for alternatives more effectively than traditional techniques. Unlike some applications of formal grammar, where complete representation is required, process grammars can be useful even when the representation of the domain is incomplete. We argue that grammatical analysis provides a complementary approach to process redesign, rather than a substitute for existing approaches.

Introduction

Process innovation is a critical component of gaining competitive advantage for modern organizations. While most organizations have made considerable efforts in streamlining existing processes, the challenge of inventing new processes remains. The challenge is particularly salient in the context of electronic commerce on the internet, where new technology has enabled a wide range of innovative business arrangements. For businesses operating on “internet time”, the ability to quickly identify alternative processes and zero in on feasible alternatives is more important than ever.

The literature on process redesign and re-engineering has grown rapidly, starting with the early calls to innovate (Davenport and Short 1990) and “obliterate” (Hammer and Champy 1993). The rhetoric of reengineering created a huge wave of practical and theoretical interest, a large number of methods, tools, and techniques have been developed to assist in the task (Kettinger, Teng, and Guha 1997; Grover and Kettinger 1995a; Grover and Kettinger 1995b; Grover and Kettinger 1997). Stripped of the rhetoric, however, process design is much like any other design problem: one needs to identify the space of alternative designs and narrow it down to a set of feasible alternatives that can be assessed systematically. Sophisticated tools have been developed for analyzing an existing or proposed process, but tools for identifying potential new processes are very limited and ad hoc. This seems unfortunate, because the potential for really significant innovation is constrained by our ability to generate interesting alternatives.

This paper proposes a grammar-based framework for generating alternative processes. In this framework, our knowledge of what is possible or not possible is captured in the form of a grammar that can be used to generate a variety of possible models. We argue that existing approaches can be understood as implicitly using grammatical reasoning, but without taking advantage of the power that a systematic grammatical representation can provide. This is because every process design involves selecting from a space of alternatives, but usually without any explicit recognition of what that space might be. A grammar provides a way to represent our knowledge about the space of alternative processes economically and generatively. Not all processes are feasible or desirable in every situation, but a grammatical framework can make the space of alternatives explicit, visible and searchable. In this way, it has the potential to augment the generation of truly innovative alternatives.

We proceed as follows. First, we review four general approaches to process redesign as identified in the literature. While each general approach is valuable, each has important limitations, as well. Secondly we present an overview of the grammatical approach. Then we discuss each component of the grammatical approach: lexicon, rewrite rules, and constraints. We illustrate each of these components with an example grammar of the sales process. We then conclude by summarizing the contributions of the grammatical approach and describing the direction of future research.

Existing Approaches to Process Design/Redesign

Kettinger, Teng, and Guha (1997) provide an extensive survey of process reengineering practices. Based on this survey, they identify six stages of business reengineering, and propose the Stage-Activity framework: envision, initiate, diagnose, redesign, reconstruct, evaluate. For each of the phases, representative methodologies, tools and techniques currently available are discussed. Of these phases, the redesign phase is the most relevant for the topic of this paper. In Kettinger et al's (1997) framework, this is the phase where the new process is defined and selected from among the alternatives.

Kettinger et al. (1997) identify a remarkable variety of tools that might be applied at this stage, including:

- Creativity techniques (brainstorming, out-of-the-box thinking, nominal group, visioning, etc.)
- IDEF modeling technology

- Process simulation – variation of process variables such as cycle time, queuing times, inputs/outputs, resources.
- Data modeling – data flow diagramming, flow charting, case-based information engineering tools.

Only a few of these tools, however, genuinely address the generation of new design. For example, data modeling helps you analyze a given process, but it does not offer much help in redesigning or creating a new process model. Indeed, as Hammer and Champy (1993) point out, reliance on analytical techniques can have just the opposite effect, resulting in "analysis paralysis." Several of the techniques identified by Kettinger et al (1997) are explicitly oriented toward search (e.g., brainstorming), but none of these techniques provides any assurance that combinations of ideas are being systematically explored.

To understand what we mean by searching the design space, it is worth looking more closely at some typical process design practices. While this list is not exhaustive, it is representative of the kinds of techniques identified by Kettinger et al (1997). These practices provide important points of reference for understanding and assessing the potential value of a grammatical approach.

The "Loose-leaf Binder"

To search for alternatives, one frequently starts from a set of fully formed "cases". These may be documented by a consulting organization (e.g., International Process Benchmark Clearinghouse), or they may be collected as part of a benchmarking study. We call this the "loose-leaf binder" approach because interesting cases or best practices are compiled with simple categorization and keyword-based indexing, at best. For example, typical examples might include “order entry”, “hiring”, “customer service”, and so on. This has the advantage of simplicity, but it is difficult to see how various cases are related.

In terms of our grammatical framework, a binder of cases or benchmarks provides a set of scattered, disconnected possibilities. It is like having a brief phrasebook in a foreign language. With no information about how to construct alternative sentences, one is limited to an impoverished and risky mode of communication: “Point at the sentence that best expresses your intent…” If the phrasebook happens to contain the right sentence for your current situation, and you choose wisely, then everything works out fine. If not, then it is practically useless or worse.

Clearly, it is better to have a phrasebook than not to have one. Benchmarks and other examples have great value because they do convey a set of easily understood possibilities. But we would argue that the use of benchmark examples may lead to a phrasebook (or cookie cutter) approach to process design because it does not support any systematic representation of the underlying possibilities. And in any event, mimicking an existing process at another company is not a fruitful strategy for truly innovative design.

Local Variations

In spite of the rhetoric of radical change, incremental improvement is probably the most commonly used approach to process design (Jarvenpaa and Stoddard 1998). Indeed, Harkness, Kettinger, and Segars (1996) found that many process redesign efforts are based on Total Quality

Management techniques, which adopt an inherently incremental approach (Mizuno 1988). Techniques in this category start with an existing process and vary one or two parameters to create alternatives. The full array of analytical techniques is helpful in assessing the alternatives generated in this way. Indeed, it is particularly easy to assess incremental changes because the rest of the process model can be taken as "fixed."

While these techniques are widely used and unquestionably effective, they are also intentionally limited in scope. When an important process is being redesigned, the risk of failure may preclude any consideration of radical changes. Organizational or institutional constraints may also limit the search for alternatives. The political realities of process redesign efforts may rule out "clean slate" approaches (Stoddard and Jarvenpaa 1995).

To return to our phrasebook analogy, the use of incremental change is like only knowing one, single sentence (the existing process) and a few words that could be substituted. For example, "Where is the …?" is a very useful phrase, especially when coupled with a list of important nouns, such as "bathroom" and "train station." But it is not much help in expressing different ideas, such as "When does the train station open?" Incremental improvement is useful, but it is not a good strategy for significant innovation (Henderson and Clark 1990).

Creative Techniques

Kettinger et al (1997) mention the use of techniques such as brainstorming, nominal group technique, and "visioning." Here we rely on the psychological processes of individuals and groups for generating new ideas. Such techniques are also very important in generating a sense of involvement and buy-in for the implementation of whatever design is eventually adopted. For this reason alone, some kind of participatory technique is probably an essential part of the design process.

While participation is inherently valuable, it seems unlikely that brainstorming (or any other group technique) can provide a systematic exploration of feasible process alternatives. These techniques rely implicitly on the collection of cases available in the minds of the participants. Choosing diverse participants can be helpful because it expands the collective repertoire of possibilities. Still, brainstorming and other group techniques bear a close resemblance to a loose-leaf binder, except there is no index. The various elicitation techniques are designed to create the "index", which consists of whole cases or fragments of cases.

Design from "first principles"

In certain domains of engineering design, such as VLSI design, there are well-established principles for creating a design that meets a given set of specifications (Whitney 1997). Unfortunately, this generalization holds more broadly for the design of artifacts than processes. In other words, one can design a new microprocessor using VLSI tools, but creating a process to manufacture it reliably and economically requires experience, intuition, and to some extent, trial and error.

In the case of business processes, "first principles" are scarce. Most of the relevant knowledge is, to some extent, idiosyncratic to the specific kind of process being designed. While there are certainly some general rules of thumb (e.g., minimize the number of "handoffs"), others seem destined to be broken as new technology changes the rules of organizational design (e.g., "activities with high interdependence should be physically co-located to facilitate coordination"). Thus, while we mention this approach for the sake of completeness, it is not really a viable strategy for business process design.

An architectural analogy may be useful way of understanding these four general approaches. Suppose you want to come up with an innovative design for a building. To generate a good one, you may take one of the following approaches. You might collect a set of designs that you like. You might take one that is good and vary it in different ways to make it work for you. You could rely on creative techniques to generate entirely novel ideas. Local variations can be explored based on a design generated through a creative technique or chosen from a collection. And general principles of structure and function can be applied to any design. All of these approaches are important parts of generating an effective design, and they usually work together. However, if we want to make this process systematic, we need to understand the relationship between the alternatives and how to generate them. As Alexander (1964) has done with his pattern languages for architectural design (Alexander, Ishikawa and Silverstein, 1969; 1977), we need a grammar for process design.

A Grammatical Synthesis

This paper proposes a grammatical approach to process classification and design which we believe can complement and integrate the use of other techniques. This approach is based on the concept of formal grammar, which originated in the field of linguistics (Chomsky 1986) and has been applied to organizations (Salancik and Leblebici 1988; Pentland 1995), mechanical design (Schmidt and Cagan 1997), manufacturing (Baldwin and Chung, 1995), and a range of other topics (Honavar and Slutzki '98, Miclet 1996). Grammar augments the use of other tools by suggesting combinations and alternatives that might otherwise be missed. It provides principled basis for organizing and exploring the cases in loose-leaf binders. It helps us to explore deep, cross-domain, analogies that go beyond local variations. Because a grammar is an inherently generative mechanism, it can be viewed as a framework for systematic out-of-the-box thinking.

Process grammar would provide other benefits as well. It would present a compact way to represent a potentially infinite number of seemingly diverse cases. Research on sentential grammar has helped us to explore the relationship between surface structure and kinds of constraints – linguistic, psychological, and sociological – that govern language use in different situations. We expect that research on process grammar could also help us identify the different types of constraints -- physical, organizational, and technological – underlying the processes we currently observe. By better understanding the origins and constraints on process structure, we can better understand and predict the implications of new technology or new organizational forms (Pentland 1999).

Related work

Pentland (1995) noted the importance of the grammatical metaphor in organization studies (e.g., Weick 1979; Barley 1986) and suggested the possibility of formalizing the metaphor into a more explicit system. Grammatical models seem especially promising because they “provide a natural way of describing the layering and nesting of actions that typifies organizational processes.” The actual application of this approach (Pentland and Reuter 1994) has been limited to a very narrow case (software support hot lines) and in a purely descriptive mode.

The Process Handbook (Malone et al 1999) provides another basis for this work. The Handbook is a product of seven years’ theoretical research, case study, and implementation. It contains a library of several thousand process descriptions organized and categorized in a semantic network based on “part-of” and “kind-of” relations between the processes. The result is an elaborate classification system that can be interpreted as having grammatical structure, but the grammar is implicit. In one application of this work, the structure is used to automatically generate ideas for new processes (Bernstein, Klein, and Malone, 1999), but there is no explicit statement of the “rules” by which new cases might be generated. Chung and Baldwin (1995) have developed a system for manufacturing and design integration that uses a process grammar to represent and generate process alternatives. Their work is focused on the problem of actually selecting particular alternatives, however, in the manner of an expert system. Indeed, there has been a great deal of work on generating process plans from sets of rules (Allen, Hendler, and Tate1990; Bond and Gasser 1988). Like Chung and Baldwin (1995), however, most of this work has sought ways to elaborate the details of a particular process (e.g., how to generate code for a numerically controlled machine tool). While this is certainly an important problem, our emphasis here is quite different.

In this paper, we build on this previous work to offer an example of how a process grammar can be explicitly formulated and applied to the problem of high level process design. Rather than elaborating the details of a particular process, we are focusing on a different problem of generating alternatives. We believe that grammatical methods will help us search the space of possible alternatives more effectively than the existing approaches (loose-leaf binder, local search, or pure creative techniques).

Sales Process Grammar

To illustrate our grammar, we use the example of the sales process. Data for this example is taken from the Process Handbook (Malone et al 1999), which contains a large number of sales related processes. One critical (and exciting) insight here is that the grammar need not be complete to be useful. By allowing the combination and recombination of process components, even a partial grammar can generate interesting possibilities.

Our grammar consists of the usual elements of any phrase structure grammar, each of which is discussed and illustrated below.

- Lexicon (terminal symbols) – John, the, apple, eat.
- Non-terminal symbols – noun phrase (NP), determiner (DE), verb (V)s
- Rewrite Rules -- S -> (DET) (NP) V (NP)

Readers unfamiliar with the concept of formal grammar may find the terminology burdensome and difficult to follow, but the basic idea is very simple. A grammar describes all and only the valid possibilities in some set In the case of natural language, a “sentential” grammar describes the set of all the valid sentences in that language. Since this set is potentially infinite, direct enumeration is impossible. So a grammar represents the potentially infinite set of valid possibilities by generating them from a lexicon and a set of rules. The process of generating valid possibilities is called “re-writing.”

Terminal and non-Terminal symbols

The lexicon of our sales grammar consists of the major steps in a sales process, as shown in Table 1

Table 1. **Terminal and Non-Terminal Symbols in the Sales Grammar**

Terminal symbols (lexicon)	Sell, Identify-Potential-Customer , Identify-Customer-Needs, Inform-Potential-Customer, Take-Order, Deliver-Goods, Receive-Product, Maintain-Customer-Relation
Non-Terminal symbols	Act, agent, location, temporality, object, beneficiary, instrument, how, and all the terminal symbols

The lexicon of our sales grammar consists the specific activities such as Sell, Identify Potential Customer, Identify Customer’s Needs, Information Potential Customers, and so on. They are the basic building blocks of the process as it is modeled in this grammar. There are also non-terminal symbols, which can be further elaborated or specified. The non-terminal symbols consist of generic categories such as Act, agent, object, location, and temporality, as well as all the terminal symbols.

In formulating this grammar, we have taken the unusual step of including the “terminal” symbols as non-terminals, for the following reason. What we put into lexicon or the terminal symbol set depends on what we want the surface structure to be. In sentential grammar, sentences that we can utter constitute the surface structure. The surface structure in turn depends on when we want our grammatical analysis to stop. We want to stop rewriting when all the symbols are terminal.

In the context of a process grammar, there is no specific stopping point for the level of detail or abstraction. In principle, one can proceed to an infinitesimal degree of refinement (Abell 1987). But at some point, the practical value is exhausted. We want our rewriting to stop when we have interesting processes that are concrete enough to start assessing and comparing. Therefore, if we have a specific level of analysis in which we are interested and if we can characterize this level with a specific set of vocabulary, then we can define this set to be the lexicon, or terminal symbols. For example, if we wanted to see how a given process model gets implemented in a workflow system, then the lexicon would consist of the constructs for the chosen workflow system.

More often than not, however, there is no fixed level that can be characterized in terms of a set of specific symbols. The interesting level could be quite abstract: “Sell via the Internet.” Or it might be quite specific: “Sell support service by establishing a web-based help desk staffed on a rotating basis by members of the development department…”. And naturally, much more specific details would be required to actually implement the process. But regardless of the level of abstraction, some of the same terminology is typically used. Because we have allowed terminal symbols to appear as non-terminals, different levels of abstraction can use the some of the same vocabulary. The action SELL, for example, can appear as the initial string as well as at the surface structure we end up with.

Rewrite Rules

Our process grammar includes a set of rewrite rules that allows us to generate the set of possible processes from a completely generic sales process (the "deep structure"). These rules are of two kinds: a decomposition rule and a set of domain-specific rules. The generic decomposition rule is generic; it is based on a framework proposed by Fillmore (1968) and can be applied to any action or process. The domain-specific rules, however, reflect knowledge about a variety of alternative sales processes.

Generic Decomposition Rule. The core of our grammar is a generic decomposition rule that can be applied to any process (or part of a process) any number of times, including zero times. This rule can be expressed formally as:

Act → [Act [agent | locality | temporality | object | beneficiary | instrument | how]]*

The rule expresses the fact that there are a number of different cases that can be explored and potentially specified for any given action. This rule specifies that any activity (Act) can be rewritten as itself plus some additional information that makes the description of the activity more specific. The alternatives for what kind of additional information is needed are called "cases" (Fillmore 68). For example, if our action is "Inform-Potential-Customer" (IPC), this rule says that we can re-write in any of the following ways

IPC → IPC agent

IPC → IPC locality

IPC → IPC agent IPC locality

IPC → IPC agent IPC locality IPC temporality

In grammatical terms, “agent”, “locality” and “temporality” are non-terminal symbols that will be rewritten further with domain-specific rewrite rules, which are discussed below. In practical terms, they are aspects of the process design that need to be explored and finalized. For example, if one is selling vacation packages to the Bahamas, one might consider running television advertisements in Northeastern United States during the early January. To express this possibility in terms of re-write rules, we would need to include rules such as: agent → TV, locality → Northeastern US, temporality → first week of January, object → advertisement for the Bahamas. Thus, each of these cases can be elaborated into some specific aspect of the activity.

There are many proposed sets of cases (Chase 1970; Fillmore 1968; Fillmore 1975; Grime 1975; Simmons 1973). Which set to adopt would depend on the domain and the purpose of analysis. For our purpose of exposition, we have chosen to include cases that correspond most closely with Fillmore. (Fillmore 1968): agent, object, location, temporality, instrument, result, beneficiary, how, The first four roughly correspond to Who, What, Where, and When. These are essential to the specification of any real activity.

Not all cases need to be fully elaborated. The instrument and beneficiary are less universal but illustrate activity-specific cases. That is, for some activities, it might be valuable to know the instrument with which the activity was accomplished and for whom it was important (e.g., the customer, the employee, and so on). For other activities, this information may not be important. The application of the decomposition rule is, in this sense, optional. The choice of whether to elaborate some aspect of a process description depends on the practical needs of the situation.

Domain-specific rules. Domain-specific rules rewrites a current process design by elaborating on the cases associated with the process. These cases represent the different aspects of the deep structure (Winograd 1981) such as the agent, object, location, temporality, and instrument. As in the example of selling vacation packages, one might have a large body of data and experience about geographic regions, market segments, seasons, destinations, and alternative advertising media. It is worth remembering, of course, these dimensions comprise only one step in the overall sales process: Inform-Potential-Customers. Each of the other steps may entail similarly rich domain knowledge.

The case, “how”, is qualitatively different from the other cases and may embody a particularly rich set of domain specific knowledge. This is because answers to “how” questions tend to involve a combination of the other case slots. For example, when asked how potential customers were identified, the following are perfectly good answers:

- Identify potential customers on sales call
- Identify potential customers through tele-sales
- Potential customers identify themselves
- Obtain mailing list of potential customers

By describing different ways of executing an activity, each “how” case can be seen as a page in a “loose-leaf binder”. This case allows us to treat a complex of other cases closely intertwined such that it is better to treat them as a unit. It also acts as a temporary holder of such cases until they are sorted out into individual cases.

Case independence. For the purposes of this paper, we assume that the case alternatives are independent. That is, all combinations of cases are considered to be legal alternatives. In special cases where this is not true, domain-specific rules can be added to express the constraints that prevent certain combinations of cases, such as selling liquor across state lines on the internet. We discuss the more general issue of introducing constraints into the grammar later in the paper.

“How” offers an interesting exception to case independence, however. Each alternative case of “how” holds a combination of different slot values encompassing specific cases for who, what, when and where. Thus, the “how” cases may hold important clues about the implicit constraints that may be operating in the domain. If we find certain combinations of cases and not others, it may indicate that some technical or physical constraint is limiting the valid combinations.

The assumption of independence simplifies the representation, but it is not necessary for the grammatical structure outlined here to be useful. The main issue is the extent to which one relies on automatic processing (versus common sense) to identify and rule out erroneous (invalid) alternatives. By assuming independence, the current representation will tend to generate more than just the valid combinations – it will generate some invalid ones, as well. While this would be a flaw in a formal system intended to express only the valid combinations, it might provide an interesting aid to creativity. For example, it may have seemed implausible to sell large durable goods (like furniture) or professional services (like medical diagnosis) over the internet, but these models are now widely accepted as plausible (if not always economical).

What about "why"? Since we include who, what, when, where and how, some readers may wonder about the exclusion of "why" as an explicit case. After all, processes are often described or classified in terms of their goals, so "why" seems like natural case to include. However, unlike the other dimensions, why is not inherent in the activity itself. A given activity can be used for many different purposes, serially or in combination. "Why" resides in the external relation between the activity and other factors. Treating why as an attribute of an activity unnecessarily limits the potential applicability of the activity.

ADVANTAGES OF THE GRAMMATICAL APPROACH

The potential benefits of a grammatical approach can be seen mostly clearly when contrasted with direct enumeration (i.e., listing the alternatives in a loose-leaf binder). Table 2 shows the wide range of the sales processes obtained from primary and secondary sources in the context of the Process Handbook (Malone et al. 1999). Note that this is only a brief, partial enumeration of the universe of possible sales processes. Like a loose-leaf binder (or the results of a brainstorming session), the descriptions in this list are unstructured. For example, items at vastly different levels of abstraction are included side by side, the product being sold is conflated with the other aspects of the process, and so on.

When confronted with a long list of alternatives, the analogy to the foreign language phrasebook becomes even more compelling. We can easily get bewildered by the alternatives in a list like this and searching among them is not straightforward. Instead of just showing unorganized lists like the one in Table 2, the Process Handbook organizes the alternatives into a tree-structured taxonomy with various "bundles" or groups of related activities, for each of independent dimensions represented. A grammatical approach formalizes this approach by explicitly representing some principled regularity behind all the variations. In doing so, it can provide a better understanding of what is really responsible for these variations. In other words, it can be helpful to understand the design space and be able to even generate new alternatives.

Table 3 contains an example of a grammar that is loosely induced from the examples given in the previous table, and the structures used to organize these examples in the Process Handbook, augmented with general knowledge about the sales process (e.g., forms of payment). The first row shows the seven generic steps in the sales process which form the "deep structure" that should be included in every sales process. The generic decomposition rule for actions appears next, followed by a set of rules that allow us to expand the deep structure into an infinite variety of alternative sales processes. These include domain specific rules that define the lexicon (e.g., kinds of products and services), as well as the rules that outline the alternative ways to accomplish each of the seven parts of the sales process. Note that some of these rules are "context free" – they have a single token on the left-hand side and can be applied whenever that

Table 2. **Variations of the Sales Process**

Sales by subscription retail store sales mail-order sales sell advertising space/time sell by free-trial sell names sell utilities sell shrink-wrapped software sell expertise sell via auction/reverse-auction sell through virtual mall sell at garage sale sell via hierarchical marketing sell via classified ads sell classified ads sell repair services sell installation services sell software services sell cataloging services sell service by giving away product sell product by giving away product sell through sales force sell customized product sell customized service sell by email sell by fax sell via distributor sell via broker sell direct	sell to consumers sell to businesses sell extractor service (WSJ) sell transportation service, (FedEX) sell ads delivered to prospects via email sell coupons displayed on extractor’s web pages sell untargeted advertisement sell ads targeted by content sell ads targeted by search query keywords sell ads targeted by web directory category sell ads targeted by placing on related pages sell ads targeted by user profile sell ads targeted by specified digital community sell ads targeted by demographics of audience sell ads targeted by the IP domain of the client sell ads targeted by technology used by the client sell links from extractor to sponsor sites sell digital storefront creation/maintenance services sell space for displaying discount coupons sell standard item from stock sell custom item to order sell standard item to order sell business to business e-commerce sell display ads on extractor’s web pages sell banner advertisements sell real estate sell capital investment loan sell telephony service sell wireless service

token appears. Other rules are context sensitive -- they contain two (or more) tokens on the left-hand side and can be applied only when those tokens appear as a group. The inclusion of context sensitive rules allows us to elaborate certain parts of the process selectively.

Table 3 Rewrite Rules for the Sales Process Grammar

Generic template for sales process ("deep structure"):

1. Sell → Identify-Potential-Customer Identify-Customer-Needs Inform-Potential-Customer Take-Order Deliver-Goods Receive-Product Maintain-Customer-Relation

Generic decomposition rule:

2. Act → [Act [agent | location | temporality | object | beneficiary | how]]*

Generic Sales Lexicon rule:

3. agent → by customer | by seller | by network-of-experts | by computer-agents | by store | by sales company | by marketing-research-firm | by customers | by broker | by delivery-company
4. object → product | service | needs
5. temporality → before event | after event | during event
6. event -> register | buying | delivery | order | malfunction
7. beneficiary → for customer | for distributor | for manufacturer | for vendor | …
8. how → virtually | physically
9. service → consulting-service | training-service | customer-service | repair-service | diagnostic-service |…
10. needs → repair-needs | remotely-detectable-needs | .. …
11. location → customer-site | manufacturing-site | need -site
12. communication-medium -> phone | fax | email | web | face-to-face
13. payment-method -> cash | credit-card | check

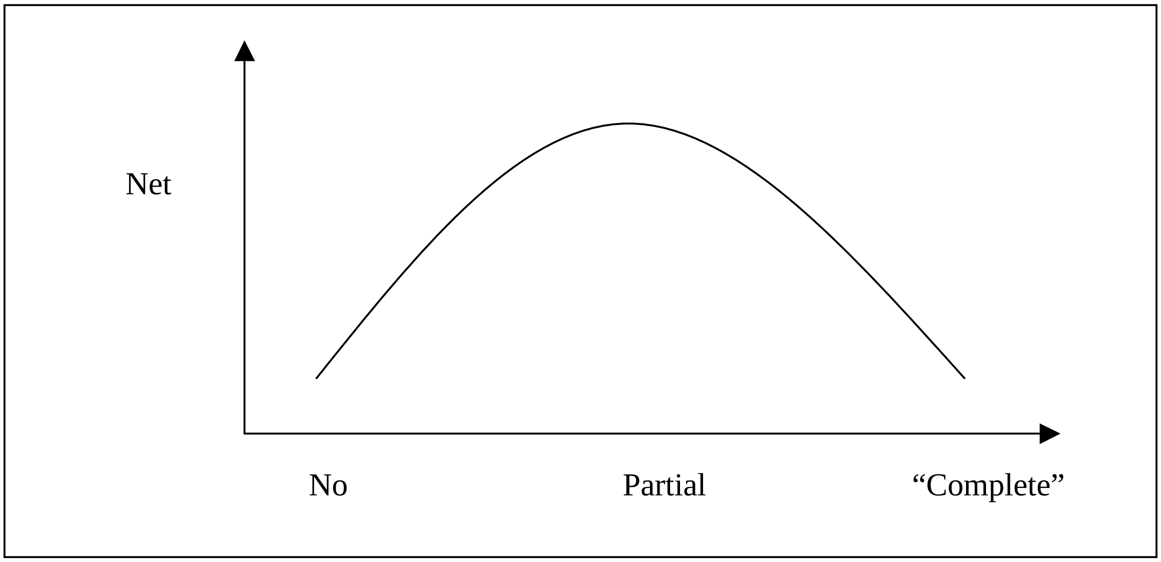
Context-Sensitive Rules

14. Identify-Potential-Customer object -> IdPC product-information | IdPC promotion-information | IdPC recall-information
15. Identify-Potential-Customer (IdPC) how → IdPC by-data-mining| IdPC through-magazine-database | IdPC through self-identification
16. Identify-Customer-Needs (IdCN) how → IdCN via-telephone-survey | IdCN via-usage-patterns | IdCN via-web-survey
17. Inform-Potential-Customer (IPC) how → IPC via communication-medium
18. Deliver-Goods (DG) how → DG by-geographical-proximity
19. Receive-Product (RP) how → RP payment-method
20. Maintain-Customer-Relation (MCR) how → MCR by-providing-initial-free-service
21. Maintain-Customer-Relation (MCR) by-providing-initial-free-service → MCR by-providing-initial-diagnostic-service-for-free
22. Deliver-Goods (DG) how -> DG by geographical-proximity

The process grammar also provides modularity because as we discover additional alternatives, we can just add to the rewrite rules.   Adding a new rule is likely to generate some spurious new combinations, but it may also generate valuable new possibilities.

The grammar in Table 3 is meant to be illustrative, not definitive. In fact, one of our key claims is that partial grammars, such as this one, can be effective tools for exploring the process space. Inducing a complete, correct grammar for the universe of possible sales processes is probably an impossible task. The main reason is that for each one of the activities in Table 3, there are a potentially very large number of plausible alternatives. Each of these alternatives could, in principle, be elaborated down to a very fine-grained level of detail (e.g., lines of code in a program). As a demonstration of the infinite variety of human activity, this makes an interesting thought experiment, but it does not seem like a very practical research goal. Figure 1 suggests a trade-off between the completeness of the grammar and its likely utility as a practical tool for process design. At some point, additional effort on the grammar is likely to reduce the overall net benefit.

Figure 1 Effort/Return Tradeoff



To justify our claim of a potential net benefit, it is worth returning to our generic process design strategies: the loose-leaf binder, local variations, brainstorming, and top-down. How could a grammatical approach help to augment each of these practices?

We believe that the grammatical approach espoused in this paper provides a principled framework for integrated use of these existing approaches. Like the loose-leaf binder approach, it leverages the existing practices by expressing them in terms of the rewrite rules, but by doing so, it provides dimensionality to this knowledge along the different cases. Unlike the loose-leaf binder approach, the representation is compact and makes salient the similarity and the differences among different variations. Like the local variation approach, it allows us to explore variations along a dimension at a time, but the variation can be cross-domain as in the case of context free rewrite rules, thus providing some of the functionality of the creativity tools. Like the top down approach, the decomposition rule specifies how the generic structure of the sales process can be broken down into its components, each of which can be further specialized along the different cases and/or decomposed into its own components. Unlike the top down approach, however, the domain-specific rewrite rules allow us to explore combinations of existing practices as discussed above.

Generating a new process: an example

To be useful as a tool for design or redesign, the grammar must be able to generate new processes. By systematically applying some of the rewriting rules of a sales process at some generic level, we can generate and explore new designs of the sales process.

Suppose we are interested in a start up e-commerce business and want to explore innovative business models. Table 4 shows one possible derivation of the grammar in Table 3. Since the search space is infinite, one cannot exhaustively enumerate the choices and rank them using some criteria. Rather, we started with the basic rules in the grammar and used our judgment to decide which ones to apply. The grammar cannot make the search space smaller, but it can provide some systematic structure to the search process. Simply identifying the seven core dimensions in the decomposition rule is a significant benefit, for example.

This derivation roughly corresponds to the following steps. First you decompose the sales process and look at individual components (Steps 1 and 2). You know that you want to sell services (vs. products) that satisfy some customer needs. So you specialize Take-Order (TO) object to Take-Order service and Identify-Customer-Needs (IdCN) object to Identify-Customer-Needs needs (Steps 3 and 4). You decide to focus on repair services and hence on repair needs because you feel that many types of repairs (e.g. plumbing, refrigerator, heater) are causing much more inconveniences than can be avoided with recent technologies. Therefore you rewrite *needs* to *repair-needs* and *service* to *repair-service* (Step 5). In particular, one of the services that you want to provide is to inform the customer about the repair need before any malfunctioning arises (Steps 6, 7, 8, and 9). Now you want to focus on how to inform such needs before malfunctioning (Step 10). You have been intrigued by the idea of Application Service Provider, where the customer's need is remotely satisfied by an application running on a server (Lee and Krappe 2000). Therefore you rewrite *Identify-Customer-Needs how* to *Identify-Customer-Needs remotely* (Step 11). Assuming that the need can be so detected, you then

Table 4. An Example Derivation of a Sales Process from the Sales Grammar

Legends:
IdPC: Identify-Potential-Customer, ICN: Identify-Customer-Needs, IPC: Inform-Potential-Customer
TO: Take-Order, DG: Deliver-Goods, RP: Receive-Payment, MCR: Maintain-Customer-Relation

1. **SELL** [1]

➔ **2. IdPC ICN IPC TO DG RP MCR**

➔ 3. IdPC **ICN object** IPC **TO object** DG RP MCR [2,2]

➔ 4. IdPC ICN **needs** IPC TO **service** DG RP MCR [4,4]

➔ 5. IdPC ICN **repair-needs** IPC TO **repair-service** DG RP MCR [10,9]

➔ 6. IdPC ICN repair-needs **IPC object IPC temporality** TO repair-service DG RP MCR [2,2]

➔ 7. IdPC ICN repair-needs IPC **needs** IPC temporality TO repair-service DG RP MCR [4]

➔ 8. IdPC ICN repair-needs IPC **repair-needs** IPC **before event** TO repair-service DG RP MCR [10, 5]

➔ 9. IdPC ICN repair-needs IPC repair-needs IPC before **malfunction** TO repair-service DG RP MCR [6]

➔ 10. IdPC ICN repair-needs **ICN how** IPC repair-needs IPC before malfunction TO repair-service DG RP MCR [2]

➔ 11. IdPC ICN repair-needs ICN **remotely** IPC repair-needs IPC before malfunction TO repair-service DG location DG agent DG how RP MCR [8]

➔ 12. IdPC ICN repair-needs ICN remotely IPC repair-needs IPC before malfunction TO repair-service **DG location DG agent DG how** RP MCR [2,2,2]

➔ 13. IdPC ICN repair-needs ICN remotely IPC repair-needs IPC before malfunction TO repair-service DG **customer-site** DG **network-of-experts DG by-geographical-proximity RP MCR** [11, 3, 18]

➔ 14. IdPC ICN repair-needs ICN remotely IPC repair-needs IPC before malfunction TO repair-service DG customer-site DG network-of-experts DG by-geographical-proximity RP **MCR how [2]**

➔ 15. IdPC ICN repair-needs ICN remotely IPC repair-needs IPC before malfunction TO repair-service DG customer-site DG network-of-experts DG by-geographical-proximity RP **MCR by-providing-initial-free-service** [20]

➔ 16 IdPC ICN repair-needs ICN client-server-mode IPC repair-needs IPC before malfunction TO repair-service DG customer-site DG network-of-experts DG by-geographical-proximity RP **MCR by-providing-initial-diagnostic-service-for-free** [21]

➔ 17. IdPC ICN repair-needs ICN client-server-mode IPC repair-needs before needs TO repair-service DG customer-site network-of-experts by-geographical-proximity RP MCR by-providing-initial-diagnostic-service-for-free [2, 2, 2]*

* The last transformation removes the extra IPC and DG's by using Rule #.2, which allows a member of A be transformed into an empty string.

explore how the goods, in this case the repair service, can be delivered to the customer, prompting the question of where and how the service can be delivered by whom (Step 12). Exploring the rewrite rules, you decide that the service would be delivered physically at the customer's site by somebody close to the location chosen among the network of experts (Step 13). The last step (Step 14) simply makes the resulting process description easier to read by removing the extra Act symbols (e.g. IdPC, DG) using Rule #.2 (c.f. Table 3).

The resulting sales process can be described as follows: It identifies customer's repair needs before any appliance (e.g. A/C, TV, refrigerator) registered malfunctions by remotely monitoring periodic data them and looking for patterns that signal repair needs. The company informs the customer of the needs. If the customer places an order for the repair contract, it delivers the service by choosing from a partner in a network of experts based on geographical proximity. It maintains the customer relation by providing diagnostic services for free in exchange for the repair service. We are not aware of any existing business embodying this model, but if there is, its existence proves all the more the potential usefulness of generating such a model systematically.

Once we get a process design thus specified, there would be a number of other design issues and details that need to be worked out. Also, to decide if this is a viable model, a great deal of additional analysis would be required. A grammatical approach is not a magic bullet that substitutes for business strategy and planning or operations management. The grammar simply provides a systematic way of combining and instantiating the cases associated with any of the components. By having them explicitly captured in the form of rewrite rules, the grammatical approach allows us to a new model to be generated beyond as a local variation of a given model.

Constraints

One of the most useful functions that a grammatical approach can serve is to help us identify the constraints that make some combinations invalid ("ungrammatical"). These constraints are linguistic, psychological, or sociological in the case of grammars for natural language. They are physical, organizational, and technological in the case of process grammars (Pentland 1995).

Physical/Logical constraints

Some process descriptions are logically or physically impossible. A product cannot be delivered before it is manufactured or a physical good cannot be delivered electronically. The grammar should reflect this type of constraint. It can do so by creating different non-terminal categories that correspond to the different properties and have separate rules for them. For example, that is why we distinguish *Physical-Goods* from *Electronic-Goods* and have rules such as *Deliver Electronic-Goods how* -> *Deliver Electronic-Goods over-the-web*.

Technological/Economical

Some possibilities are logically and physically possible, but impossible with current technology. For example, it is not currently possible or at least economically not feasible to remotely detect the potential malfunctioning of a house, but it would be someday. So the above repair-service business model might not work or might not be cost-effective for many of the home appliances now, but it may be soon. Economic constraints are treated in the same manner as technological constraints because from a business perspective, a technology is useless unless it becomes cost-effective.

Again in the grammar, we express a constraint by differentiating non-terminal categories. So we create a category called *remotely-monitorable-and-repairable-product* and update which products belong to that category. It would be desirable to separate categories whose membership would change over time from ones whose membership is fixed by the law of nature or logic. This kind of representational issue provides a variety of interesting research challenges, as discussed in more detail below.

Organizational/Institutional

Organizational policies and regulations restrict what is legal and what is not. Tele-marketing may be illegal in some countries, thus *Sell how -> Sell via-tele-marketing* should not be a rewrite rule in the grammar for that organization. Again there should be a way to mark such constraints from other types of constraints so that we are aware of what is invariant across organizations or not.

Socio-Psychological Norms

There are also softer constraints, which often originate from social norms or individual preferences. Even among countries where tele-sale is legal, it may be regarded ruder in some countries than in others. In the process grammar, this can be expressed by either having separate grammars for the different societies or by having context-sensitive rules. For example, if there is a strong norm against tele-sales in Group A but not in Group B, then the rule *Sell B how -> Sell B by-tele-sales* would ensure that tele-sales would be an option only when the location case of the Sell process is Group B, but not in Group A. Ideally the grammar should be able to express the strength of the norm, but that is beyond the scope of this paper.

CONCLUSION

Just as you cannot write Othello by randomly generating grammatical sentences, or by choosing them from a phrase book, designing a successful organization requires much more than generating a collection of grammatical processes. The approach described here does not substitute for good judgment or analysis. Yet just as grammar reveals many insights into language, we believe that a process grammar can reveal many insights into the process and contribute to effective process design.

A grammatical approach complements the existing approaches in several ways. A process grammar can express, with a relatively compact set of rewrite rules, the invariants of a process (decomposition rule) and its numerous variations (domain-specific rules). Each domain-specific rule focuses on one dimension (case) along which the process can be specialized. Thus, a grammatical representation provides us with a principled way to represent and organize the existing practices collected, for example, in best practices databases. A process grammar also allows us to go beyond the collection of existing practices and systematically explore various generation of alternative process designs. While context-sensitive rules (e.g. Inform-Potential-Customer locality -> ..) can keep the scope of rewrite rules in the proper domain, context-free rules allow our exploration to go beyond the local variations. Furthermore, because non-terminal categories represent various levels of abstraction in decomposition or in the amount of details, a process grammar allows us to weave our exploration in and out of varying levels of abstraction, not restricted to the surface level (bottom up) or to the top level (top down). Thus it provides an overall framework within which the existing tools such as the process modeling and case-based technologies can be used. Articulating a grammar also enables us to identify the different types of constraints and explore the implications of such constraints or their relaxation.

The grammatical approach presented here offers several interesting topics of research for process or organizational researchers. The most challenging is the development of a methodology for identifying and representing the constraints in the grammatical framework. We have suggested some of the ways it could be done -- e.g. identifying the appropriate non-terminal categories as in remotely-detectable-repair-needs. However, that is only a beginning step, and many questions remain. For example, are the types of constraints proposed complete? How do you represent these constraints in such a way that we know their consequences -- e.g. what become possible when they are relaxed. Although we have assumed the independence of the cases, in order to express such constraints, we would need to introduce variables that bind across the cases. That would require a more complex representation than one used here (e.g. Skvortez and Fararo 1980).

Other topics of research include the identification of cases for specific processes so that we know what kind of information to collect or how to represent existing processes. Other research would be empirical -- e.g. collecting the data that will provide a basis for the rewriting rules. It will be a hard work, but we believe that the grammatical approach, when buttressed by such work, lays a solid foundation for the study and exploration of process or business model design/redesign from both theoretical and practical perspectives.

References

Abell, P. The syntax of social life: the theory and method of comparative narratives. New York: Clarendon Press, (1987)

Alexander, C. Notes on the synthesis of form. Cambridge, Harvard University Press, (1964)

Alexander, C., Ishikawa, S. & Silverstein, M. *A Pattern Language Which Generates Multi-Service Centers*. Berkeley, Calif., Center for Environmental Structure, (1968)

Alexander, C., Ishikawa, S. & Silverstein, M. *A Pattern Language : Towns, Buildings, Construction*. New York : Oxford University Press,. (1977)

Allen, J., Hendler, J., and Tate, A. (1990) *Readings in Planning*. San Mateo, CA: Morgan Kaufmann Publishers.

Baldwin, R. and Chung, M. J., “Design Methodology Management: A Formal Approach,” *IEEE Computer*, (1995), pp. 54-63.

Barley, S. R. “Technology as an Occasion for Structuring: Evidence from the Observation of CT Scanners and the Social Order of Radiology Departments.” *Administrative Science Quarterly* 31, (1986), 78-108.

Bernstein, A., Klein, M., & Malone, T. W. "The Process Recombinator: A tool for generating new business process ideas." *Proceedings of the International Conference on Information Systems*, Charlotte, NC, December 13-15, 1999

Bond, A.H. and Gasser, L. *Readings in Distributed Artificial Intelligence* San Mateo, CA: Morgan Kaufmann (1988)

Chafe, W. L. *Meaning and Structure of Language*. Chicago, University of Chicago Press, (1970).

Chomsky, N. *Knowledge of Language: Its Nature, Origin, and Use*. New York: Praeger, (1986)

Couger, J. D., Higgins, L. F., and McIntyre, S. C. "(Un)Structured Creativity in Information Systems Organizations," *MIS Quarterly* 17, 4 (1993), 375-398.

Davenport, T. H. "Business Process Reengineering: Where It's Been, Where It's Going," in *Business Process Change: Concepts, Methods and Technologies*, V. Grover and W. J. Kettinger (eds.), Idea Publishing, Harrisburg PA, (1995), 1-13.

Davenport, T. H. *Process Innovation: Reengineering Work Through Information Technology*. Harvard Business School Press, Cambridge, MA, (1993).

Davenport, T.H. and Short, J. E. "The New Industrial Engineering: Information Technology and Business Process Redesign," *Sloan Management Review* 31, 4 (1990), 11-27.

Davenport, T.H. and Stoddard, D. B. "Reengineering: Business Change of Mythic Proportions?" *MIS Quarterly* 18,:2 (1994), 121-127.

Earl, M. J. "The New and Old of Business Process Redesign," *Journal of Strategic Information Systems* 3,1 (1994), 5-22.

Fillmore, C. "The case for case." E. Bach and R. T. Harns. (Eds) *Universals in Linguistics Theory*. Chicago, Holt, Rinehart and Winston**,** (1968), 1-90.

Fillmore, C. *Principles of Case Grammar: the Structure of Language and Meaning*. Tokyo, Sanseido Publishing Company, (1975)

Grimes, J. *The Threads of Discourse*. Mouton, The Hague, (1975)

Grover, V. and Kettinger, W. "Toward a Theory of Business Process Change", *Journal of Management Information Systems* 12, 1 (1995a), 1-21.

Grover, V. and Kettinger, W. (Eds.), *Business Process Change: Concepts, Methods and Technologies* Harrisburg, PA: IDEA Group Publishing, (1995b).

Grover, V. and Kettinger, W. "Introduction to the Special Issue: Business Process Change and Organizational Performance", *Journal of Management Information Systems* 14, 1 (1997), 9-13

Hammer, M. "Reengineering Work: Don't Automate, Obliterate," *Harvard Business Review* 68, 4 (1990), 104-112.

Hammer, M. and Champy, C. *Reengineering the Corporation: A Manifesto for Business*

Revolution. New York: Harper Business, (1993).

Harkness, W. L., Kettinger, W. J. and Segars, A. H. "Sustaining Process Improvement and Innovation in the Information Systems Function: Lessons the Bose Corporation," *MIS Quarterly* 20,3 (1996), 349-368.

Henderson, R. M. and Clark, K. B. "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms". *Administrative Science Quarterly, 35*, (1990), 9-30.

Honavar, V. and Slutzki, G. (Eds.) *Fourth International Colloquium on Grammatical Inference (ICGI-98)* Iowa State University (1998)

Jaarvenpaa, S. and Stoddard, D. B. "Business Process Redesign: Radical and Evolutionary Change" *J. Business Research* 41, 1 (1998), 15-27

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

Lee, J. and Krappe, K. *ASP Revolution: Winning Business Strategies Using Application Service Providers* New York: McGraw-Hill (2000)

Malone, T. W., Crowston, K. G., Lee, J., Pentland, B., Dellarocas, C.,Wyner, G., Quimby, J., Osborn, C. S., Bernstein, A., Herman, G., Klein, M., & O'Donnell, E. "Tools for Inventing Organizations: Toward a Handbook of Organizational Processes." *Management Science*, 45, 3 (1999), 425-443.

Miclet, L. 1996 *Proceedings of the Third International Colloquium on Grammatical Inference (ICGI'96)* Lecture Notes in AI, Springer-Verlag

Mizuno, S. *Management for Quality Improvement: The 7 New QC Tools* Cambridge, MA: Productivity Press, (1988)

Pentland, B. T. "Grammatical Models of Organizational Processes." *Organization Science*, 6, 5 (1995), 541-556

Pentland, B. T. and Reuter, H. H. "Organizational Routines as Grammars of Action" *Administrative Science Quarterly* 39, 3(1994), 484-510.

Pentland, B. T., (1999) “Building Process Theory with Narrative: From Description to Explanation” *Academy of Management Review*, 24(4): 711-724.

Salancik, G. R., & Leblebici, H. “Variety and Form in Organizing Transactions: A Generative Grammar of Organization.” Nancy DiTomaso and Samuel B. Bacharach (Eds). *Research in the Sociology of Organizations* Greenwich, CT: JAI Press, (1988)1-32

Schmidt, Linda C. and Cagan, Jonathan "GGREADA: A Graph Grammar-Based Machine Design Algorithm" *Research in Engineering Design*, 9 (1997),195-213

Simmons, R. F. "Semantic networks: their computation and use for understanding English sentences." In R. Schank and K. M. Colby (Eds) *Computer Models of Thought and Language*.. San Francisco, Freeman**:** (1973), 63-113.

Skvoretz, J. and Fararo, T. J. "Languages and grammars of action and interaction: A contribution to the formal theory of action," *Behavioral Science*, 25 (1980), 9-22.

Stoddard, D. and Jarvenpaa, S. "Business Process Reengineering: Tactics for Managing Radical Change," *Journal of Management Information Systems* 12,1 (1995), 81-108.

Weick, K. E.. *The Social Psychology of Organizing*, 2nd Ed., Reading, MA: Addison-Wesley. (1979)

Whitney, Daniel E. "Why Mechanical Design Cannot be like VLSI Design" *Research in Engineering Design* 8 (1996) 125-138.

Winograd, T. *Language as a Cognitive Process*. Reading, MA: Addison-Wesley. (1981)