A Goal-based approach for business process learning

Johny Ghattas, Pnina Soffer, Mor Peleg

Department of Management information systems,
University of Haifa, Carmel Mountain 31905, Haifa, Israel.
{GhattasJohny@gmail.com, SPnina@is.haifa.ac.il, Peleg.Mor@gmail.com}

Abstract. We establish a business process learning model as part of a process lifecycle management approach. We define business process learning as the mechanism which provides the ability to perform a specific process better with time, based upon the experience acquired while executing the process, which is reflected through the accomplishment of better soft-goals. The Learning Business Process Model (LPM) relates the process outcomes, the process context and the process soft-goal measurement in order to establish proposals for paths that would improve the soft-goal outcomes. We demonstrate LPM through a process adopted from a real life scenario of a family's busy morning routine.

Key words: Business process learning, Generic Process Model, goal based process model, soft-goals, business process lifecycle

1. Introduction

Business processes usually have a life-cycle which includes distinctive phases of design, deployment, and operation, where evaluation of operational processes may lead to their redesign so a new cycle begins. This paper proposes a new model for business process life cycle, where some initially (even partially) designed process is launched, and becomes the subject of constant improvement through a Business Process Learning (BPL) approach. BPL should provide the ability to perform a specific process better with time, based upon the experience acquired. By improving process performance, we understand that while a process goal may be accomplished successfully, different grades of accomplishment may be achieved. These grades are sometimes termed soft-goals [10]. The focus of the paper is on how to utilize accumulated experience for improving the process in terms of soft-goal attainment.

In the current literature, process mining tools are used for investigating the actual practice of processes. Process mining has been applied for different objectives, such as process discovery, process conformance, process verification/validation, and others. Some authors [1][2] have integrated different machine-learning algorithms into one system to provide the potential user with different mining views (e.g., social network mining, statistical view of process paths). Others [5][6] provided a platform for dynamic process changes at different levels, so authorized users may perform ad-hoc deviations from the pre-defined process model during runtime. These deviations are recorded, and sophisticated mining capabilities are provided for analyzing change logs. However, process mining has not addressed up to now the relationship between the mined process paths and the outcomes achieved in terms of goals.

Goal-based approaches for process modeling and validation, such as TROPOS [3] assume that a process' objective is to accomplish an a-priori goal. The Generic Process Model (GPM) [9][10] extends goal-based approaches and provides precise definitions of hard and soft goals. GPM defines the goal of a process as a stable state which marks the termination of the process. A process may have several possible goal states and soft-goals establish the relative desirability of each possible goal state in terms of business objectives.

So far, we found little reference to an integrated approach of learning, which augments the entire process management lifecycle, starting at process design, through implementation, execution and adaptation to changes, by using acquired knowledge through the continuous execution of the process to improve business processes. Such framework would depart from an initial process model, and based on mining process execution logs, would produce a recommendation of the path that a process should follow in order to attain better performance of its soft-goals, given a specific context. This paper proposes a first step towards a learning-based process life cycle. The general principals and aims have been reported in [4][6].

The paper is structured as follows. We first provide the description of the proposed LPM framework in section 2. In section 3, we provide a detailed illustration of LPM through a daily life scenario of a family morning. We briefly discuss the outcomes and future work directions in section 4.

2. LPM – A Goal Based Approach

The proposed model for LPM is based upon an automated learning approach guided by the user through inputs in different steps of the processing. In this section we propose a learning algorithm, aimed at establishing the relation of outcomes (goals and soft-goals) and process path as we stated before.

2.1. The Learning Algorithm

The proposed algorithm, whose architecture is shown in Fig. 1, requires as inputs the initial process model and run-time process instances.

We consider the domain of the initial process model as being comprised of sub-domains that interact with each other, each having its goals[1] and relevant external events. M process instances are available, represented as sequences of executed steps and stored in the historical database.

For each process instance, the needed data is detailed in Table 1.

We postulate that each process instance has its own context, which is defined as follows:

Definition 1: Process context data is a tuple <I, X>. This definition includes both initial process data and external environmental effects on the process.

[1] A goal of a sub-domain is a state where the sub-domain completes its activities and becomes stable, as opposed to the process goal, where the entire domain is stable.

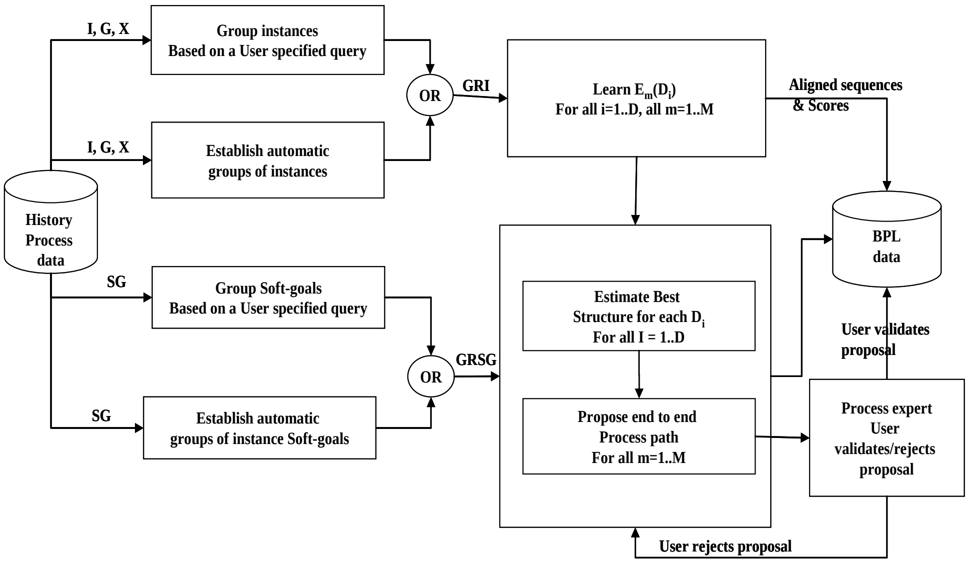

Fig. 1. LPM algorithm architecture.

Table 1. Process data per each process instance.

Data	Description
I	Initial process data: state variable values at process triggering moment.
X	The set of run-time external events, collected during the process execution.
E	The sequences of transitions ("events" in GPM) the process went through during its execution.
G	Attained goal states per each sub-process.
SG	Soft goal scores.

LPM classifies the process instances to compute different instance clusters arranged by similarity of (a) context and (b) soft-goal scores. Alternatively, a user may determine the context-based and soft-goal related grouping of instances manually, as shown in Fig. 1. Context groups, GRIi, are determined based on the process context (I, X) and goal states. Soft goal groups, GRSGi, are determined based on soft goal scores, SG. To relate paths and outcomes to the process instances, LPM identifies the sequences of events per sub domain (Em(Di)) for each context group GRIi. These sequences are pair-wise aligned and assigned a score which depends on a global alignment score, the frequency (%) of the occurrence of each pair of sequences (Fj) and the potential soft-goal score of the aligned pair. Next, the algorithm proposes to the user the best estimated path (sequence) for each sub-domain (Di), for a given context cluster. Combined over the sub-domains, it is assumed to provide the user with an overall best process path estimate, which is proposed to the user. In case the user rejects the proposal, the algorithm proposes to the user the next best alternative, and so on. The groups, sequences, and scores generated are all stored in a database, which will be enriched with more data as more process instances are generated.

3. Illustration – Family Morning Scenario

We illustrate the Learning Process Model (LPM) algorithm through a real-life process describing a family getting ready to leave the house in the morning.

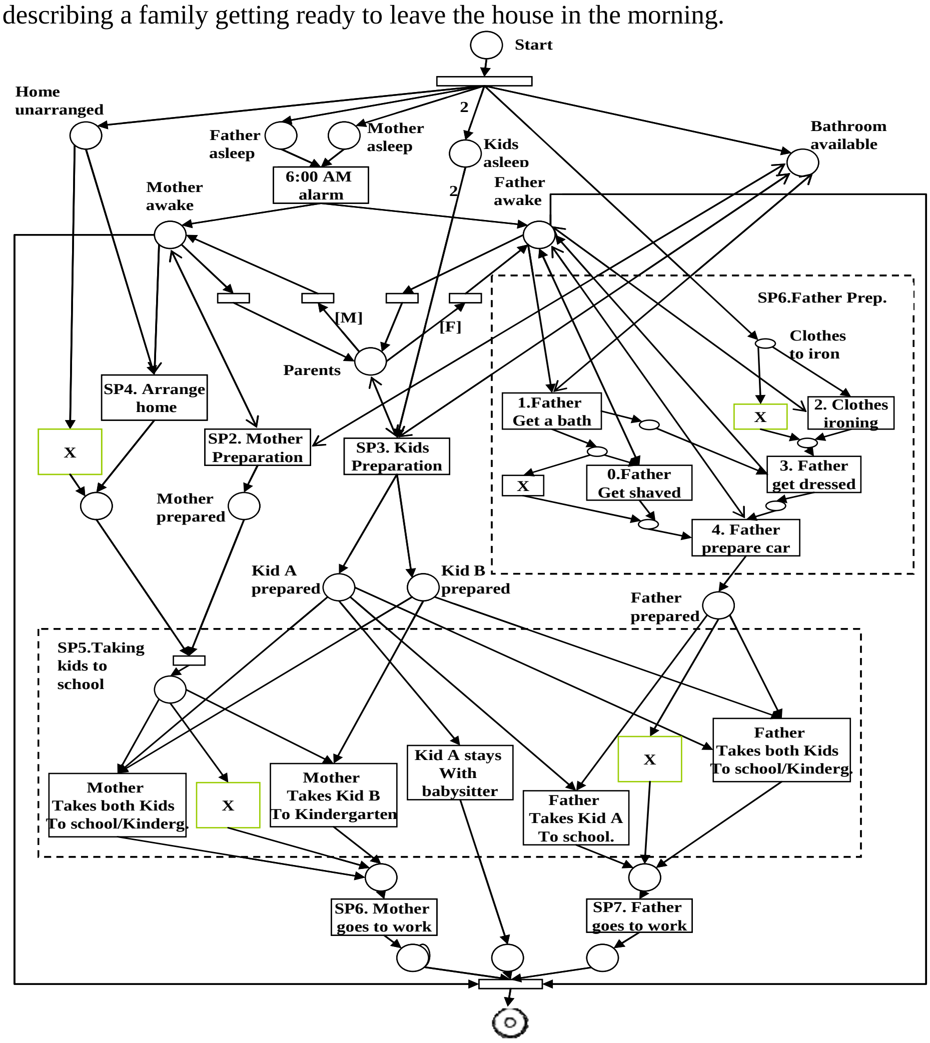

Fig. 2. Petri Net represention of the process of the "Family morning scenario". Note that the granularity of the process leaves sub-processes as "black boxes" (i.e. one single transition). We have illustrated the detailed version of two subprocesses: SP1-Father preparation and SP5-Taking kids to school. Note: Bidirectional arrows are a simplified represention of two arrows, one in each direction. Tasks "X" are skipping tasks.

In this process, both parents must bring their children to school and reach work, while complying with the constraints imposed by the environment, such as getting the kids to school by some specified hour and arriving at work in time. The process may also entail unexpected or special events such as kids' illnesses, holidays, parent illness, etc. We depart from an initial process model, based on a-priori knowledge that we already have, assumed to be complete and valid (see [8][9] for formal definition of completeness and validity of process models). A high level representation of the overall process is presented in Fig. 2.

3.1 Collecting Process Data

Identifying Sub-domains (D) and Sub-goal States (G)

For each sub-process, the process model provides a set of relevant state variables that constitute the sub-process domains $D_{i=1..n}$. Each sub-process domain has its own sub-goal states. Sample sub-goal states are provided in Table 2. Note that any sub-process may have several sub-goal states- for example, as illustrated in Table 2, sub-process Father Preparation goal states (G1.1,G1.2 and G1.3) may be distinguished only by the state variables "shaved" and "clothes ironed", which affect the soft-goals, as discussed later on.

Table 2. Sample Goal states of the different sub-process domains

Goal state ID	State variables value	Sub-process ID	Sub-process Domain ID
G1.1	Shaved= Y, Clothes ironed= Y.	SP1	D1
G1.2	Shaved= Y, Clothes ironed= N.	SP1	D1
G1.3	Shaved= N, Clothes ironed= N.	SP1	D1
G2	Mother prepared = Y.	SP2	D2
G3	Kids prepared= Y.	SP3	D3
G5	Kids at school= Y.	SP5	D5
G6	Father at work = Y.	SP6	D6
G7	Mother at work = Y.	SP7	D7

The domain (D) definition depends on the level of granularity at which we are considering the process. Higher level of granularity would require extending the process data with additional information. For example, the overall process (Fig. 2) would require for the transition "Father preparation" a single state variable ("Father prepared"), while if we like to detail the "Father preparation", we would need to consider additional state variables (such as "clothes ironed", "Father get dressed").

Identifying State Variables for Initial Process Data (I)

Table 3 shows samples of I-sets for the considered scenario. Note that state variables "clothes ironed" and "Father got a bath" are not affected by transitions in other sub-domains, while the state variable "Bath availability" reflects a dependency between several sub-processes (e.g. "Mother preparation", "Kids Preparation").

Table 3. Samples of initial process data sets.

Initial data ID	Initial data definition (set of state variables)
I_1	={Day= "Monday", Kids health="OK", Special event= "No"}
I_2	={Day= "Tuesday", Special event= "School teachers on strike"}

Identifying Run-time External Events (X)

In runtime, external events (that may or may not be expected) may occur and potentially affect the process flow or may lead the process into exceptions. Examples of external events in the considered scenario is the "6:00 AM alarm", which is external to the whole process, while "bath available" is an external event to the "father preparation" sub-process. External events such as "kid feeling dizzy", "car does not start" would result in a totally different path for several sub-processes.

Identifying Process Paths (Execution Events E)

Each process instance would go through a different sequence of events. Table 4 provides sample sequences of the sub-process "father preparation" (illustrated in Fig. 2), as found in our set of collected process instances.

Table 4. Sample sequences for "Father Preparation". Events are numbered according to Fig. 2.

Sequence ID	% of total instances	Events (E) sequences
Seq (a)	22 %	2 1 0 3 4
Seq (b)	65 %	1 0 3 4
Seq (c)	13 %	1 3 4

Identifying Soft-goal Values (SG)

A specific state in the goal may be more or less desirable than others. As an example, following Table 2, goal state G1.3 (Father arriving unshaved and with wrinkled clothes) is less desirable than G1.1 (shaved + ironed clothes). If we define Father Stress Level as a soft-goal of the process, it is straightforward to deduce that goal state G1.2 ("clothes not ironed") contributes negatively to this soft-goal, while G1.1 (clothes ironed) contributes positively (less stress) to the Father Stress Level soft-goal. Note that Father Stress Level may be affected by other sub-processes, and even other processes. We define sample soft-goals and their possible values in Table 5.

Table 5. Soft-goals for the overall process

Soft-goal	Soft-goal description	Values
SG_1	Mother work arrival	10 – "On time": arrival > 7:30 AM; 0 – "Tolerably late": 7:30- 8:00; -10 – "Unacceptable" – after 8:00 AM.
SG_2	Father work arrival	10 – "On time": arrival before any scheduled meeting & < 8:30 AM; 0 – "Tolerably late": arrival > 8:30AM, no missed meetings; -10 – "Unacceptable": arriving late to customer facing events.
SG_4	Father stress level	-10 – "High stress level"; -5 – "Medium stress level";
SG_5	Mother stress level	0 – "Normal stress level"; 10 – "happy morning".

At this stage, we have identified all basic elements of the process data: sub-domains (D), sub-goal states (G), process context (I, X), events (E) and soft-goal values (SG).

3.2 LPM Algorithm Application

In the following, we illustrate the application of the algorithm.

Clustering Process Instances into Context Groups

First, we cluster process contexts based on the process context data (I, X) and the sub-goals' set G, with the objective of identifying similarities between contexts. Sample process context groups (GRIi) of our scenario are shown in Table 6. Note that after the soft-goal groups are generated automatically, the end user would examine them and assign to each one of them a soft-goal group score, based on his domain knowledge. In addition, note that process context data is independent of G, in contrast with the process context groups which are process context data grouped by sub-goal sets G. The end user may provide each context group with a name, for example, he may choose to name GRI1 as a "Normal day" context, GRI2 "Kid ill", and GRI3 "School on strike".

Table 6. Process context groups identified for the case study.

Context group ID	Group specification	GRI_i score
GRI_1	= {"Special event= No" & Kids health="Healthy"}.	1
GRI_2	= {Kid A health OR Kid B Health="ill" }.	0.5
GRI_3	= {Special event= "Teachers on strike"}.	0

Clustering Instances Via Attained Soft-goals

LPM proceeds to cluster the process instances according to similarity of instance soft-goal scores (SG), as shown in Table 7.

Table 7. Soft-goal Groups ($GRSG_i$) established through clustering the soft-goal scores.

ID	Soft-goal group relation to goal states
$GRSG_1$	(SG_4 & SG_5 = 10) & (SG_1 & SG_2 >= 0).
$GRSG_2$	(SG_4 OR SG_5<=0) & (SG_1 OR SG_2<= 0).
$GRSG_3$	(SG4 & SG5 <= -5) & (SG1 & SG2 < 0).

Note that while the clustering is done automatically, a group name may be provided by the user, after analyzing the different clusters proposed by the automatic clustering algorithm. For example group GRSG1 would be called "a successful day" while group 3 may be named "an unsuccessful day".

Sub-processes Path Mining and Scoring

LPM aligns pairs of event sequences (E) of each sub-process per process instance, computing sequence similarity. It grades the sequences according to their similarity and their soft-goal scores, hence relating process path to process outcome. This is done for each context group (GRI_i). As an illustration, we align the sequences presented in Table 4, related to the sub-process "Father preparation", for a "normal day" process context (GRI_1).

We align all pairs of sequences, using the Needleman-Wunsch sequence-alignment algorithm [11] as shown in Table 8. While the original algorithm computes the score of each pair of sequences and considers only sequence matching (based on a similarity matrix and gap penalty), in our case we need to consider the frequency (% occurrence) and the average soft-goal score of each sequence. The overall alignment score of each pair (OAS) is calculated for a single context group (e.g., "normal day") and is a combination of the global alignment score of each sequence (as in the original algorithm), the frequency (%) of the occurrence of each pair of sequences (Fj) and the potential soft-goal score of the aligned pair (Pj).

Table 8. Event sequences alignment results.

j	Seq. pair	Aligned sequence					Alignment Score (AS_j)	Pair Frequency (F_j) %	Maximum soft-goal score P_j	Overall alignment score (OAS_j)
1	(a),(c)	2	1	0	3	4	60 %	11.1 %	0.365	6.6 %
		-	1	-	3	4				
2	(b),(c)		1	0	3	4	75 %	55.8 %	0.5	57.18%
			1	-	3	4				
3	(a),(b)	-	1	0	3	4	80 %	33.1%	0.5	36.18%
		2	1	0	3	4				

Note that in Table 8, Fj represents the frequency (%) of joint occurrence of the pair of sequences (j) being aligned, where the frequency of (a) is 22 %, of (b) is 65 % and of (c) it is 13 %. We have:

F1 = 22 % * 13 % / (22%*65% + 65%*13% + 22%*13%) = 286/2561 = 11.1 %. (1)
F2 = 22 % * 65 % / 2561 = 55.8 %. (2)
F3 = 13%* 65 %/2561 = 33.1 %. (3)

Pj represents the maximum soft-goal score that may be attained by the aligned sequence and the overall alignment score. For example P3= P(a,b)= Max(ASG(b), ASG(a))= 0.5, where ASG is the average soft-goal score of the sequence as defined in **Table 9**.
Note we have set soft-goal group scores to 1 for GRSG1, 0.5 for GRSG2 and 0 for GRSG3.

The overall score (OASj) is the normalized product of ASj, Fj and Pj. The overall score (%) represents an indication of how much the aligned pair is desirable from a soft-goal score point of view. As an example, OAS of alignment (a,c) would be calculated as follows:

OAS1=OAS(a,c)=60*11.1*0.365/(60*11*0.365+75*55.8*0.5+80*33.1*0.5)=6.6% (4)

Table 9. Average soft-goal group score, where each group is assigned a score according to Table 7. For each group and for each sequence we note the % of occurrence of the soft-goal score group ($GRSG_i$).

Sequence	Soft-goal group score			Average soft-goal group score (ASG_i)
	$GRSG_1$	$GRSG_2$	$GRSG_3$	
Seq (a)	1 (15%)	0.5 (24%)	0 (55%)	0.27
Seq (b)	1 (30%)	0.5 (40%)	0 (30%)	0.5
Seq (c)	1 (23%)	0.5 (27%)	0 (60%)	0.365

OAS scores indicate that the sequence pair alignment (b,c) is the best one. Between the two sequences composing the pair (b,c), LPM chooses the sequence whose potential soft-goal score is bigger, that in our case it is sequence (b) with average soft-goal = 0.5, as illustrated in **Table 9**.

Moreover, note that transition 2 (Fig. 2) appears in no alignment, and transition 0 does not appear in sequences (c) (which has minimal average soft-goal score). Hence, LPM would propose the user to further investigate the necessity of transition 2 in sequence (a) and the inexistence of transition 0 in sequences (c), as potential cause of negative contributions to soft-goals. LPM performs a similar analysis to all identified sub-processes, and finally combines the recommendations to an end-to-end recommended process path.

4. Conclusions

We have proposed and illustrated a framework for business process learning (LPM) based on process outcomes (goals and soft-goals). Such framework enables launching an initially defined process and continuously improving it to achieve higher levels of soft-goals, as opposed to the currently practiced process life cycle, whose phases are distinct. We illustrated how we can investigate and set the path for each specific sub-domain. In addition to an automated path selection, some cases may entail human-based thinking. While a complete reengineering might be impossible to perform automatically, our approach would enable the user to identify situations where none of the existing paths leads to satisfactory outcomes, or where a large portion of process instances result in exceptional states.

This research relates and complements other research domains such as process mining and case based reasoning (CBR). Referring to process mining, the proposed framework, LPM, can complement frameworks such as ProM [1] or ADEPT2 [7]. While these frameworks provide a platform for understanding the actual processes, they do not tie this understanding to the process outcomes. LPM can establish this relationship and provide them with an integrated process management lifecycle.

Case-based reasoning (CBR) is an approach to problem solving and learning [2], which has been applied to process modeling by different authors, and hence has some relation to our research objectives. New problems are dealt with by drawing on past experiences, described in cases stored in case-bases, and by adapting their solutions to the new problem situation. In [5][12], a modified CBR approach – a conversational CBR (CCBR) aims to allow run-time changes to process models by the end user. LPM can improve the performance of CBR-based approaches by providing a formal definition of similarity criteria as proposed earlier based on Context and Goal

similarities combined with the normally used occurrence statistics in CBR variants researches. Still, the proposed approach does not establish a methodology to analyze the relevance of past experience to specific process contexts, nor does it use instance outcomes (soft-goal levels) in its analysis.
Our model is currently being validated and extended based upon different case studies. Future work includes its application to different domains, such as the clinical domain.

References

1. A.J.M.M. Weijters, W.M.P. van der Aalst, B. van Dongen, C. Günther, R. Mans, A.K. Alves de Medeiros, A. Rozinat, M. Song, and E. Verbeek. Process Mining with ProM. In M. Dastani and E. de Jong, editors, Proceedings of the 19th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC). (2007).
2. Aamodt, A. and E.P., "Case based reasoning: foundational issues, methodological variations and system approaches". AI Communications, 39-59. (1994)
3. Giorgini P., Kolp M., Mylopoulos J., Pistore M.. The Tropos Methodology: an overview. In Methodologies And Software Engineering For Agent Systems. Kluwer Academic Publishing. (2004).
4. J. Ghattas, P. Soffer, M. Peleg : Learning Business Process Models: A case study. Business Process Modeling Conference Workshop: 1st International Workshop on Process-oriented Information Systems in Healthcare, Brisbane, Australia. September 2007.(2007)
5. Kim, J.H., Suh W., Lee, H.: Document-Based Workflow Modeling: A Case-Based Reasoning Approach. Expert Systems with Applications, Vol.23(2) 77 – 93. (2002)
6. M. Peleg, P. Soffer, J. Ghattas: Mining Process Execution and Outcomes. Business Process Modeling Conference Workshop: 1st International Workshop on Process-oriented Information Systems in Healthcare, Brisbane, Australia. September 2007. (2007)
7. M. Reichert, S. Rinderle, U. Kreher, P. Dadam: *Adaptive Process Management with ADEPT2*. Proc. ICDE'05, (2005)
8. P. Soffer, M. Kaner, and Y. Wand, Assigning Ontology-Based Semantics to Process Models: the Case of Petri Nets, CAiSE 2008 (to appear).(2008).
9. P. Soffer, Y. Wand. Goal-driven Analysis of Process Model Validity. In: Advanced Information Systems Engineering (CAiSE'04) (LNCS 3084); 2004. p. 521-535.(2004).
10. P. Soffer, Y. Wand. On the Notion of Soft Goals in Business Process Modeling. J. of Business Process Management;11(6):663-679.(2005)
11. S. B. Needleman, C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology, 48:443–453. (1970).
12. Weber, B., Rinderle, S., Wild, W., Reichert, M.: CCBR-Driven Business Process Evolution. In: Proc. 6th International Conference on Case-Based Reasoning (ICCBR 2005), Chicago, USA (2005)