Learning Business Process Models: A case study

Johny Ghattas, Pnina Soffer, Mor Peleg
Management Information Systems
University of Haifa, Carmel Mountain 31905, Haifa, Israel
Email: GhattasJohny@gmail.com, SPnina@is.haifa.ac.il, Peleg.Mor@gmail.com

Abstract. Learning how to improve business processes is an evolutionary process that must be managed as other business processes (BPs) are managed in modern organizations. The proposed model – the learning process model (LPM) – suggests a closed-loop-model approach applied to a generic process model (GPM), which is a formal state-based and goal-based approach to process modeling. LPM strives to establish a learning process by (1) identifying goal and soft-goal states of the initial process model, (2) identifying exceptional states and incomplete state definitions at runtime, and (3) adapting automatically the process model according to the discovered states. Modifications provided by the learning process may be sufficient or may need to be complemented by non-automatic changes, when unacceptable business situations arise. The learning process also aims to adapt the current process model to possible technology, specific domain (e.g., clinical procedures at specific institutions), environmental requirements (e.g., regulations and policies), and process innovations. We demonstrate the application of LPM to a vaccination process.

Keywords: Learning, business process model, generic process model, clinical guidelines, exceptions, process flexibility, process adaptation, goals, soft-goals.

1. Introduction

In a dynamic business environment, business processes (BPs) need to be changed continuously [1-4] without affecting the production of the expected business values. The continuous business environment change, the shortening of required service time to market, the increasing number of inter and intra-organization integrations and the early adoption of new business technologies makes it impossible to fully-represent business processes during their conception time.

In this research, we postulate that it is possible to substitute the practice of fully representing business processes by a flexible adaptive form of designing, implementing, and managing business processes, through a business process learning approach. This approach would allow making a partial (minimal) definition of the business processes that would enable the organization to launch the required services with minimal time to market and seize business opportunities. Once the required business initiative is launched and operating, data regarding the process execution and the extent to which goals and soft goals are attained are collected and analyzed, and deviations between the currently defined process model and the actual business process are detected. This forms the basis for learning and adapting the process based on the day by day process enactment experience.

Such an approach is of particular importance in health-care processes, which may change on the fly in adaptation to specific patients, or change over time as a result of the availability of new knowledge. The clinical guidelines modeling research field provides several approaches for medical guidelines automation, discovery and even adaptation [5-8].

This paper envisions learning in business processes, and constitutes a first step towards the design of automated processes that (1) define and track suitable parameters about the process execution and (2) learn how these processes should improve and update the process model accordingly. We base our approach on the Generic Process Model (GPM) [9, 10] which is a formal process specification, suitable for our purposes due to its explicit representation of goals. The paper illustrates a potential model for the business process learning through a medical immunization case study.

2. Business Process learning model – a proposal

We postulate that a process model needs to relate the process goals to the workflow required to accomplish them. Such a relation is necessary in order to evaluate whether the process execution is attaining its desired outcome (i.e., goals) and the performance of the process execution (i.e., soft-goals). In order for our model to be as formal as possible, we base it on the Generic process Model (GPM).

2.1 Using a Formal BPM – the Generic process model (GPM)

The focus of analysis in GPM is a domain, which is a part of the world. We describe the behavior of the domain using concepts from Bunge's ontology [11, 12] and its adaptation to information systems [13, 14]. A domain is represented by a set of state variables, each depicting a relevant property of the domain and its value at a given time. We view a successful process as a sequence of unstable states of the domain, leading to a stable state, which belongs to a set of states that reflect the process goal. An unstable state is a state that must change due to actions within the domain (an internal event) while a stable state is a state that does not change unless forced to by action of the environment (an external event). Internal events are governed by transition laws that define the allowed (or necessary) state transitions (events).

It is possible to define the projection of a process over a sub-domain, where the set of state variables addressed by the law is a subset of domain state-variables. Then, all transitions outside the sub-domain are considered external events, and the sub-domain may be in a stable state while the process is active in other parts of the domain.

The process goal, as addressed by GPM, is the state achieved by the process. However, the goal concept is sometimes used also to describe business objectives.

GPM distinguishes process goals from *soft-goals*, which are defined as an order relation on goal states [10]. In other words, soft-goals relate to the desirability of possible states in the goal set (all meeting the condition that terminates the process) according to defined business objectives. These establish a ranking (order) among the goal states. For example, assume the goal of a process is a set of states where some medical treatment has been given to a patient, but a lower level of the patient's blood pressure following this treatment is considered better than a higher one.

Finally, GPM entails criteria for assessing the *validity* of a process, namely, its ability to achieve its goal [10]. It enables the analysis of a process to identify causes for invalidity and suggests appropriate redesign actions to eliminate these causes.

2.2 Business process learning - definition

While a process can reach its goal states through different paths, these paths may attain different soft-goal levels. In addition, while a particular path may improve a specific soft-goal it might simultaneously worsen another. Based upon the GPM process definition, the result of process instance executions may be categorized to two main categories:

(1) Valid process instances which attain some process goal state.
(2) Invalid process instances which do not reach any process goal state and terminate into exceptions (exceptions occur when a process attains an unlawful state during its execution and cannot reach its goal).

The category of valid process instances may be further divided into an undefined number of sub-categories depending on attained levels of soft-goals. It is needless to say that a process path that leads to better soft-goal levels is preferred to others that lead to lower levels of soft-goals. Hence, it is clear that an organization must strive to improve continuously its capabilities to select better process paths in order to attain better soft-goal levels as well as fewer exception occurrences in runtime, namely, fewer instances where the process fails to achieve its goal. In parallel, the organization must strive to adapt (modify) its BP models because of new knowledge generated in the environment. *Business process learning* is the organization's capability to improve path selection through experience acquired from executing the business process, measuring the attained levels of soft goals and the exceptions rate.

2.3 Business process learning- a process-based approach

We postulate that a business process learning model can be established analogously to control systems theory, which bases the continuous improvement of the behavior of a controlled system through a closed-loop system-control model [15]. The closed-loop model is based upon a real-time feedback-loop that continuously modifies the system behavior in order to minimize the output error attained during system runtime. A thorough discussion of control systems can be found in [15].

Adapting the model to BPM, at each process enactment, implies that:

(1) Soft-goals are measured and following their scores, the process model is evaluated to identify what specific segments of the state flow caused the improvement or worsening of each soft-goal.
(2) Whenever exceptions occur they are analyzed and learned lessons are used to recommend process model changes
(3) Once soft goal scores and exceptions are identified, the process law may be changed accordingly, so similar state flow will be repeated or avoided in future instances of the process. The updates may require changes in state definitions (i.e., modification of the set of relevant state variables, addition/modification of predicates defining sets of states, and/or transformation definitions).

A closed-loop BP learning approach leads to the definition of the following learning process:

(1) During process instance execution, a set of state variables that would enable the evaluation of the attained level of soft-goal and the occurrences of exceptions are collected.
(2) Process instance soft-goal levels are evaluated.
(3) Collected process datum of the specific process instance, together with soft-goal levels and exception occurrence indicators are stored.
(4) The current process instance is compared to past experiences and assigned a relative score.
(5) Future executions would use the collected information and score to select in runtime the best known path.

2.4 Learning Process model (LPM) Assumptions

For any formal conceptual model to be complete and valid, the ontological assumptions of the model need to be made explicit.

The proposed LPM relies on the following assumptions:

(1) The initial process model is valid, as far as we know (i.e., it was set in a way that is meant to attain its goals considering an expected set of possible external events independently of the process learning capability).
(2) Process mining capabilities do not affect process run time nor process performance.
(3) Process soft goals are known a-priori and are measurable through the process mining capabilities.
(4) Learning is based on the gaps between desired goals and actual execution of process instances. The process model is modified by: (a) comparing actual soft goals measurement to historical values of the process soft goals; (b) analyzing the current state flow as compared to past occurrences in order to explain the accomplished levels of soft-goals; (c) drawing required process model changes from exceptional process instances.

2.5 LPM components

We establish the following postulates:

Postulate 1: In order for the overall business process to be a learning process, it should include three main sub-processes: Acting (A) process, Documenting (D) process, and Learning (L) process, as described below.
Postulate 2: These three processes are (in terms of the GPM) projections of the business process executed by the organization upon the respective sub-process domains (Acting, Documenting and Learning process domains).
Postulate 3: The A, D, and L processes interact through a set of well-defined *commitments*, as explained below. In addition, the external environment can affect these processes.
Postulate 4: The overall BP needs to adapt according to environment inputs.

The respective five model components are hereby described:
Component 1: An acting sub-process – the process that acts in order to accomplish the goals and soft goals of the process.
Component 2: A documentation sub-process – the process that collects the necessary data from the acting process for three main purposes: (1) conditioning next actions in the current acting process instance depending on data collected in previous process instance steps and data collected from past process instances; data may also be collected from external business processes (the fifth component of the model- see below); (2) affecting actions in other process instances (current and future ones); (3) providing learning processes with data collected and processed during the process enactments (i.e., soft-goal measures and exception details).
Component 3: A learning sub-process – the process in charge of adapting the business process model. It analyzes the collected data, produces required changes to support incurred exceptions, models needed changes, and introduces changes to the BPM. Note that each one of these sub-processes has its own goals and soft-goals.
Component 4: Inter-process dependencies: The commitments between the internal business sub-processes (learning, acting, documenting) is the basis for defining a valid overall business process. The acting process commitments are: (1) to provide necessary data for documentation process; (2) to execute according to the business process model that may be changed by the Learning sub-process.
The documenting process commitments are: (1) to collect necessary data for the acting process control/decision points; (2) to provide the acting process with data collected from previous process steps, past process instances, and data collected from external business processes; (3) to collect data for the learning sub-processes.
The learning process commitments are: (1) to provide/execute necessary changes to the business process model. Note that changes (both to D and A processes) have two major sources: needed adaptations following soft-goal assessments and exceptions detected; (2) to provide visibility/traceability of changes, structured process history, reports needed for human intervention.
Component 5: External (Human) processes & commitments: These are processes that affect the L, A, and D processes through inputs that are generated by the external environment**.** External processes are of two kinds: (1) modeling new cases, processes, and introducing innovation, when new medical knowledge is available (e.g., new drug, new available immunization); (2) manual management for handling exceptions. These are scenarios where the acting, learning, or documenting processes fail to continue their executing due to anomalies or unexpected situations. In such cases,

automatic learning is not always feasible, and an external entity (e.g., a human) may intervene to correct the situation.

3. LPM Illustration through a case STUDY

We use a clinical process based upon the guidelines for immunizations provided by the Institute for Clinical systems improvements (ICSI, [16]) as a case study. We start by presenting a hypothetical local version of the generic algorithm that addresses flu vaccination and is adapted to the workflow and regulation of a particular implementing healthcare institution. Next, we map the local process flow to our learning model and demonstrate how monitoring electronic medical record (EMR) data can be used to follow whether the executed process attains its goals or reaches undesired states, and how we can learn, that is, modify the process model in order to improve the extent to which it attains its goals or avoids undesired states.

3.1 Defining a local version of the immunization process

Fig. 1 shows a hypothetical local flu immunization process that follows the ICSI clinical algorithm [16]. When the patient is vaccinated for the first time in her life, the vaccine is provided in two portions, which are to be administered in two separate visits that are spaced one month apart.

3.2 Identification of the Flu Vaccine Business Process

We map the algorithm represented above into the components of the GPM process model – goals, soft-goals, states, intermediary states, and laws.

Identification of process goals and sub-goals:

Process goals: **(G1)** "Identify an eligible child"
(G2) "Provide vaccinations to an eligible child"
Soft-goals: **(SG1)** "Do our best to make the parent accept vaccinating his child. Parent Refusal to vaccinate is an undesirable outcome". The score of this soft-goal is evaluated at the reached goal state; we consider it having a binary value: {desirable, undesirable}.
Goals and soft-goals are represented by process states in GPM. Although the process flow of Fig. 1 does not have states, only activities, we use the convention that when a process has completed an activity it is in a state named by the activity, and this state remains until execution of a new activity begins. In this way, we associate goal and soft-goal states with outputs of different steps in the algorithm, as summarized in Table 1 and as marked in Fig. 1. Note that since goal states are stable states, the goal state corresponding to G1 is step 16 in the process model and not step 3. Note also

that the soft-goal in this case is of a discrete nature (i.e., goal state desirable or undesirable), whereas in other cases it may relate to continuous values.

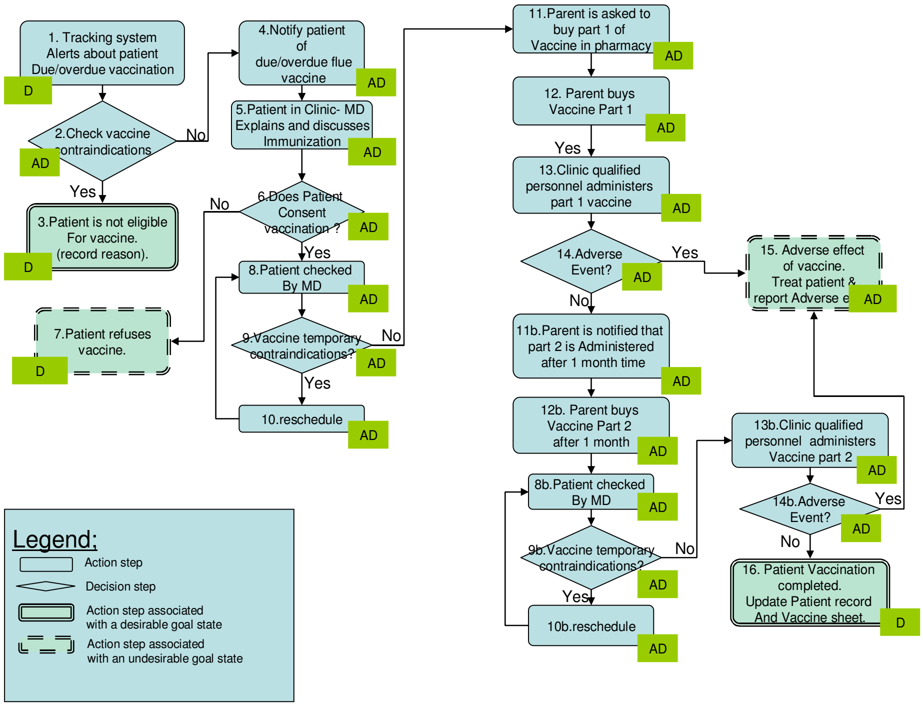

Fig. 1. Local version of the flu vaccination algorithm (first time vaccination). Each step has been identified with the sub-processes: A-Acting, D-Documenting, L-Learning. We identified soft goals: Undesirable Goal states (steps 7, 15) and Desirable Goal states (steps 3, 16)

Table 1. Goal mapping to Flu vaccination algorithm steps (algorithm represented in **Fig. 1**.

Goal states	Goal state definition	Desirable/Undesirable (soft-goal accomplishment level)	Step Outputs
G1	Identify an eligible child	Desirable	16
		Desirable (or normally expected)	3
G2	"Provide flu vaccination to an eligible child".	Desirable	16
		Undesirable	7
		Undesirable	15

Process states identification

Mapping intermediary process states is done in the same way as process goal states. Each state is identified by a set of variables, as we demonstrate in Table 2 (we present here only state variables that are changed within each state, not the whole set of variables associated with all process states). Note that the process has several states that represent goal states (S2, S6, S11 and S14); process soft-goals are evaluated through the desirability of these goal states (see Table 1).

Table 2. Business Process states mapping for the vaccination algorithm. State types may be S-stable, U-Unstable or G-Goal. The state description indicates (in parentheses) the steps whose outcome corresponds to these states. Note that the outcome of several steps correspond to the same state. In addition, not all steps necessarily cause a state transition.

ID	**State name (corresponding step outcome)**	**Type**	**State variables update**	**Next state**
S0	Initial state (1).	S	--	S1, S2
S1	Checking Vaccine contraindications (2).	U	Eligible flu V. = Yes.	S3
S2	Contraindications present (3).	G	Eligible flu V. = No.	--
S3	Patient notified-overdue vaccine (4)	S	Flu V. Status= eligible; Notified=Yes.	S3, S4
S4	MD discusses Vaccine with Patient (5)	U	Flue V. MD Check=done.	S5, S6
S5	MD checks Patient (6,8,8b)	U	Flue V. MD checking = Yes.	S7, S8,S13
S6	Patient refuses vaccine (7).	G	Flu V. Status = Refused.	--
S7	Parent asked to buy part 1 of Vaccine (11,12).	S	Flu V. Status= "waiting - part 1"	S7, S9
S8	Temporal contraindications present (9,9b,10,10b)	U	Flu V. Status= "rescheduled-contraindications".	S7, S8, S14
S9	First part vaccine administered (13)	U	Flu V. Status= "1st part administered".	S10, S11
S10	Patient Notified- part 2 within 1 month (11b).	S	Flu V. Status= "waiting for part 2".	S10, S12
S11	Adverse event present reported (14,14b,15)	G	Adverse event = Yes; Adverse event report=<…>.	--
S12	Parent buys part 2 Vaccine (12b)	U	Flu V. Status= "Prepared for part 2".	S5
S13	Part 2 vaccine administered (13b)	U	Flue V.Status= "Part 2-completed".	S11, S14
S14	Flue vaccination completed (16)	G	Flu V. Status= "completed".	--

Our Learning approach is based on collecting data from process instances and evaluating the process execution based on them. In clinical applications, the electronic medical record (EMR) may be used for documenting patient-related process-flows within the clinical system, providing a full tracking of the process state. Obviously, the EMR should include all relevant state variable data of the current and past process states. This has two major outcomes:

(1) The EMR becomes our major data base not only for patient data, but also for process flow information.

(2) We establish important foundations for a formal definition of the data required for process state tracking.

Note that data to be included in the EMR need to reflect not only the current state of the process but also enable the user to have a clear view of the process history. This is a challenge that our approach is capable of solving through the described state mapping- the inherent state variable based state definition. The described example shows how we can assure process tracking by mining the specified state variable data.
Process Law specification:
The process law specifies the possible transitions within the process flow. We mapped the possible transitions for each state in Table 2-"Next state" column. Transitions may be triggered by internal events (i.e., part of the process model) or external events (i.e., events that are generated from outside the process domain). According to GPM, internal events are sufficient for triggering a state transition from an unstable state, whereas a transition from a stable state requires an external event. In Table 2, we establish the state type for each state ("Type" column).

3.3 LPM in the Case Study

We can now demonstrate how learning can be performed.

Outcome data assessment: After the first year of executing the local process, the following conclusions were drawn at the implementing institution:

(1) The number of parents that were willing to vaccinate their children is high (state S5 reached).
(2) The number of patients that completed their vaccination was smaller than those who were willing to vaccinate. This was judged by an exception that occurred along different instances: the process did not reach a goal state, but remained stuck in step 12b (state S12).

Identifying possible causes through execution and external data analysis: Further investigation showed that some of the parents who consented to vaccinate their children could not complete their vaccinations due to shortage of the vaccine in the market. This is due to the logistical process in pharmacies, where inventory dependencies were not established between first and second portions of the vaccines. The shortage of vaccines invalidated the whole vaccination process for these patients. That is, the actual process flow was different from the model of the local process flow: step 8b (in Fig. 1) was not always reached.

GPM-based analysis: In GPM terms, state S5 was not reached for the second time, and the process was “stuck” in state S12. Surprisingly, S12 was identified as an unstable state. This implied that one of the following was occurring:
Option (1): S12 is a stable state and an external event is not being delivered, or,
Option (2): the model is missing a state between states S12 and S5 (steps 12b and 8b).

Considering the vaccine purchase within the process domain, option (2) is the valid one for the current case. The vaccine shortage is a newly discovered stable state. Note that we could consider the market as being in the environment of the process domain. Then S12 would be a stable state, awaiting an external event (vaccine purchasing).

Suggesting and evaluating possible modifications to the process model: First, the learning process can modify the modeled local system to reflect the actual flow, turning step 12 and 12b into decision points and adding a new flow into a new step 17 "market shortage" in addition to the normal flow into step 8b. In GPM terms, this results in a new undesired stable state. However, this modification by itself did not solve the problem, as parents did not wish to vaccinate their other children after their bad experiences. Nevertheless, the modification made the problem explicit and understood, so two manual solutions could be proposed:

(1) Modifying the external environment processes in order to eliminate this exception. In our case, the inventory management processes of the clinic's pharmacy were not controlled by the clinical teams and therefore this solution was not feasible. External pharmacies were willing to adapt their processes only if the clinic could give them exclusivity agreements, which contradicted current legislations.

(2) Modifying the internal process flow to eliminate the stable state. To do so, the clinic required parents to purchase both immunization portions before administering the first portion. The second portion would be stored at the clinic until its administration, ensuring that the immunization process will be completed and thus eliminating the undesirable situation of portion 2 shortage. The modified process, including the automatically learned state and manual modification, is presented in Fig. 2. Note that shortage is still possible, but either a patient gets both portions, or none.

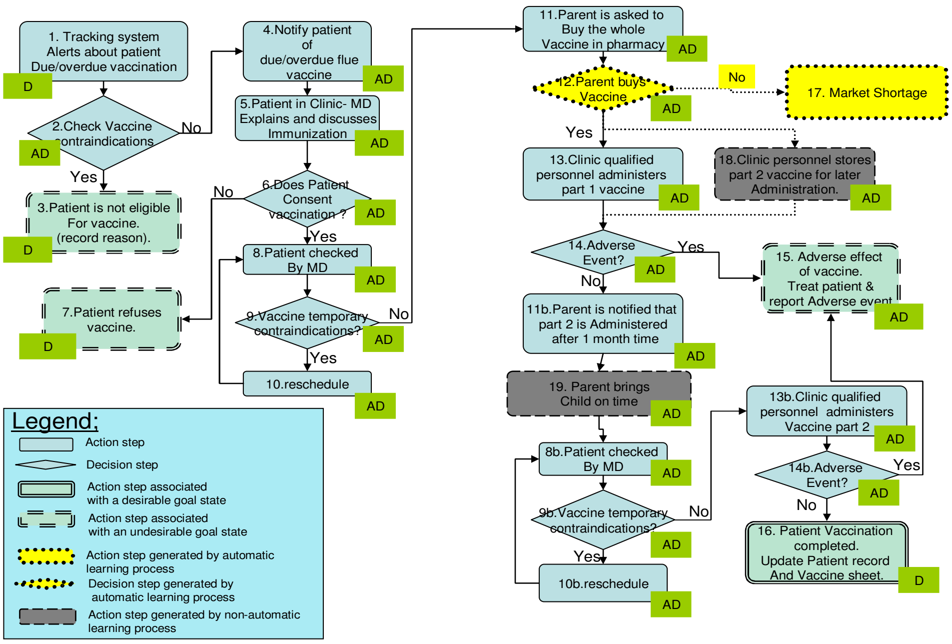

Fig. 2. Flu vaccine process modified following the learning and external changes. Note the automatic learning process has generated a new process step (step 17-market shortage) and detected a transition to it from step 12 (converted to a decision step). Note also the effect of non-automatic learning - elimination of step 12b (Parent buys part 2 vaccine) and the addition of steps 18 and 19.

4. Discussion

We illustrated how BP learning can be established based upon evaluation of soft-goals and exceptional states in process instances, and proposal of changes (manual, automatic, or semi-automatic) to the process model in order to minimize exceptions and to improve soft-goal accomplishment levels.

Examining the literature, we found several process model adaptation approaches proposed in the general BPM field [2, 4, 17, 18] as well as in medical guidelines modeling [2, 6]. Different process mining methods have been already discussed and applied in the BPM field [1, 4, 17] and in the clinical guidelines research [5, 19]. Comparison of a defined process model to the actual one discovered has been proposed [7, 19], as well as a statistic-based approach to propose ad-hoc changes both at process instance level an at process model levels [2, 3]. However, as far as we know, the examined literature is focused on understanding the actual control flow, without relating it to the process outcomes. Furthermore, it does not provide any methodology for defining the data needed for extending process mining to address both the process and its outcomes**.** Examining the process states definition of Table 2, we can easily identify the minimum EMR data needed to identify the process current state and history at any moment. Moreover, as state transitions cause changes in a subset of the domain state variables, tracking these state variables is sufficient for tracking the process state. Hence, the LPM can provide the data model required for mining a more holistic view of the actual process through the identification of the required state variables and the according documenting process. This is independent of which process mining algorithm is used and whatever process runtime model is used. We consider this to be a considerable advantage of LPM.

Learning is an evolutionary process, whose initial model is presented in this paper. As we execute the process we may face several situations which the learning process must address:

(1) Incomplete process model specification, i.e., unidentified states, incomplete state definitions, missing external events, and potential exceptions that invalidate the process execution.
(2) Causality relationships of selected process paths and achieved soft-goal values.
(3) Progress of new knowledge in the field, novel technology, or process innovations, which require changing the process

We must aim to provide the required changes at a process model proactively. To this end, LPM proposes a learning lifecycle-model based upon the process-goal approach. The contributions that we have made so far include: (a) A formal business process model-based approach that identifies the process execution state and how it attains its goals at any time. (b) A methodology for identifying what data sets need to be collected as part of process mining in order to enable process-learning. (c) A learning approach that introduces changes to the process model based upon identified exceptions, missing external events, and attainment of goals and soft-goals (the data sets described in (b)). The learning demonstrated in this paper relates to exception occurrences, while learning in general should also relate to soft-goal attained as a result of path selection. This will require the application of learning algorithms, which we intend to develop as future research. Note that while some learning may be done automatically, the modification of the business process may require, in no-error

tolerant applications (such as health care), some sort of expert confirmation before applying the modification. This does not contradict our approach.
Further work must be done to analyze potential automatic updates to the business process model. Another issue yet to be investigated is how knowledge integration could be accomplished based upon LPM.

References

1. Aalst WMP vd, Guenther C, Recker J, Reichert M. Using Process Mining to Analyze and Improve Process Flexibility - Position Paper. In: Proceedings 18th International Conference on Advanced Information Systems Engineering Workshops - BPMDS'06; 2006; 2006. p. 168-177.
2. Reichert M, Dadam P. ADEPT-flex: supporting dynamic changes of workflows without loosing control. JIIS 1998;10(2):93-129.
3. Rinderle SB, Reichert MU, Dadam P. Correctness criteria for dynamic changes in workflow systems - a survey. Data & knowledge Engineering 2004;50:9-34.
4. Weske M. Formal foundation and conceptual design of dynamic adaptations in a workflow management system. In: Proceedings of the 34th Annual Hawaii International Conference on System Sciences; 2001; 2001. p. 7051.
5. Quaglini S, Stefanelli M, Lanzola G, Caporusso V, Panzarasa S. Flexible Guideline-based Patient Careflow Systems. Artificial Intelligence in Medicine 2001;22:65-80.
6. Peleg M, Kantor R. Approaches for guideline versioning using GLIF. In: Proc AMIA Symp.; 2003; 2003. p. 509-13.
7. Peleg M, Tu SW, Bury J, Ciccarese P, Fox J, Greenes RA, et al. Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. J Am Med Inform Assoc 2003;10(1):52-68.
8. Advani A, Lo K, Shahar Y. Intention-Based Critiquing of Guideline-Oriented Medical Care. In: Proc. AMIA Annual Symposium; 1998; 1998. p. 483-487.
9. Soffer P, Wand Y. On the Notion of Soft Goals in Business Process Modeling. Business Process Management Journal 2005;11(6):663-679.
10. Soffer P, Wand Y. Goal-driven Analysis of Process Model Validity. In: Advanced Information Systems Engineering (CAiSE'04) (LNCS 3084); 2004; 2004. p. 521-535.
11. Bunge M. Treatise on Basic Philosophy: Vol. 4, Ontology II: A World of Systems. Boston: Reidel; 1979.
12. Bunge M. Treatise on Basic Philosophy: Volume 3: Ontology 1: The furniture of the world. Boston: Reidel; 1977.
13. Wand Y, Weber R. An Ontological Model of an Information System. IEEE Transactions on Software Engineering 1990;16(11):1282-1292.
14. Wand Y, Weber R. On the Ontological Expressiveness of Information Systems Analysis and Design Grammars. Journal of Information Systems 1993;3:217-237.
15. Oppenheim AV, Willsky AS. Signals & Systems, Second edition: Prentice Hall; 1997.
16. Institute for Clinical systems Improvements. Immunization, Eleventh edition,. In; 2006.http://www.icsi.org/Immunizations
17. Weber B, Reichert M, Rinderle S, Wild W. Towards a framework for the agile mining of business processes. In: Proceedings of BPM 05 BPI workshop; 2005; 2005.
18. Aalst WMPvd, Dongen BF, Herbst JL, Maruster L, Schimm G, Weijters AJMM. Workflow Mining: A Survey of Issues and Approaches. Data and Knowledge Engineering 2003;47(2):237-67.
19. Guth V, Oberweis A. Delta analysis of Petri-net based models for business processes. In: Proceedings of Applied Informatics; 1997. p. 23-32.