A Cognitive Framework for Reengineering Knowledge-intensive Processes

Hans van Leijen, Walter R.J. Baets
NOTION, Universiteit Nyenrode
Breukelen, the Netherlands
h.vleijen@nyenrode.nl, w.baets@nyenrode.nl

Abstract

We develop a framework for improving knowledge-intensive aspects of administrative processes. We depart from the premise that changing business processes amounts to manipulating process knowledge in humans and machines, and can thus be seen as a special case of knowledge management. Such a view requires a clear view of what process knowledge is. In this paper, we develop such a view based on concepts from cognitive science. We give some implications of this model for analyzing, redesigning, and implementing knowledge-intensive processes.

1. Introduction

This paper develops a framework for improving knowledge-intensive processes and knowledge-intensive aspects of processes. Processes that involve knowledge-intensiveness should be handled with care in reengineering efforts, as trying to impose structure might hurt their effectiveness [4]. We try to explain why this is, and outline a method to guide choices in improvement efforts. We depart from the premise that changing business processes amounts to manipulating process knowledge in humans and machines, and can thus be seen as a special case of knowledge management. Such a view requires a clear view of what process knowledge is; in this paper, we develop such a view based on concepts from cognitive science.

We focus on knowledge-intensive operative processes [25]: processes in which knowledge is used to make decisions or create an output. Examples are bidding, new product development, legal support and risk assessment. However, we contend that almost any process involves knowledge-intensiveness to some extent for recovering from errors, handling exceptional cases and complaints, and for improving or adapting the process itself. Furthermore, many processes that have a clear and simple structure, involve knowledge-intensive tasks at a lower level of abstraction.

Eppler et al. give defining attributes of complex and knowledge-intensive processes [8]. Knowledge intensiveness is characterized by highly contingent outcomes or dependency on chance events, many possibilities an agent has to choose from, necessity of creativity in solving problems, quick obsolescence of knowledge, dependency of performance on agent skill, and long learning time. They distinguish an additional complexity dimension for business processes: the number of steps and the number of agents involved, interdependencies between steps, and the importance of the order in which steps are performed. In this paper, we define knowledge-intensiveness to include process complexity.

Business Process Reengineering [6], [11], is the restructuring of tasks in an organization in such a way that information technology is optimally exploited. In the first half of the nineties, it was proposed as a radical, clean-sheet approach to business transformation. Currently, its methods and tools are usually applied in an incremental fashion as just one aspect in more extensive organizational change projects.

Traditionally, BPR has been approached from two angles [6]. The top-down, "strategic" approach from the management literature tries to improve certain performance measures by offering step plans and guidelines (see [22] for an overview). The bottom-up, "technical" approach from the information systems field, on the other hand, offers modeling formalisms to represent and analyze existing and future processes.

In the literature, cohesion between these two approaches is often low: step plans offer little in the way of methods and instruments, while modeling formalisms pay little attention to their usage situation. The management literature does little to carefully define and delineate the process aspect of organizations, while the modeling formalisms literature often neglects issues of economy of modeling by insisting that "the end-to-end process" should be modeled in order to "gain insight". In practice, modeling is often considered too expensive or too slow.

0-7695-1874-5/03 $17.00 © 2002 IEEE

We believe there is a need and a means for theory integration. Recently, authors have started to develop a middle ground between top-down and bottom-up by systematically gathering and integrating redesign principles and heuristics. Nissen [18], in pursuit of a knowledge-based decision support system for BPR, describes a graph theory-based framework for analyzing processes, that is able to incorporate experiential knowledge of redesign heuristics. Kock and Murphy, in [21], give a set of redesign principles based on manipulating information and knowledge exchange. Lastly, Reijers [24] gives a comprehensive collection of heuristics gathered from the literature. These principles and heuristics give good hints regarding where to look for improvements in business processes, and we believe they can be used to substantiate step plans and direct modeling efforts. Ideally, we think a BPR method should be decision-*driven* and performance-*determined*: the process is analyzed with the aim of deciding whether a particular design principle is applicable, and to assess consequences of alternatives on performance indicators.

An early paper that partially incorporates some of these ideas is [14]. Another approach that fulfills these requirements is the Testbed method [2]. De Bruin et al. start from improvement objectives, derive relevant performance indicators and guide modeling efforts to gather the information relevant to improving just those. However, their method is tailored to processes with low knowledge-intensiveness.

Davenport et al. [4] were one of the first to consider applying the idea of reengineering to knowledge-intensive processes, and were quick to point out the differences with more traditional business process reengineering (BPR). An important critique on BPR is that knowledge is often lost in its application. In [26], it is argued that BPR often leads to a loss of vulnerable implicit knowledge embedded in "communities of practice", if those communities of functional experts are broken up into horizontal processes. This is extra dangerous because an organization usually isn't aware of the existence and importance of this knowledge for its operational vitality: its capacity to deal with exceptional situations and structural changes in its environment.

This paper is a first step towards a framework for reengineering knowledge-intensive processes that solves these problems and fulfills the requirements we described for BPR in general. We propose to replace or underpin concepts used by BPR methods using concepts from knowledge management, to do more justice to the increasing complexity of processes. Traditionally, BPR uses formalisms for the description of the structure of *actions* and, more recently, the structure of *communication* and *information* such as Petri nets, IDEF0 and DEMO [6]. We are trying to find ways of using knowledge modeling formalisms such as CommonKADS [28] and knowledge mapping [7], [29], for explaining the knowledge used to perform those actions.

Conceptual modeling of knowledge is hard because it is such an ill-defined concept. Therefore, this paper uses a cognitive theory that serves to clarify our understanding of knowledge and its relation to business processes. The solution to the knowledge problem that we propose is to conceptualize the handling of a case in a business process as solving a problem, where the organization is seen as a goal-directed planner who has to exhibit the right response in reaction to a stimulus. People use *scripts*, patterns of events stored in individual or organizational memory, to be able to solve these problems.

Scripts are a generic and powerful concept, derived from psychology [27], taken up in organization science [9], and slowly permeating thinking in knowledge management and in information systems [17]. Scripts might become a unifying concept in disciplines related to knowledge and processes, such as BPR, knowledge management, workflow management, use case analysis, and even user interface design.

We will use the knowledge-based theory of the firm [10] to connect scripts with the function of knowledge in organizations. We will then use our model to argue for the view that a process should be seen as an abstraction of coordinated competencies, and therefore, that the process design task is not about designing activities but about designing task competencies, coordination mechanisms to integrate task competencies and shared knowledge to inform the use of those coordination mechanisms. This has a number of implications for the analysis, design and implementation phases of reengineering efforts.

This paper is structured as follows. First, we describe knowledge-based and cognitive theories of the firm. Then we describe the cognitive cycle, consisting of a description of knowledge, learning and coordination. Next, we describe business processes in terms of this model. Section 3 gives implications for the analysis, design and implementation phases of a reengineering effort. We conclude with a discussion of related work, conclusions and future research.

2. The cognitive identity of the organization

Grant proposes a theory that explains the structure and behavior of the firm based on its use of knowledge [10]. Grant gives a number of features of knowledge, three of which are relevant for our purposes. Firstly, knowledge is hardly transferable between agents, unless made explicit. Secondly, knowledge elements are hard to combine because of their context specificity: to be able to cooperate, specialists must transfer part of their knowledge to each other. Thirdly, it is necessary to specialize in the acquisition of knowledge because of cognitive limitations: we cannot all know everything.

According to the knowledge-based theory of the firm, integration of task knowledge for production is the most important task of the firm. However, knowledge transfer is too expensive as a means of knowledge integration. The key to efficiency could therefore be to distribute knowledge and design *coordination mechanisms* in such a way that the necessity to transfer knowledge is minimized.

For these mechanisms to function, *shared knowledge* has to exist: knowledge about other people's knowledge, and shared beliefs. The importance of shared knowledge is that it enables employees to integrate task knowledge that is *not* shared.

Because we are interested primarily in procedural knowledge, we use Schank and Abelson's *script* [27], a psychological construct elevated to the organizational level by Gioia and Poole in [9]. A script is a contingent pattern of events in the past that an agent has extracted from a set of episodes. According to Schank and Abelson, a script is not executed directly, but is a causal model on the basis of which plans can be constructed for execution. The particular plan constructed depends on the specific values substituted in a script and the particular goal an agent has in mind.

A script is a sequence of nodes; each node representing a class of events or a class of possible actions of the agent. Such a sequence can be temporal, logical or causal. It is a logical framework for categorizing events: a schema for both understanding and behavior. To understand is to be able to interpret an event as a node in a script, because that makes the near future and behavior expected from the agent somewhat more predictable.

Nooteboom explores the notion of scripts to describe the process of knowledge creation and exploitation in organizations [20]. According to Nooteboom, a cognitive agent has structures for *perception*, *interpretation* and *evaluation*. Data enters the agent via perception, and is subsequently transformed into *information* by *interpretation* in terms of knowledge structures. *Understanding* transforms information into beliefs and causal insights. *Knowledge* is information that enables the agent to transform understanding into *action*, which delivers *performance*. This performance is the subject of improvement in BPR.

A firm has a cognitive identity: a set of scripts in which people are nodes or contribute actions to nodes. This set has a recursive structure: a script can be a node in another script. The total script memory is distributed across people, documents and computer programs. No one person in the firm knows all the scripts, although it is part of the task of management to keep an overview. This notion of cognitive identity as a set of related scripts substantiates Grant's task and shared knowledge.

Eventually, we would like to replace the *activity* concept in BPR with the script concept, but there is an inconsistency between knowledge-intensiveness and scripts: the former signifies dynamics and variability, while the latter seems to refer to a fixed sequence of steps. In the next section, we view executing a business process as goal-directed planning to restore the dynamic character of knowledge-intensiveness and we examine the relationship between business processes and knowledge-as-scripts one step further, so that we can start to derive redesign principles.

2.1. The cognitive cycle

We present a straightforward model of the knowledge and learning of agents in business processes, inspired by work in artificial intelligence. We view executing a case in a business process as solving a problem, and analyze this problem-solving behavior in a set of generic phases. To choose a cognitive model means speaking in terms of the mental processes and states of an agent: representations in the mind and problem solving behavior that manipulates those representations. To solve a problem is to plan a path of interventions in an object world to reach goals. In the service organizations that we focus on, the goal is always an action that serves or protects the interests of a stakeholder, such as a customer, an employee or the principal agent of the organization (such as the shareholders). The agent we will speak about is usually a human individual, but an information-processing machine can also be considered a cognitive agent.

Each agent in the organization has one or more competencies to transform information in such a way that it eventually leads to appropriate actions, if coordinated successfully with other agents' actions. For example, in a car insurance company, a claims handler knows how to send orders to a towing company to assist customers in recovering their damaged car, but also has to balance the interests of the customer with the interests of the company, possibly by asking an inspector whether damage inspection is necessary.

The task of an agent is to work toward solving a set of problems, where a problem is a set of features about a situation outside the organization, that offers an opportunity for value creation. An agent is fed with stimuli that indicate changes in the outside world that might require appropriate responses. Stimuli correspond to existing or new problems.

An agent remembers episodes previously experienced, where each episode is a succession of states of the world in the past, or a *story*. Such a succession is described in terms of *variables* and their values. This story describes behavior of part of the world, the agents' interventions in that part, and the effects of those interventions.

Aside from episodes, the agent's memory also stores *causal models*: contingent sequences of cause and effect situations, abstracted from sets of episodes that resemble each other. Some of these causal models are still linked to the episodes they were abstracted from; with others, these linkages have long been forgotten. An example is the causal model that guides you tying your shoe laces: the memory of someone teaching you how to tie them is not necessary anymore once a causal model has been built up.

For each problem, a plan is held. A plan is a set of future interventions the participants in the plan have committed themselves to; these interventions are contingent upon uncertain events, in other words they are decision rules with uncertain outcomes.

Creating a new plan starts with a feature-by-feature comparison of a stimulus with existing episodes and causal models in memory. If those are found, the agent will try to find or construct a causal model that predicts the near future in terms of probable paths of events. This process of recollection and construction of a causal model is the first phase in solving a problem, that we call *recognition*. Lord and Foti, in [15], call this *script selection*.

Information about the problem is usually incomplete, which leads to two problems. Firstly, in collecting episodes and causal models, it will often be difficult to decide which of multiple episodes resembles the problem most. To decide, more discriminating features of the problem will have to be searched for. The same discrimination problem can occur if during plan construction the values created by different intervention alternatives are not far enough apart to decide which one is most appropriate. The agent will therefore try to construct a more accurate picture of the problem by collecting extra information. This is the second phase in solving problems: *decomposition*. Recognition and decomposition alternate until the agent is sufficiently confident that the problem matches the model constructed. For example, the claims handler will inquire about the presence of any bodily injuries, and if so, will ask more questions to be able to choose the type of assistance likely to benefit the customer most.

The next phase is *planning*: trying to find an adequate path of interventions. For this, memory is searched for previous intervention paths associated with similar situations, which are then judged for their consequences in the present situation (plan retrieval). If a suitable plan is not found, the agent can try to find an adequate path by searching the space of all possible intervention paths in the causal model constructed and judging each on the benefits of its consequences. However, this is an expensive combinatorial operation. An example of dynamic search might be when the car has multiple damages and the claims handler has to think about the order in which to undertake repair actions. While judging proposed plans, hitherto unexpected consequences might crop up in the agent's mind, possibly necessitating a return to *recognition* and *decomposition*.

Remaining steps are *action* and *evaluate / register*. During the former, the agent will execute the plan and recursively encounter sub-problems to solve. After executing the plan, the agent might store the new scenario in its memory, and may adapt representations to account for success or failure. In the insurance case, an example is registering the case in an information system for subsequent actuarial analysis.

The cognitive cycle can be recursively refined in all its five phases; that is, executing one phase may result in executing all five on a lower level of abstraction. For example, in a process such as bidding, *recognition* might be implemented by searching a library for all bids made in the last couple of years, or by bringing people together for a meeting (which goes for *evaluation*, too). *Decomposition* is a complex process in itself in, for example, medical diagnosis. *Action* will usually lead to sub-problems requiring a solution. And *register* might entail using complex information systems or taking measurements.

Causal models are non-deterministic and thus plans are fallible. An incoming stimulus may indicate that a plan turns out to have the wrong effects. If confronted with failure during plan execution, the agent has to perform backtracking and restore or repair results of executed interventions, and then construct an alternative plan. This is likely to incur an expensive exhaustive search of the space of intervention paths in a causal model.

An agent will usually not carry out the process sequentially but will interleave phases to increase efficiency. For example, decomposition will often be skipped assuming the problem is "typical". This increases efficiency but also risk of failure. If decomposition is postponed too much, backtracking will be more frequent.

2.2. Learning

Scripts generally evolve through learning. Learning can be done by knowledge transfer in messages, or by feedback on the results of interventions. Transferring knowledge in a message is done by verbalizing causal models. Such a message has three parts: a characterization of a class of situations, a set of intervention alternatives, and a description of resultant behavior of the outside world. The sender of such a message has to determine the size of the class of situations in which this representation is valid. This in turn depends on the amount of context that sender and receiver share. If the sender is unsure about this, he might weaken the message by limiting the class of situations or by weakening the probabilities on the resultant behaviors.

For example, in a car insurance company, John might tell Mary: "drivers under age 24 are reckless; rejecting them protects our loss ratio". Upon second thought, John might limit the class of applicable situations: "*male* drivers under age 24…" Alternatively, John might believe Mary can decide for herself but weaken the message by weakening the probabilities expressed: "*some* drivers under age 24 are reckless; rejecting them *might* protect our loss ratio".

Lord & Foti [15] describe two basic learning mechanisms. *Abstraction* means decoupling a causal model from the set of scenarios it originated from by introducing abstract variables to summarize concrete ones. *Compilation*, or chunking, means neglecting plan judgment based on observed frequency of chosen plans in relation to problem features. This improves speed but increases the risk of choosing the wrong plan, and decreases flexibility of adapting plans to specific situations.

Using a causal model that has been built from a message (knowledge transfer) will be slow until it is sufficiently compiled. For example, the first time Mary encounters a prospective customer under age 24, she has to explicitly consider the effect of accepting the customer on the loss ratio, before she can decide to reject him or her. After repeated encounters however, she will simply recollect a suitable plan (rejection) without judging its consequences. This speeds up her process but makes it less flexible, since she is not anymore able to weigh in circumstances unique for the case at hand.

Learning is generally performed from the outcomes of interventions and can be done at two times: at evaluation time in the cognitive cycle, when the outcome becomes known (proactive learning), or, when a new problem presents itself and triggers recognition (lazy learning). The amount of laziness in learning behavior is a design decision that affects process adaptiveness.

2.3. Coordination as distributed cognition

Because competencies are distributed over multiple agents, intervention paths will have to be composed by integrating multiple agents' competencies. Because a customer is also an agent, any action the organization performs needs integrating at least two competencies. We call integrating competencies *knowledge coordination*. Traditionally, research into coordination focuses on tasks that are completely known beforehand (see for example [16]). By contrast, in knowledge-intensive tasks, coordination is an expensive process of dynamic search.

Seen from the individual, coordination consists of the following steps. Confronted with a problem, a knowledge *seeker* first has to realize he doesn't have the competence to solve it. Then a knowledge *provider* has to be found that might have the right competence. The two have to reach agreement about the adequacy of their mutual competencies, and have to agree on how to integrate the provider's competence into the seeker's intervention path. A crucial criterion here is how much knowledge to transfer back and forth. This view is similar to the language/action perspective of Winograd and Flores [32]. They discern conversations for action, and for clarification, possibilities, and orientation. The latter three are elaborated here.

Because someone else's knowledge is not easily knowable, it is risky to exploit this knowledge because the consequences of the combined intervention path are uncertain. This is because recognition and decomposition are now distributed: the provider does not know for sure whether the seeker's problem matches the class of problems in which the provider's representation is applicable. This has to be determined in a dialectical process in which provider and seeker decide together how recognition is to be distributed amongst them. They have two possibilities:

1. The provider describes the class of situations in which his competence is applicable, and seeker matches this with the current problem. In this case, the seeker does the recognition. For example: John tells Mary "I specialize in drivers under age 24".
2. The seeker describes the current problem and the provider matches this with the representation. In this case, the provider does the recognition. For example, Mary tells John "I have a driver here under age 24," to which John responds "I know all about that".

After having thus constructed a plan together, there are three possibilities for using the provider's competence in executing the seeker's plan: the first is that the provider executes his intervention triggered by the seeker. For example, John tells Mary to hand him over the insurance application. Second possibility is that the provider advises on a decision to be taken or an atomic action to be executed by the seeker. For example, John advises Mary to reject the customer. The third is that the provider transfers his or her knowledge to the seeker by verbalizing the associated causal model, so that he or she can apply it him- or herself from now on. For example, John tells Mary all about when to reject drivers under age 24.

We can see that knowledge coordination comprises two phases: distributed recognition/decomposition and distributed execution. In both phases, the difficulty is in deciding how much knowledge to transfer from provider to seeker and the other way around. Transferring knowledge has to be done sparingly as empirical evidence shows it is much more expensive than transferring mere information [13].

The result of each process of knowledge coordination is that a certain amount of knowledge has been transferred among participants: the seeker has a better representation of the provider's competencies, while the provider has a better representation of the kinds of situations the seeker often finds himself in. This means less knowledge will have to be transferred on future occasions. This is the way shared knowledge is built up.

Even if a business process is a completely known task, it has probably evolved from a not yet completely known task via build-up of shared knowledge. Since this process is idiosyncratic, it has likely resulted in sub-optimal procedures; this is why an account of how shared knowledge is built up is relevant to identifying process improvements.

To summarize, we have given an account of how Grant's knowledge integration is brought about by coordination. Coordination requires some transfer of knowledge, which is a form of learning. This is the way knowledge, learning and coordination are interconnected.

2.4. Business process as coordinated competencies

In preceding sections we have given an account of the structure and distribution of process knowledge in organizations. In this section, we link this model to the concept of a business process more explicitly.

Handling a case in a knowledge-intensive business process is like solving a problem: using a causal model to derive an intervention path using goal-directed planning. We previously stated that the concept of a script was too static to describe knowledge-intensive work; now, we can identify a script as a particular plan retrieved most often if a particular causal model is selected, and that corresponds with a "prototypical" instance in a class of problems.

In terms of scripts or causal models, process knowledge can be divided into three levels of abstraction. Firstly, the knowledge that you need to be able to execute subtasks in the process: Grant's task knowledge [10]. Secondly, the process itself, its salient structure, can be regarded a kind of script. At this level of abstraction, process models, use cases in requirements analysis, and workflow definitions, are all kinds of scripts. At the highest level of abstraction is the rationale behind a process, or knowledge of the way a business process relates to its environment to create value for stakeholders. These are causal models needed to be able to correct mistakes, handle exceptions, and design or adapt processes.

This begs the question of whether, in a knowledge-intensive process, the concept of a business process or a traditional activity-based process model is still meaningful. We think it is, but only when redefined as one of two things. On the one hand, in relatively simple processes, a process can be considered a script that is a compilation of the rationale: a deliberately flattened version of the deep causal knowledge of a process expert. The process model that describes such a process is an accurate description of the course of action to be taken, while leaving out details about tasks in the process. On the other hand, in genuinely knowledge-intensive processes, it can be considered a more or less crudely simplified version of a more complex reality, in which exceptional situations can be handled in ways not allowed for in the process model. In this case, the process model is a script that represents the "typical" course of action chosen on the basis of a complex causal model that allows far more elaborate courses of action. Thus viewed, a process model as a structured set of activities is a more or less crude abstraction of what really is a set of coordinated competencies.

3. Implications for reengineering knowledge-intensive processes

Similar to [17], we interpret the reengineering of knowledge-intensive processes as the manipulation of scripts to improve the way knowledge is integrated to create value. To transform the theory proposed into an instrument for process improvement, we have to elaborate a number of aspects. First, how to model and analyze processes; then, how to find improvements and assess the effects of different alternatives on performance indicators of processes, and lastly, how to implement such a design by influencing scripts in people and machines. Below, we elaborate each of these.

Processes cannot be improved boundlessly, because there is a fundamental limit to the knowledge available, the speed with which new knowledge can be created and the speed with which knowledge can be transferred between agents. Yet, it is possible to systematically search for opportunities to create new knowledge or exploit existing knowledge better.

When defining and discussing *cognitive process reengineering*, we need to account for a number of aspects. First of all, cognitive process reengineering is the *manipulation of scripts* in *cognitive agents*. This means we explicitly acknowledge that a process is implemented in scripts in the minds of people and in programmed machines, and these scripts have resulted from the hard to reverse psychological process of *compilation*. Second, reengineering serves the interests of *stakeholders*; generally, these are customers, employees, and the principal agents behind the organization. These stakeholders generally impose certain *restrictions* on the scope of the reengineering project. Third, reengineering always targets some *performance criteria.* Those criteria are generally speed, amount of effort spent, quality, flexibility, and adaptiveness. Quality might refer to the number of errors made. Flexibility is the extent to which behavior can be adapted to the specifics of a case, and adaptiveness is the ease with which behavior can be adapted in response to structural changes in the environment.

There are three fundamental *restrictions* to any process improvement effort. The first one is the available knowledge already mentioned. Second are restrictions set by the environment: the scope of an improvement project. Third are the cognitive limitations of the agents available (man, machine): memory, speed, accuracy, and flexibility.

3.1. Modeling and analysis

If a business process is governed by goal-directed planning, then process design amounts to designing the right goal structure and enabling the attainability of those goals, but not the specification of exactly how those goals are to be attained. Therefore, designing a knowledge-intensive business process amounts to modeling the set of incoming stimuli as a hierarchically decomposed set of more and more fine-grained events, a goal-subgoal structure, and modeling the set of possible intervention paths as a search space instead of as an explicit and optimized design. In order to analyze an existing process, we have to model two things: first, the knowledge used in terms of event structure, goal structure, and search space. Second, the current implementation of this knowledge: the way it is distributed as task and shared knowledge in men and machines.

The goal-subgoal structure generally consists of stakeholders' interests at different levels of abstraction, goals that serve those interests, and constraints that specify which goal states are allowed. The search space consists in the first place of a set of permitted interventions. This space can be bounded by specifying constraints on state successions, and we might be able to assess underlying complexity and knowledge-intensiveness by looking at "typical" scripts, scripts for handling errors and scripts for customization or exceptional cases. Scripts may be analyzed by assessing Eppler et al.'s complexity and knowledge-intensiveness attributes [8].

Existing formalisms might help to perform this task. CommonKADS [28] is a method for the development of knowledge-based systems and knowledge management by identifying and describing knowledge assets in organizations. DEMO (Dynamic Essential Modeling of Organizations, see for example [23]) is a method based on the Language/Action Perspective, specifically meant for modeling communication between agents and therefore suitable for gaining insight into existing coordination mechanisms. For modeling the distribution of knowledge over different agents, a number of knowledge mapping techniques exist [7], [29].

From this short list of existing formalisms, it appears that what is sorely lacking is insight into how to model adaptiveness requirements: required creativity and innovation, knowledge half-life, and learning time. Also, experiences with measuring knowledge in processes is scarce, one notable exception being [13].

3.2. Static aspects of redesign and improvement

After modeling and analysis, improvement opportunities have to be looked for. Business process improvement for knowledge-intensive processes is based on the common premise that work can by improved by scrutinizing existing practice and exploiting information technology. What is different is that pressing for optimization too hard can have adverse effects [4]. In our framework, this can be explained by observing that in these domains, the search space of intervention paths is so vast that trying to structure this space too much would cut many potentially valuable courses of action.

In the above, we have looked at knowledge-as-scripts in two ways: the distribution of those scripts among agents as shared and task knowledge, and the internal structure of scripts as the cognitive cycle. We can now define three fundamental ways to improve processes: by making new knowledge available or distributing existing knowledge better (meta-level), by analyzing and enhancing existing knowledge (object-level), and by using information storage (technology or paper) to compensate for cognitive limitations in human agents.

New knowledge can be imported or imitated from competitors, can be created by experimenting, or by combining existing knowledge from multiple agents: collecting *best practices*. In redistributing knowledge, two considerations play a role. On the one hand, it is advantageous to attune the characteristics of a competence to the cognitive limitations of the agent possessing the competence: if little flexibility is required, for example, a task can be handed off to a machine. On the other hand, fine-grained distribution of knowledge, or specialization, increases coordination costs. Redistribution of knowledge to less flexible agents is hard, as it requires explicitation of hitherto tacit knowledge.

Information systems development is in this view seen as an extreme form of knowledge explicitation and is therefore very difficult. The most extreme form of knowledge redistribution is *automation*: the complete handing over of a task to a machine. In case of task knowledge, knowledge-based systems are an example, while automating shared knowledge can be done by employing workflow management systems.

This last case is an example of changing the coordination search space: changing the shared knowledge available to people for performing knowledge coordination. A workflow management system structures and therefore restricts the number of possible collaboration opportunities between agents, but also speeds up the search for combined intervention paths. "Yellow pages" or "expert directories" do the exact opposite: they enlarge the number of collaboration opportunities and trade off speed for effectiveness, flexibility and quality, because the agent can now coordinate much more knowledge.

The second fundamental form of knowledge-based process improvement is enhancing the structure of existing knowledge. The cognitive cycle lends us some insights with which to analyze scripts and alter some balances in them. We give three examples.

We viewed the execution of a business process as the classification of a new problem situation into known classes of situations and selecting an appropriate response by recognition and decomposition. We can now ask ourselves if our number of responses is appropriate with respect to the variety of incoming situations. Having more possible responses costs more effort in recognition and decomposition, but might save effort or increase value down the line, and in general increases flexibility. Deepening or flattening the stimulus-response function of a business process is the first balance to manipulate.

We have said that an agent will in general not execute the cognitive cycle in sequential order: assumptions will be made and actions will be taken before all information is available. Increasing or decreasing interleaving is our second balance. Increasing interleaving will improve speed at the expense of quality, as atypical situations will be misjudged and more often lead to failure and backtracking.

Lastly, the proactivity of learning can be adjusted to the adaptiveness requirements of the process. We have defined learning as the adaptation of causal models based on success or failure of problem solving in the *register/evaluate* phase. We can choose to perform this stage after handling a problem, or just before handling the next. In a highly knowledge-intensive task such as new product development, learning is often done up front by looking for similar projects in the past and finding success factors, as well as afterwards by having an evaluation meeting. Proactive learning should not be chosen when good information about success or failure becomes available only after long periods of time, such as in insurance or in some branches of medicine.

The third fundamental form of knowledge-based process improvement is using information storage to compensate for cognitive limitations. We can use information technology or paper means to store either *facts* or *scripts*. Examples of the former are databases, document repositories, and, more specifically, data warehouses built for statistical analysis. Examples of the latter are workflow systems and process charts. Both amount to enhancing memory. Yet, whereas the former is meant to retain important features of *episodes*, the latter is meant to retain *causal models* or their compilation into *scripts*. Storing facts increases adaptiveness since it offers opportunities for learning from the past, yet it decreases speed since the storage itself and its analysis takes time. Storing scripts decreases flexibility because people are not anymore allowed to depart from standard practice in exceptional circumstances. It might also have the effect of decreasing adaptiveness since the prescribed course of action becomes ingrained. Yet, on the other hand, making knowledge explicit often triggers reflection on its validity, which might in turn lead to a an attitude more favorable to change.

3.3. Dynamic aspects of the design process and implementation

The phases of analysis, redesign and migration require conversion of knowledge from tacit to explicit, and subsequent transfer of knowledge between individuals. There is common agreement that knowledge explicitation and transfer are very hard. This is psychologically plausible since scripts are meant for improving information processing efficiency, while sacrificing adaptiveness [15]. In section 2.1, we have described how, through compilation, efficiency is increased at the expense of adaptiveness, because "higher goals" are forgotten. This explains why changing processes that are relatively simple but have long been operative is much harder than expected.

Grant's distinction between task and shared knowledge raises interesting questions. As mentioned above, business processes usually evolve through build-up of shared knowledge. Reengineering largely amounts to changing and facilitating this shared knowledge, which requires disentangling of task and shared knowledge. An interesting research question is the extent to which task and shared knowledge really are entangled in practice, that is, do scripts cross boundaries between task and shared? If so, this would seriously complicate process change.

The process of modeling, analyzing, redesigning and transforming a business process can be seen as a case of Nonaka and Takeuchi's knowledge cycle [19]: it is a process of externalizing, combining and internalizing knowledge via explicitation and socialization, where externalization requires codification of tacit knowledge. Various other authors have addressed the dynamics of such knowledge conversion. Nooteboom speculates about a logic of change [20], violation of which would lead to failure. This logic consists of the phases experimentation, consolidation, generalization, differentiation, reciprocation, each of which can be defined by describing manipulation of scripts in terms of node substitution, reordering, combining scripts etc. Jorna and van Heusden use a semiotic framework to describe changes in mental representations [12], and describe evolution of organizational knowledge from tacit via codified to abstract. According to them, organizational change requires conversion of knowledge along these dimensions. These three theories describe knowledge change from an evolutionary, incremental perspective. It is not yet clear how these ideas can be used in the case of deliberate process redesign and planned change.

4. Discussion and related work

In the introduction, we mentioned three collections of BPR principles and heuristics [18], [21], [24]. There seems to be some overlap and some complementarity between the works cited and the work presented here, but our framework is as yet much less developed. As a preliminary conclusion, we can say that our work has a focus on effectiveness rather than efficiency, is more cautious with change, and analyzes processes at a higher, semantically richer abstraction level.

0-7695-1874-5/03 $17.00 © 2002 IEEE

Lee, in [14], describes a systematic method for process redesign called Goal-based Process Analysis (GPA). Distinctively, GPA is one of the few methods that support the identification of design alternatives. It departs from the concept of a goal-subgoal hierarchy to represent the intentional structure of a business process. This strongly resembles our concept of executing a business process as goal-directed planning. However, finding alternatives in GPA relies on a pre-supplied typology of goals, and not on manipulating dimensions in a theory of the structure of the knowledge behind business action.

The same focus on coordination mechanisms is put forward by Crowston and Malone in [3], [16]. Their approach to process reengineering is based on choosing appropriate coordination mechanisms to manage dependencies between tasks. This work is an important building block of ours, yet seems to need expansion for dealing with situations where tasks are not completely known beforehand.

Vidgen, in [31], proposes a synthesis of the Viable Systems Model with BPR. The VSM, developed by Stafford Beer, is an application of cybernetics to organizational design. While this work is not concerned with knowledge per se, the central notion of *variety* in cybernetics seems to bear strong resemblance to complexity and especially knowledge-intensiveness, since it has to do with adaptiveness to change. The allocation of variety might coincide with the allocation of knowledge in organizations.

Our approach resembles Manheim's *Cognitive Informatics* [17], defined as "the use of knowledge about how people think and act to design information technology (IT) support that enhances the way people think and act." Where Manheim uses the term *patterns*, we use the term *scripts* to describe the contents of human experiential knowledge.

In the knowledge management field a lot has been written about the relationship between knowledge and processes (see for example [1], [25], [30]). In that literature knowledge is usually described as a resource that is used in processes, instead of as a set of competencies from which processes are dynamically woven. Another difference is that knowledge is usually categorized in different ways (for example in *tacit* and *explicit*) but is otherwise left a black box. In our cognitive cycle, the structure of knowledge itself is analyzed. A third difference is that knowledge maintenance is often assigned to a separate process (e.g., see [1]), while in our framework it is an integral part of executing a process.

5. Conclusions and future research

We have taken a cognitive perspective on business processes, in order to be able to design the proper coordination, maintenance and use of operational knowledge in service organizations. This has resulted in a framework that interprets organizations as goal-directed planners and analyzes processes in terms of the five phases recognition, decomposition, planning, action, and evaluation. Knowledge is seen as a causal model that stores experiences regarding regular patterns of behavior of the outside world and the effects of actions therein. The psychologically well-motivated script construct connects knowledge management to business process management.

Our model is complementary to more traditional process reengineering methods and principles, but also partly overlapping them. The crucial difference between our model and existing approaches to analyzing knowledge and business processes is that it views a knowledge-intensive process as a result of coordinated competencies rather than the process itself specifying the coordination and using knowledge as a resource. Related approaches to modeling knowledge and processes described above seem strongly complementary, but do not share this view on the relationship between knowledge and business processes. An important area for future research is comparing these analytical frameworks with ours.

We believe that taking the cognitive view provides opportunities to integrate results of research into organizational learning, knowledge management, and process management. It would be interesting to do a detailed comparison and theoretical integration of principles for reengineering processes along a scale of knowledge-intensiveness. This paper is a first step in that direction.

6. Acknowledgements

This paper has benefited from comments made by a reviewer. This research is supported by a research grant from Achmea.

7. References

[1] Allweyer, T. (1997), "A Framework for Re-designing and Managing Knowledge Processes", IDS Prof. Scheer GmBH, whitepaper.

[2] de Bruin, B., Verschut, A., and Wierstra, E. (2000), "Systematic analysis of business processes", *Journal of Knowledge and Process Management*, Vol. 7, No. 2, 87—96.

[3] Crowston, K. (1997), "A Coordination Theory Approach to Organizational Process Design", *Organization Science*, Vol. 8, No. 2, pages 157—175.

[4] Davenport, T.H., Jarvenpaa, S.L., and Beers, M.C. (1996), "Improving knowledge work processes", *Sloan Management Review*, Vol. 37, Summer, 53—645.

[5] Decker, S., Erdmann, M., Studer, R. (1996), "A Unifying View on Business Process Modelling and Knowledge Engineering", *Proceedings of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada.

[6] Dietz, J.L.G., Mulder, H.B.F. (1996), "Integrating the strategic and technical approach to business process engineering", *Proceedings of the International Symposium on Business Process Modelling*, Springer-Verlag, 1996.

[7] Eppler, M.J. (2001), "Making Knowledge Visible Through Intranet Knowledge Maps: Concepts, Elements, Cases", in: *Proceedings of the 34th Hawaii International Conference on System Sciences – 2001*.

[8] Eppler, M.J., Seifried, P.M. and Röpnack, A. (1999), "Improving Knowledge Intensive Processes through an Enterprise Knowledge Medium", in: Prasad, J. (ed.): *Proceedings of the 1999 ACM Conference on Managing Organizational Knowledge for Strategic Advantage*, New Orleans (USA), April 8th-10th, 1999.

[9] Gioia, D.A. and P.P. Poole (1984), "Scripts in organizational behavior", Academy of Management Review, Vol. 9, No. 3, pp. 449—459.

[10] Grant, R.M. (1996), "Toward a knowledge-based theory of the firm", *Strategic Management Journal*, Vol. 17 (Winter Special Issue), 109–122.

[11] Hammer, M. and Champy, J. (1995). "Reengineering the Corporation: A Manifesto for Business Revolution". Nicholas Breadley Publishing, London.

[12] Jorna, R. and van Heusden, B. (2000), "*Cognitive Dynamics: A Framework to Handle Types of Knowledge*", In: J. Schreinemakers (Eds.), Knowledge Management: the 7th ISMICK Conference, Rotterdam University Press, pp. 128—145.

[13] Kock, N. and McQueen, R.J. (1998), "Knowledge and Information Communication in Organizations: An Analysis of Core, Support and Improvement Processes", *Knowledge and Process Management*, Vol. 5, No.1, 29—40.

[14] Lee, J. (1993), "*Goal-Based Process Analysis: A Method for Systematic Process Redesign*", In: Proceedings of the Conference on Organizational Computing Syustems, Milpitas, CA.

[15] Lord, R.G., and Foti, R.J. (1986), "Schema Theories, Information Processing, and Organizational Behavior", in: *The Thinking Organization*, Sims, H.P., Jr., and Gioia, D.A. (eds.), San Francisco: Jossey-Bass.

[16] Malone, T.W., and Crowston, K (1994), "The Interdisciplinary Study of Coordination", *ACM Computing Surveys*, Vol. 26, No. 1, pages 87—119.

[17] Manheim, M.L. (1996), "Beyond Groupware and Workflow – the Theory of Cognitive Informatics and its Implications for a People-based Enterprise Information Architecture", in: *Proceedings of PRIISM 1996 Conference*, Colorado Springs, Colorado.

[18] Nissen, M.E. (1998), "*Redesigning reengineering through measurement-driven inference*", MIS Quarterly, Vol. 22, No. 4; pp. 509—534.

[19] Nonaka, I. and Takeuchi, H. (1995): *The Knowledge Creating Company*. Oxford University Press.

[20] Nooteboom, B. (1996), "Towards a Cognitive Theory of the Firm: Issues and a Logic of Change", EIASM conference on organisational cognition, Stockholm.

[21] Kock, N. and Murphy, F. (2001), "Redesigning acquisition processes: a new methodology based on the flow of knowledge and information", Defense Acquisition University Press, Fort Belvoir, VA.

[22] Peppard, J., and Rowland, P. (1995), "The Essence of Business Process Re-engineering", Prentice Hall, New York.

[23] van Reijswoud, V.E., Mulder, H.B.F. and Dietz, J.L.G. (1997), "Speech Act Based Business Process and Information Modelling with DEMO", In: *The Proceedings of the CAiSE'97/IFIP8.1 International Workshop on Evaluation of of Modelling Methods in Systems Analysis and Design*, Barcelona.

[24] Reijers, H.A. (2002), "Design and Control of Workflow Processes: Business Process Management for the Service Industry", Ph.D. dissertation, Technische Universiteit Eindhoven, 2002.

[25] Remus, U. and Lehner, F. (2000), "The Role of Process-oriented Enterprise Modeling in Designing Process-oriented Knowledge Management Systems", *Proc. of the AAAI symposium on bringing knowledge to business processes*, 2000.

[26] Scarbrough, H. (1998), "BPR and the Knowledge-Based view of the firm", *Knowledge and Process Management*, Vol. 5, No. 3, 192–200.

[27] Schank, R. and Abelson, R. (1977), "Scripts, Plans, Goals and Understanding", Hillsdale: Lawrence Erlbaum Associates.

[28] Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W. and Wielinga, B. (2000), "Knowledge Engineering and Management – the CommonKADS Methodology", MIT Press, 2000.

[29] Seemann, P., Cohen, D. (1997), "The geography of knowledge: from knowledge maps to the knowledge atlas", *Journal of Knowledge and Process Management*, Vol. 4, No. 4, 278—285.

[30] Staab, S., Schnurr, H.P., Studer, R. and Sure, Y. (2001), "Knowledge Processes and Ontologies", *IEEE Information Systems*, February / January 2001 (Special issue on knowledge management).

[31] Vidgen, R. (1998), "Cybernetics and Business Processes: Using the Viable System Model to Develop an Enterprise Process Architecture", Journal of Knowledge and Process Management, Vol. 5, No. 2, pp. 118—131.

[32] Winograd, T. and Flores, F. (1986), "Understanding Computers and Cognition: a New Foundation for Design", Ablex, Norwood NJ.