A Goal-Based Approach for Learning in Business Processes

Pnina Soffer, Johny Ghattas and Mor Peleg

Abstract Organizations constantly strive to improve their business performance; hence they make business process redesign efforts. So far, redesign has mainly been a human task, which relies on human reasoning and creativity, although various analysis tools can support it by identifying improvement opportunities. This chapter proposes an automated approach for learning from accumulated experience and improving business processes over time. The approach ties together three aspects of business processes: goals, context, and actual paths. It proposes a learning cycle, including a learning phase, where the relevant context is identified and used for making improvements in the process model, and a runtime application phase, where the improved process model is applied at runtime and actual results are stored for the next learning cycle. According to our approach, a goal-oriented process model is essential for learning to improve process outcomes.

1 Introduction

Organizations constantly strive to improve their business performance. This has been reflected in efforts made in the area of business process redesign since the early 1990s [5]. Typically, business process redesign initiatives can be characterized on a continuum from radical reengineering [10] to incremental continuous improvements. Various analysis methods can be used in order to identify weaknesses and improvement opportunities in existing business processes. However, the actual task of process redesign still relies exclusively on human reasoning and creativity.

Improving business performance over time is also associated with the concept of organizational learning, defined as the capacity within an organization to improve its performance based on experience [12]. One of the main ideas of organizational learning is that while individuals can learn from past experience, this knowledge has to become shared and applied across the organization to facilitate constant improvement. The knowledge an individual has can be manifested in decision criteria used for selecting a process path at a specific situation or in deviations from a predefined process model in order to solve specific problems. This knowledge is gained through mistakes as well as successful process executions.

The knowledge can become shared by others to support organizational learning by embedding it in a process model which evolves over time.

Organizational learning and business process redesign initiatives require in-depth understanding of the current practices. In particular, it should be possible to identify which actions have a positive or a negative effect on business measures in given situations. Relying only on a static predefined process model in order to identify opportunities for improving a process may be limiting or even impossible, since the actual way processes are performed (including ad hoc decisions) is usually not reflected in such models. The actual way in which a business process is performed can be studied using process mining techniques [24]. Process mining analyzes an event log of the information system that supports the process, and produces a model of the process as it is actually performed. Process mining serves various purposes, such as getting a clear and reliable model of the as-is process [24], performing delta analysis, in which the actual process is compared to the predefined process model [2], and analyzing the process with respect to specific performance measures, such as execution time [1]. Process mining can provide an understanding of the as-is process, including specific paths that reflect ad-hoc decisions made in exceptional situations. However, the main emphasis of existing process mining approaches has been capturing the control-flow of the process, namely, the sequence in which activities are executed. Hence, while process mining reflects the common as well as the rarely taken process paths, the situations in which path selection decisions have been made and the extent to which the corresponding executions were successful are not systematically addressed.

This chapter proposes an approach for learning and gradually improving business processes. The approach ties together three elements that comprise the experience gained through ongoing process executions: what actions have been performed, in what situations, and what has been achieved by the process in business terms. The actions that have been performed are the actual process *paths* taken; the situations in which they were performed are the *context* of the process; what has been achieved can be assessed with respect to defined process *goals*.

2 How Goals and Context Facilitate Learning

2.1 The Role of Process Goals

Learning from experience means understanding what mistakes were made (leading to failure) to avoid repeating them, and what was done in successful process executions. Success (or failure) can only be assessed when goals are known and specified.

In this chapter we adopt the notions of business process goals as defined in the Generic Process Model (GPM) framework [19]. GPM is a state-based and goal-oriented view of a process, which relates to two types of process goals: hard goals (or simply goals) and soft goals. Below we discuss these two types of goals and their possible use for process learning.

The hard goal of a process is defined by GPM as a set of stable states at which the process terminates. The goal set is specified by a predicate over values of the state variables of the domain in which the process operates. Since a process is executed in order to bring about some state of affairs in the domain, the predicate expresses the conditions under which this state of affairs is achieved so the process can terminate. The goal is a set of states (rather than a single state), since there might be different specific states that meet the termination condition. For example, a sales process reaches its goal once the order is fulfilled and paid for. This may include different states (e.g., the goods were shipped to the customer, the customer has taken the goods himself, payment was made in advance or upon delivery, etc.).

Considering learning, a process instance (namely, a specific execution of the process) may end up in a state which is in the goal set or in a stable state which is not in the goal set (an exception). In the sales process example, it might be that the customer received the goods, paid with an invalid credit card, and lost contact, so he cannot be located any more. In this case, the state of the process domain is stable, namely, it cannot be changed by actions of the organization, but it is not in the goal set since payment was not received. Learning seeks to avoid exceptional situations or to minimize their occurrence over time.

As explained, the (hard) goal of a process is a set of states, all satisfying the condition under which the process can terminate having achieved what it was intended to achieve. These states might be different from each other in business terms. For example, in the above mentioned sales process it might be considered more desirable to supply goods in two days than in two weeks (although both lead to goal states). Soft goals represent business objectives which differentiate states in the goal set according to how desirable they are. In other words, soft goals define a desirability order relation among states in the goal set [19].

The term "soft goals" is borrowed from requirements engineering, where it relates to desired properties whose satisfaction is not on a binary scale. Similarly, considering business processes, soft goals correspond to performance indicators whose increased values are sought, but they can only be considered successful in comparison to others rather than absolutely. It is possible to define thresholds to performance indicator values, so values above the threshold are considered "good" (e.g., delivery time shorter than one week) as opposed to values below the threshold. Yet, different values of soft goal related performance indicators denote different levels of success even if all values are above the threshold.

Learning in a business process should seek to achieve higher levels of soft goal related indicators over time.

It should be noted that specific soft goals (e.g., minimizing execution time) and their relationships to actual paths have been addressed to some extent by process mining approaches [1, 6]. Here we address soft goals at a generic level, without limiting ourselves to specific ones. This raises two main challenges. First, different soft goals may exist and may even conflict with each other (e.g., quality and cost). Process changes may positively affect one soft goal while negatively affecting the other. Second, soft goals may be affected by more than one process. For example, the quality of a product may be affected by the production process and by the purchasing of raw materials. It follows that considerable attention should be devoted to the precise specification of soft goals with respect to a specific process for learning to be effective.

As explained, learning should assess the level of success of each process execution (process instance) with respect to the defined goals of the process (both hard and soft goals). We refer to the combination of hard and soft goal achievement by a process instance as the *outcomes* of the instance.

2.2 The Role of Context

The success of a process instance can be affected not only by the actual path performed, but also by environmental conditions, not controlled by the process, which we term the process context. Specifically, the context of a process includes the initial state at which the process is triggered (which may hold specific case characteristics, such as customer properties in a sales process) and events in the environment that may occur during its execution. The initial state is specified by values of state variables known when the process is initiated; events in the environment are external to the process domain but affect its state.

Process instances of different contexts may need to be addressed differently (i.e., take different paths) in order to achieve desired outcomes. Alternatively, we may say that if exactly the same path is applied to process instances of different contexts, it might lead to different outcomes. Considering a sales process, a regular customer may place an order and pay once the goods are supplied, while an unknown customer would be required to pay in advance to reduce the risk.

Furthermore, threshold levels of soft goals for determining whether an outcome is “desired” or not may also depend on context. For example, a desired outcome of a broken leg treatment process for an old person would be to be able to walk freely again, while for a young person it would only be considered successful if he never suffers pain again.

It follows that to learn effectively, process instances should be classified according to their context, so that learning could take context into account. However, initial information and external events may relate to a variety of factors, and it is not necessary that all factors indeed affect the outcome of the process. Hence, the challenge faced is to identify the relevant factors that should be taken into account by learning. While some factors may be well known to domain experts and even incorporated into the process model as decision criteria (e.g., regular vs. first-time customer), others may be guessed intuitively by some workers, and some even unknown in advance.

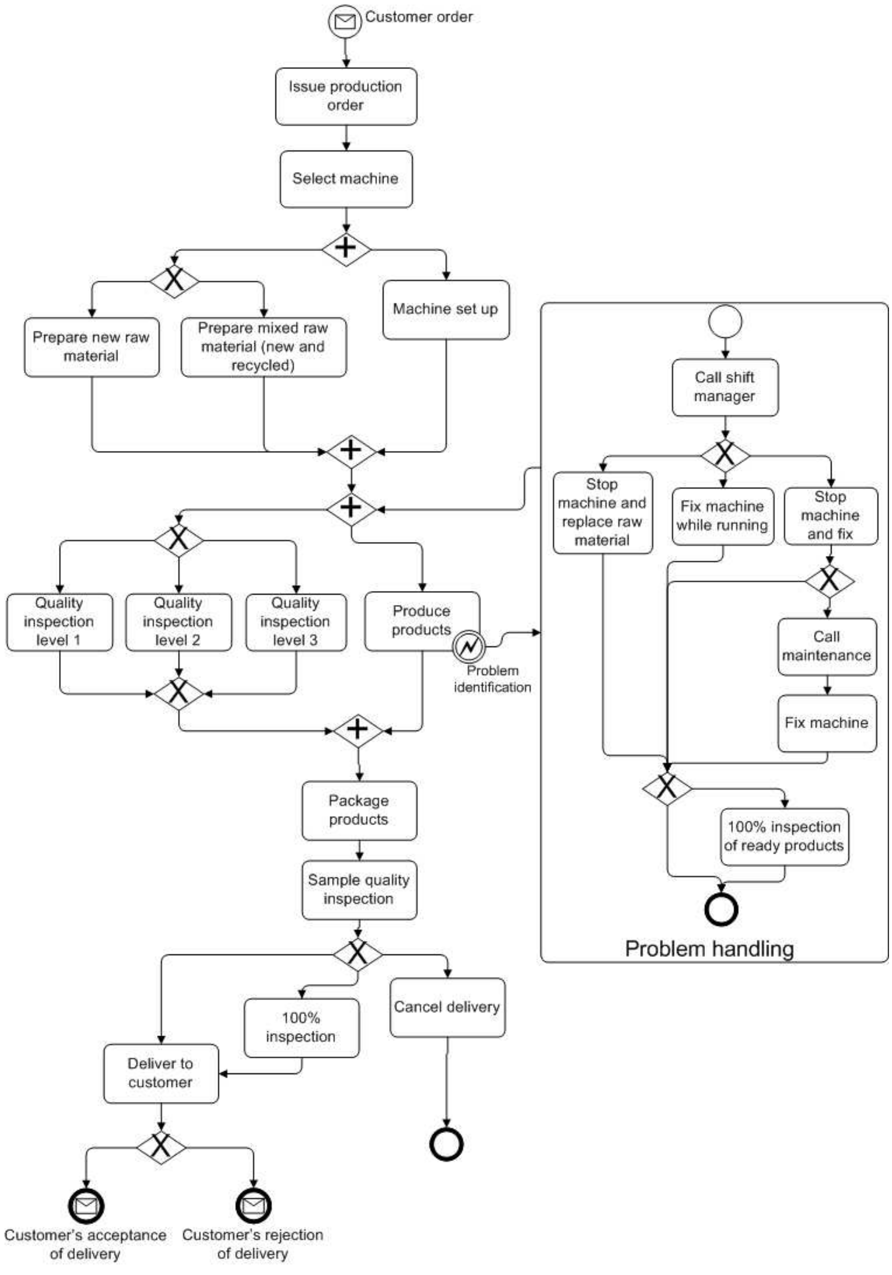

Fig. 1. Plastic bottle manufacturing process

Our learning approach seeks to identify the relevant contextual factors and group process instances into context groups, such that for process instances of a specific context group, similar process paths would imply similar outcomes.

3 Introducing the Running Example

As a running example, demonstrating the concepts introduced in the chapter, we address a production process in a plastic bottle manufacturing firm, illustrated in the BPMN model depicted in Fig. 1. The (hard) goal of the process is to reach a state where customers' acceptance of delivery is achieved. Other states in which the process might terminate (exception termination states) are states where delivery is cancelled due to quality problems and states where the customer rejects delivery (also due to quality problems).

Soft goals defined by the organization include increasing the percentage of deliveries that meet their due dates, increasing machine utilization, reducing waste of raw materials, increasing the quality of the manufactured products, and reducing the overall production costs. These different soft goals could be prioritized and weighted to form one composite soft goal. An alternative approach would be to analyze the dependencies among soft goals and identify a dominant soft goal to be addressed first. Table 1 presents the main causes for poor achievements of the defined soft goals.

Table 1. Soft goals – reasons for poor achievement

Soft goal	Main reasons for poor achievement
Meeting delivery due dates	Quality problems, machine failure, unskilled workers
Increasing machine utilization	Set up times, quality problems, machine failure
Reducing waste of raw materials	Quality problems, inappropriate machine setup
Increasing product quality	Inappropriate machine setup, poor quality of raw material, poor machine condition, inappropriate quality inspection
Reducing production costs	Raw material cost, raw material waste, labor cost

As seen in Table 1, the leading reasons for poor business results are quality problems, machine failure and poor technical condition, and the set up operations. Machine maintenance is not in the scope of this process (rather, it is part of its context), and the set up operation is part of the process path. Based on this analysis, we decided to focus on the soft goal of increasing the product quality, which will affect all the other soft goals (including the costs, through reducing material waste).

The contextual variables of the process include the initial case properties and external (uncontrolled) events during the process. Initial case properties include properties of the manufactured product and the customer, intended market of the product (food, medical supplies, chemicals, cosmetics), the main raw material (polyethylene at different density levels, polypropylene), the supplier of the raw material (three possible ones), and the supplier of the pigments (two possible ones), time since last maintenance operation of the machine, and weather (hot or dusty days may affect the machines). There might also be specific requirements made by the customer, such as requirements for the bottle to be resistant to high temperature (in case the customer uses it for storing hot liquids) or to strong chemical solutions. Events that occur during the process are mainly machine failures and quality problems and it is often impossible to tell one from the other.

It should also be noted that contextual variables may affect soft goal thresholds, e.g., higher quality is required if the intended market of the product is the medical supplies.

4 The Learning Approach

The proposed learning approach includes a learning life-cycle, described in Fig. 2. The learning cycle can be initiated when a process has already been performed for a period of time, so some experience has already been accumulated. This experience is stored in an experience base, including data of past process instances: their actual path, their achieved outcome, and their context information. Note that context information includes all the environmental variables, as we cannot tell in advance which ones are relevant.

The life cycle includes two main phases: a learning phase and the application of its results in runtime, which, in turn, produces more experience to be stored in the experience base for the next cycle.

The learning phase includes a step of initial context identification, which yields a definition of context groups. These groups serve as a basis for the generation of improvement recommendations to the process model. Note that these improvements are not automatically deployed. Rather, the analysis yields improvement recommendations which should first be reviewed by humans and only then used for updating the process model.

The runtime phase is an ongoing phase of process execution. Every new process instance is classified into a context group and follows the path recommended accordingly. Its context, path, and outcome data are then stored in the experience base. There might be instances which cannot be classified into an existing context group; they are executed and their data is also stored in the experience base. In some cases, specifically when facing unexpected external events, the process operators may decide to deviate from the process model and take a path which has not been taken before. These are also stored in the experience base. Periodically, when a considerable number of new experiences have been added to the experience base, learning can be applied again, triggering a new cycle. In what follows we provide details about the phases of the learning cycle.

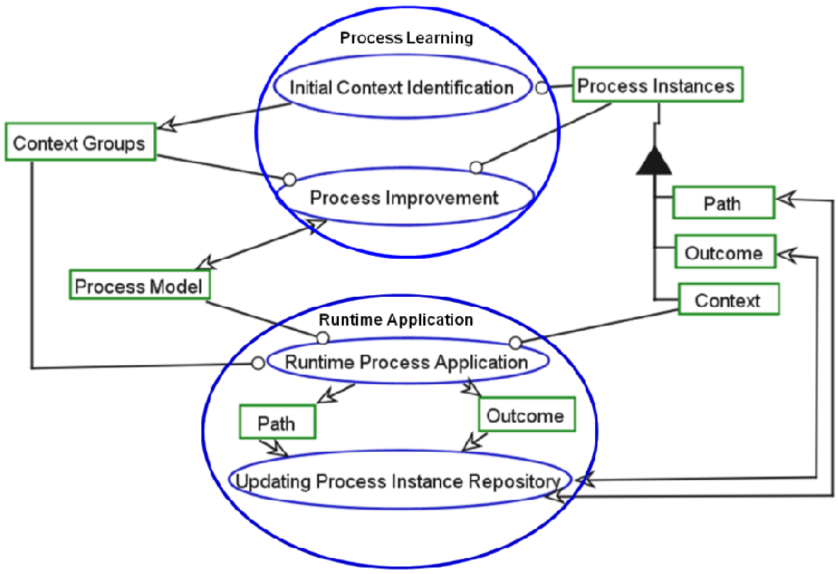

Fig. 2. The proposed learning cycle

4.1 Context Identification

As explained above, the challenge in identifying context is the huge amount of contextual information that may be available. We seek for classification criteria of process instances that would be effective in determining the best process path at a given situation. This classification should also be meaningful in business terms, so each group of instances can be characterized based on its contextual properties.

Recall, the data of the process instances stored in the experience base includes their actual path, their outcome (or termination state), and their contextual information. The path and the termination state of a process instance constitute its behavior. In a perfect world, process instances that have similar contexts would follow similar paths to lead to a given termination state. However, our knowledge of the process (relevant) context is partial. Under partial knowledge, we may not be aware of contextual variables whose different values may differently affect the process behavior, and can be considered “different contexts”. Lacking such knowledge, we may group process instances that partially share the same context but exhibit different behaviors. This would not be an effective strategy for learning the best paths that for a given context would achieve desirable outcomes. Hence, with the knowledge that exists at this phase, process instances can be grouped considering two types of similarities:

(1) Behavioral similarity.
(2) Contextual property-based similarity.

Clearly, these two groupings are expected to be different, since not all contextual properties necessarily affect process behavior, and some properties may have a similar effect. Our interest is to identify a third type of grouping, *context groups definition*, namely, groups of instances whose contextual property-based similarity can predict some behavioral similarity.

Behavioral similarity of process instances can be assessed using some path and state similarity measures. Consequently, process instances can be grouped into clusters of behaviorally-similar process instances, sharing similar paths and similar termination states (outcomes). Each process instance in the experience base would belong to one behavioral similarity cluster.

Contextual property-based similarity of process instances is possible when at least one contextual property of these instances has similar values. The possible number of contextual property-based similarity groupings is combinatorial in the number of contextual properties. Not all these groupings are meaningful in terms of behavior (e.g., grouping process instances based on the color of the customer's eyes would probably be ineffective for predicting behavior of process instances).

Based on these two types of similarity, we define a *context group* as a group of process instances, which are contextual property-based similar, and for which taking similar paths implies achieving similar outcomes.

Note that this definition relates to a situation where the behavior of process instances is fully consistent with respect to their context, namely, there are no unpredicted behaviors or noisy data. Clearly, this is not the situation in real life, where there might be "hidden" variables which cannot be tracked (e.g., distractions of the machine operator) that affect the outcomes of the process. Hence, we cannot assume full predictability of the outcomes given a context group and a process path. However, we may assume that contextual properties have a certain effect and can explain at least part of the variance in the outcomes achieved. Hence, for practical purposes we can formulate the following postulate:

Postulate 1: Consider two groups of process instances, PI_1 and PI_2, so each group includes contextual property-based similar process instances. Now consider $C_1 \subset PI_1$ and $C_2 \subset PI_2$, so the paths taken by instances in C_1 and C_2 are all similar. If statistical tests show that the termination states of C_1 and C_2 are not of the same population, then PI_1 and PI_2 are in different context groups.

Postulate 1 gives us a criterion for excluding two sets of instances from being in the same context group. It can be applied to groups of instances that follow similar paths. However, we may have groups that follow different paths. In that case, we assume the choice of path reflects some implicit domain knowledge used by the process operators. This is reflected in the following postulate.

Postulate 2: Groups of contextually similar process instances form one context group if the distribution of their behavioral similarity categories is similar (not significantly different).

The two postulates can be helpful when some grouping based on contextual properties is available. However, as discussed above, the number of such groupings is combinatorial in the number of known contextual properties. To overcome this difficulty, we employ a learning algorithm, which grows a decision tree whose independent variable is the contextual properties while the dependent variable is the behavioral similarity category of process instances. The algorithm is applied through the following procedure [7]:

Step 1*:* Use existing domain knowledge for an initial classification of process instances based on contextual properties that are known to affect process behavior.

For each partition, separately apply the following three steps.

Step 2*:* Establish the behavioral similarity of the process instances.

(a) Path similarity categories are formed using a clustering algorithm over path data of the instances. The number of path similar clusters generated is selected according to goodness of fit criteria, such as Akaike Information Criteria (AIC). The clustering algorithm can be applied several times, achieving a series of clustering results with an increasing number of clusters for each clustering set. Finally, the best cluster set is selected as the one that attains the first minima of the ratio of AIC changes.
(b) Categorize termination states to a small number of categories based on a set of predefined rules. The aim is to achieve a coarse grained categorization with a clear distinction between categories.
(c) Combine path similarity categories with termination state categories into behavioral similarity categories.

Step 3: Establish the contextual properties that affect behavior. This is accomplished by training a decision tree algorithm, using the context data as inputs and the behavioral categories as dependent variable (their label). The objective of using the decision tree is to discover the contextual semantics behind each behavioral category. We use a modified Chi-squared Automatic Interaction Detection (CHAID) growing decision tree algorithm to construct the decision tree that represents the context groups and their relationships. CHAID tries to split the context data of the process instances into nodes that contain instances whose dependent variable values (namely, behavioral similarity category) are the same. Each path from the source node to a leaf node in the decision tree represents a different combination of contextual properties. Each leaf node contains a certain distribution of instances among behavioral categories, allowing the identification of the most probable category for that leaf.

Step 4: Form the context groups. Based on Postulates 1 and 2, join tree paths into context groups if the following two conditions are satisfied:

(a) The hypothesis that the process instances in their leaves are of the same population (considering their behavioral similarity categories) cannot be rejected.
(b) If their leaves include behavioral categories that stand for similar paths but different termination states, the hypothesis that termination states for similar paths in both leaves are of the same population cannot be rejected.

Considering our bottle manufacturing running example, one of the difficulties faced is the large number of possible process paths (considering each one of the 14 machines and 40 employees who operate the machines as different paths). Furthermore, the selection of a machine and a worker at runtime is mainly done based on availability, and to a lesser extent on the context, so this choice is not expected to reflect the relevant contextual properties we are looking for. Still, this choice might affect the outcomes. To overcome this, we decided to identify a subset of very reliable employees and use only process instances they participate in for context identification. We also decided not to differentiate paths where different machines were used, but to include the time since the last maintenance operation of the machine as a contextual property.

The identification procedure is applied as follows:

Step 1*:* An initial classification of the process instances related to whether or not the process faced an event of problem identification. Clearly, this is a contextual variable that affects the process behavior. Hence, we separately performed the following steps to instances where problems were identified and to instances where production was performed without interrupts. We demonstrate the next steps with respect to the group where no problems occurred.

Step 2: Paths were clustered (disregarding machines and workers, as discussed above). The termination states were divided into two groups: (1) instances where the customer accepted the delivery without a need for a 100% inspection, (2) instances where the customer accepted the delivery after a 100% inspection, or where the customer rejected the delivery or where the delivery was cancelled. The combination of path similarity groups and termination state groups included 12 behavioral categories.

Step 3: Applying the decision tree growing algorithm resulted in the tree depicted in Fig. 3.

Each path in the tree (from the root to a leaf) represents a combination of contextual properties relevant for the behavior of process instances. Each node in the tree holds a set of process instances that can be characterized by a distribution over behavioral similarity categories. For example, node 13 stands for process instances related to products in the food and cosmetics market whose size is large and where the customer required resistance of the bottle to high temperatures. Node 12 represents process instances in the medical supplies market with special covers (children proof) where the machine used was not within a short period after its periodic maintenance (hence its maintenance state is not considered as best).

Step 4: Applying postulates 1 and 2. Due to space limitation, we only demonstrate Step 4 with respect to parts of the tree, leaf nodes 8, 9, and 13. The behavioral categories of the instances in all three nodes fall into three path similarity categories (paths 1-3) and two termination state categories, distributed as shown in Table 2.

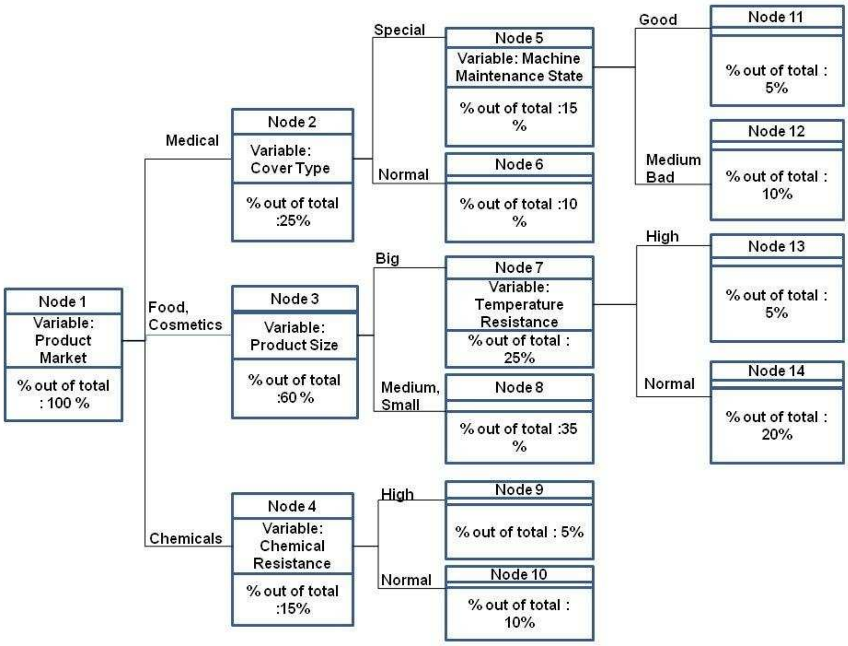

Fig. 3. Context identification decision tree

Table 2. Behavioral category distribution for leaf nodes 8, 9, and 13 (in %)

Category	Path	Termination	Leaf node		
			13	9	8
1	1	Success	13	10	7
2	2	Success	40	40	38
3	3	Success	40	42	43
4	1	Quality problems	3	3	4
5	2	Quality problems	2	2	5
6	3	Quality problems	2	3	3

Based on Table 2, the hypothesis that the instances in all the three leaf nodes are of the same population cannot be rejected, hence condition (a) is satisfied. To check condition (b), Table 3 shows the distribution of termination states for every path in the leaf nodes.

Based on Table 3, paths 1 and 2 lead to significantly different termination states in leaf node 8, as compared to leaf nodes 13 and 9. Hence, it cannot be considered in the same context group.

Table 3. Termination states for paths in the leaf nodes (in %)

Path	Termination	Leaf node		
		13	9	8
1	Success	82	77	64
	Quality problems	18	23	36
2	Success	95	95	88
	Quality problems	5	5	12
3	Success	95	93	93
	Quality problems	5	7	7

In summary, while the two conditions hold for leaf nodes 9 and 13, they do not hold for the combination including leaf node 8. Hence, leaf nodes 13 and 9 can be joined to one context group (instances with big products and resistance to high temperature for the food and cosmetics market OR with products for the chemicals market with high chemical resistance requirement), while leaf node 8 is a different group (instances with small or medium products for the food and cosmetics market).

Note that not all the existing and known contextual variables are identified as influencing the behavior (e.g., the supplier of the raw material was found irrelevant).

4.2 Suggesting Improvements to the Process Model

Phase 1 provides a grouping of process instances according to context groups. In addition, these are divided into sub-groups with similar behaviors. However, for improvement purposes a different level of granularity might be needed, both for the paths and for the termination states. The termination state classification for the purpose of context identification aims at creating a clear distinction of different outcomes. Hence, it is at a coarse granularity level, relating mainly to the hard goals of the process and possibly to a threshold over soft goal achievement levels. When attempting to suggest improvements that would affect the business results of the process, a finer granularity level is required, relating to different levels of soft goal achievement. Considering the paths, some distinctions that were disregarded for the context identification (e.g., different machines) must be taken into account, as they might affect the outcomes for a given context.

Process improvement may include three types of action:

1. **Providing criteria for path selection in a given situation**. These would rely on the paths, context groups, and outcomes achieved by process instances in the repository. Considering the granularity level defined as relevant for process improvement, process instances in each context group should again be clustered based on path similarity. These clusters are then ranked based on their average

achievement of goals, so the best performing paths for each context groups can be identified and recommended.

To illustrate path selection recommendations, below are some possible cases concerning our running example.

One situation would be a context group for which a path that uses new material and quality inspection level 3 would yield zero cases of unacceptable quality level (no customer rejects, cancellations, or 100% inspections performed). However, when a certain worker operates the machine, an average 5% defects (which is still acceptable by the customer) is obtained at the sample inspection, while other workers normally have 2% defects. As a result, the specific worker can be trained, or not be assigned to orders of that context group.

As a second example, the current process model includes a decision point where a 100% inspection can be performed (or skipped) after a problem has been identified and solved during production. It could be identified that avoiding 100% inspection at this point significantly increases the probability that the customer would reject the delivered goods. Hence, the process model would be changed so performing such inspection becomes mandatory.

A third example might identify that for a certain group of machines, when the product is for the medical market, the frequency of problems identified during production increases about four months after maintenance activities, while for other machines and other context groups it happens only after six months. This could indicate the need to perform maintenance more frequently for these machines, or to avoid using them for medical products if four months have passed since their maintenance.

2. **Addressing specific questions that might be asked**. Management may have "hunches" about possible causes of poor performance. These can be specifically addressed by analyzing path and performance for all the context groups. For example, in the bottle manufacturing process there are cases of very urgent orders for which the machine set up is done in an accelerated manner. Management wishes to check whether this accelerated set up results in decreased product quality in general or for specific context groups. Specific analysis will try to correlate the time spent for the set up activity with the outcome for different context groups.

3. **Identifying successful deviations from the existing process model as a basis for managerial decision making.** The process instances in the repository may include instances where specific ad hoc decisions were made to deviate from the "normal" process at runtime. For example, there might be cases where special quality inspection instructions were given, not compliant with the existing three levels. Since this has not been repeated enough times to get statistical significance for its results, we can only indicate that such deviations were made and the extent of their success in achieving the process goals. Such indications may be used by management, which may decide to repeat this course of action so more data becomes available for future learning cycles.

It should be stressed that we do not suggest any change to be made automatically to the process model. All the improvement recommendations should be reviewed by humans (managers, domain experts), and the performance of the improved process should be monitored.

4.3 Online Application

Learning can be performed periodically in an off-line manner. At runtime, new process instances are created and executed. The learning results should be applied to these new instances. Each new process instance should be classified to an existing context group so path selection decisions can be made according to the recommended path for the context group according to the improved process model. Some decisions (e.g., assigning a worker or a machine) are usually made based on different criteria (e.g., machine availability). However, the context group may set preferences among possible paths (e.g., prefer a certain machine out of several available for a given context group).

If a process instance cannot be classified to an existing context group (e.g., a new product for a new market or some unfamiliar external event), decisions would be made based on human (managerial) judgment. In all cases, the data of the process instance, namely, context information, specific path, and outcome, will be stored in the experience base repository and serve for future learning cycles.

5 Related Work

This chapter combines three issues that have so far been addressed separately with respect to business processes, namely, goal orientation, context awareness, and actual process paths. We claim that this combination is important in order to achieve learning and improvement over time.

The business process research area has mostly focused over the years on control-flow issues, while goals as the driving force of business processes have not been extensively used. The conceptual basis for the work presented here is the Generic Process model (GPM) framework [18, 19], which relates to goals as a fundamental part of process specification. Relying on Requirements Engineering approaches, GPM distinguishes hard goals of a process from its soft goals [19]. Incorporating goals into process specification enables assessing the ability of a process to achieve its (hard) goal, which is termed the validity of its design [22]. Understanding and explicitly specifying process goals has also been shown to be a key issue for process flexibility [21]. A similar perspective of goal orientation has been suggested by [4], but their approach deals with hard goals only.

Another approach that addresses goal oriented business process modeling is presented in [13], proposing the map representation [16] as an intentional process specification. Map representation has been assigned GPM-based semantics in [20], which highlighted the synergy gained by combining these two.

Context awareness of business processes has recently gained the attention of the scientific community. The main efforts have been devoted to identifying the relevant context of a process, to its representation in a model, and to articulating how it can affect the process execution at runtime. Context identification has mainly been done in a qualitative manner (e.g., using an onion model [17]), while the algorithmic approach used here was first reported in [7, 8]. Context representation in process model has been addressed by [11]. Some representation proposals have been made, but no agreed upon standard has emerged yet. The runtime effect of context is discussed with respect to process flexibility and variability of execution [11, 14]. In addition, context-aware exception handling in workflow systems has been proposed, where the context relates both to the process and to the specific exception [3]. To the best of our knowledge, the effect of context on the outcome achieved by a process and its utilization for learning purposes has not been developed so far.

Actual path discovery has been addressed in the process mining area. Several attempts have been made to use process mining for identifying improvement opportunities in processes. These typically relate to specific performance measures (or soft goals). Examples include [1], who address performance indicators related to time (waiting time, synchronization time) as measured in different nodes of the process model. These measures are local, but can contribute to time-related soft goals if such are defined for the process. Another work [6] relates to an extended set of performance indicators, mostly related to time spent at parts of the process, and to some extent to resource consumption. The focus of these works is on mining technology capabilities rather than on analysis of business goals. As well, context information is not considered; hence learning is only partly supported.

Another related direction deals with adaptable workflow management systems like ADEPT [15]. Such systems allow making ad hoc changes and deviations from a predefined process model at runtime. Research efforts regarding adaptable systems include mining the changes that were made [9] and supporting future changes by employing a case-based reasoning mechanism which retrieves past changes that were made to the process [23]. However, in the absence of goal specification, there is no real assessment of the level of success achieved by past changes in business terms. In addition, the similarity of the situation cannot be fully established without taking context into account. Hence, here too learning is not fully supported.

In summary, while some attempts towards supporting process improvement and establishing a learning process have been made, our work is the first to explicitly incorporate goals and context as a basis for learning.

6 Conclusion

Constantly improving business processes has long been aspired by organizations. From a business perspective, it is clear that improvement can only be established with respect to defined goals. However, organizational goals are usually discussed using high level business terms, while business process modeling and management are addressed at a technical level, and are often not goal oriented. The result of this gap is that attempts made in the business process management discipline to achieve learning are not comprehensive, and can only relate to specific issues one at a time.

The approach presented in this chapter overcomes this gap by employing a goal oriented business process model, which brings the business level goals to the technical level of process specification. Context, which is the third element taken into account, is addressed at the same technical level. Tying these three elements together, this work presents a systematic process for learning and achieving constant improvement. Addressing both hard and soft goals, our approach is expected to reduce the frequency of exceptional terminations of the process and to improve business performance over time.

The learning process we propose draws conclusions from experience gained over time while executing a business process. Comparing this kind of learning to human learning, a major difference is that humans are capable of learning from mistakes they make, by acknowledging a certain decision as a wrong decision that should not be repeated. Humans can avoid repeating a mistake that has only been made once. In contrast, our learning process is statistical in nature; hence it can only draw conclusions after a substantial number of repetitions have been made. To improve the ability of the approach to learn from episodic failures (or successes), other kinds of reasoning mechanisms (e.g., Case-based reasoning) can be used in combination with the one proposed here. Future research will develop a set of learning mechanisms that can be used in combination, so each is applied in different situations. Future research will also experiment with the learning application and validate it in real life processes.

References

1. van der Aalst WMP, van Dongen BF (2002) Discovering workflow performance models from timed logs. In: Han Y et al (ed.), Proc of the Int. Conf. on Engineering and Deployment of Cooperative Information Systems. Springer, LNCS 2480
2. van der Aalst WMP (2005) Business alignment: using process mining as a tool for Delta analysis and conformance testing. Requirements Engineering, 10(3): 198-211
3. Adams M, ter Hofstede AHM, van der Aalst WMP, Edmond D (2007) Dynamic, extensible and context-aware exception handling for workflows. In: Proc. of OTM 2007 Part 1. Springer, LNCS 4803
4. Andersson B, Bider I, Johannesson P, Perjons E (2005) Towards a Formal Definition of Goal-oriented Business Process Patterns. Business Process Management J., 11(6): 650-662
5. Davenport TH (1993) Process Innovation, Reengineering Work through Information Technology. Harvard Business School Press, Boston, MA
6. van Dongen BF, Adriansyah A (2009) Process mining: fuzzy clustering and performance visualization. In: Proc. of the 5th International Workshop on Business Process Intelligence (BPI 2009), Ulm, Germany

7. Ghattas J, Soffer P, Peleg M (2009) A formal model for process context learning. In: Proc. of the 5th International Workshop on Business Process Intelligence (BPI 2009), Ulm, Germany
8. Ghattas J, Peleg M, Soffer P, Denekamp Y (2009) Learning the context of a clinical process. In: Proc. of the Workshop on health-care processes (PROHealth 2009), Ulm, Germany
9. Guenther C Rinderle-Ma S, Reichert M, van der Aalst WMP, Recker J (2008) Using process mining to learn from process changes in evolutionary systems. Intl Journal of Business Process Integration and Management, 3(1): 61–78
10. Hammer M, Champy J (1994) Reengineering the Corporation – A manifesto for Business Revolution. Nicholas Brealey Publishing, London
11. Modafferi S, Benatallah B, Casati F, Pernici B (2006) A methodology for designing and managing context-aware workflows. In: Mobile Information Systems II; IFIP International Working Conference on Mobile Information Systems, (MOBIS), Vol 191/2005 of *IFIP*. Springer Boston
12. Nevis EC, DiBella AJ, and Gould JM (1995) Understanding organizations as learning systems. Sloan Management Review, (Winter), 73-85
13. Nurcan S, Ettien A, Kaabi R, Zoukar I, Rolland C (2005) A strategy driven business process modelling approach. Business Process Management J., 11(6): 628-649
14. Ploesser K, Recker J, Rosemann M (2008) Towards a classification and lifecycle of business process change. In: Proceedings of BPMDS 08, Montpellier, France
15. Reichert M, Rinderle S, Dadam P (2003) ADEPT workflow management system: flexible support for enterprise-wide business processes. In: Proc. Intl. Conf. on Business Process Management (BPM), Springer
16. Rolland C (2008) Intention driven conceptual modelling: In: Johannesson P, Söderström E (ed), Information Systems Engineering: from Data Analysis to Process Networks. IGI Global,
17. Rosemann M, Recker J, Flender C (2008) Contextualization of business processes. Int. J. of Business Process Integration and Management, 3(1): 47-60
18. Soffer P, Wand Y (2005) Goal-driven multi-process analysis. J. of the AIS, 8(3): 175-203
19. Soffer P, Wand Y (2005) On the notion of soft goals in business process modeling. Business Process Management J., 11(6): 663-679
20. Soffer P, Rolland C (2005) Combining intention-oriented and state-based process modeling, In: Proc. of Int. Conf. on Conceptual Modeling (ER 2005). Springer, LNCS 3716
21. Soffer P (2005) On the notion of flexibility in business processes: In: Proc. of CAiSE 2005 Workshops: Workshop on Business Process Modeling, Design and Support (BPMDS'05)
22. Soffer P, Wand Y (2004) Goal-driven analysis of process model validity. In: Proc. of CAiSE 2004. Springer, LNCS 3084
23. Weber B, Rinderle S., Wild W, Reichert M (2005) CCBR-driven business process evolution. In: Proc. Int. Conf. on Case-Based Reasoning (ICCBR'05), Chicago
24. Weijters AJMM, van der Aalst WMP (2001) Process Mining: Discovering Workflow Models from Event-based Data. In: Kröse B et al (ed.), Proc. of the 13th Belgium–Netherlands Conf. on Artificial Intelligence (BNAIC'01)