A Framework for Modeling and Enabling Reuse of Best Practice IT Processes

Hamid R. Motahari-Nezhad, Sven Graupner, Claudio Bartolini

HP Laboratories
HPL-2010-107

Keyword(s):

Best Practice Processes, Process Modeling, Process Reuse, ITIL

Abstract:

Best practices frameworks such as ITIL provide a generic description of best practice processes that are intended to be followed by people. These processes are refined into more concrete steps before they are actionable. The refinement often is specific to the organization where the process is adopted, as well as people who are enacting the process. Modeling best practice processes is challenging. On one hand, these processes need a high-level, abstract representation. Current process modeling languages are too rigid for modeling them. On the other hand, automation of the enactment of these processes among people requires formal models. In this paper, we propose a framework for modeling best practice processes at three levels: user-level, formal process model level and machine representation level to support the collaborative and ad-hoc refinement of process models as well as the automation of their enactments. We also propose an approach to learn from the past enactments of processes to enable reuse of organizational domain knowledge.

External Posting Date: September 6, 2010 [Fulltext] Approved for External Publication
Internal Posting Date: September 6, 2010 [Fulltext]
To be published and presented at the 1st International Workshop on Reuse in Business Process Management (RBPM 2010), New York, USA, 13 September 2010

A Framework for Modeling and Enabling Reuse of Best Practice IT Processes

Hamid R. Motahari-Nezhad, Sven Graupner, Claudio Bartolini

Hewlett Packard Laboratories,
Palo Alto, CA, United States
{hamid.motahari, sven.graupner, claudio.bartolini}@hp.com

Abstract. Best practices frameworks such as ITIL provide a generic description of best practice processes that are intended to be followed by people. These processes are refined into more concrete steps before they are actionable. The refinement often is specific to the organization where the process is adopted, as well as people who are enacting the process. Modeling best practice processes is challenging. On one hand, these processes need a high-level, abstract representation. Current process modeling languages are too rigid for modeling them. On the other hand, automation of the enactment of these processes among people requires formal models. In this paper, we propose a framework for modeling best practice processes at three levels: user-level, formal process model level and machine representation level to support the collaborative and ad-hoc refinement of process models as well as the automation of their enactments. We also propose an approach to learn from the past enactments of processes to enable reuse of organizational domain knowledge.

Keywords: Best Practice Processes, Process Modeling, Process Reuse, ITIL.

1 Introduction

Best practice frameworks such as ITIL (IT Infrastructure Library)[1] [1] describe a general set of guidelines and processes for IT management. In particular, ITIL is a best practices framework for IT Service Management (ITSM). ITIL V3 covers the lifecycle of offering ITSM as a service including phases of service strategy, design, transitions, operation and continual service improvement. Each lifecycle phase describes a number of specific processes such as supplier management (part of service design) and incident management (part of service operation). These descriptions are intended to be followed by people in organizations with respective work domains. In order to allow variation and flexibility of organizations, best practices frameworks provide their descriptions at a rather high and generic level. We refer to these processes as *descriptive processes* as opposed to *prescriptive processes* which are processes specified using existing business process languages.

[1] www.itil-officialsite.com

The prescriptive processes are often enacted with workflows (they may involve human interaction, as well). Descriptive processes are interpreted, refined and enacted by people. They are often used in *collaborative*, *ad-hoc* and *agile* work environments where the exact process steps become known as the work progresses among the people but may not completely be known ahead of time.

Currently, there are two main categories of tools supporting best practice processes in ITIL. On one hand, there are enterprise-grade tools that support the whole service lifecycle with processes (such as HP Service Manager[2]). Those tools encode a specific interpretation of the processes from best practices in internally coded logic. They impose rigid processes onto the organization and do not support flexibility and the ad-hoc nature of such process. They do not capture people interactions in the context of process enactment. On the other hand, there are productivity and office automation tools that are used among people through which processes are enacted. This category of tools usually has no explicit support for processes definition and hence has no visibility into its execution. A major issue in both approaches is *information loss* and the inability to reuse organizational domain knowledge on how people refine descriptive processes and enact them.

In this paper we focus on the problem of providing a modeling framework that addresses the following challenges: first, how to model descriptive processes supporting people to define, refine and enact processes in a collaborative and ad-hoc manner, and second, given the fact that descriptions from the best practice process provide only informal high-level guidelines, how to capture the knowledge of how people refine and enact those processes. Learning and representing process enactments enables the reuse of organizational domain knowledge.

The work in this paper complements our previous work, the IT Support Conversation Manager (ITSCM) [2]. ITSCM supports people in the context of ITIL incident management process while they define, refine and enact the incident management processes in a collaborative, flexible and ad-hoc manner. A conversation is a container for the interactions of people and process steps. ITSCM allows monitoring and tracing how people perform their job. This paper describes the ITSCM's multi-level framework for modeling descriptive processes. The framework consists of a user level, a formal process model level and a machine representation level. The paper also outlines an approach for learning the organizational knowledge on refinement and enactment of descriptive processes by people to enable reuse.

The paper is structured as follows. Section 2 presents the lifecycle of descriptive processes. Section 3 presents the modeling framework for descriptive processes. Section 4 presents the progress for learning refined process templates from previous process enactment instances. Section 5 discusses related work and open challenges.

2 Lifecycle of descriptive processes

The concrete form of a descriptive process from best practice frameworks is often influenced by two considerations: organization adaptation, for accommodating the

[2] www.managementsoftware.hp.com

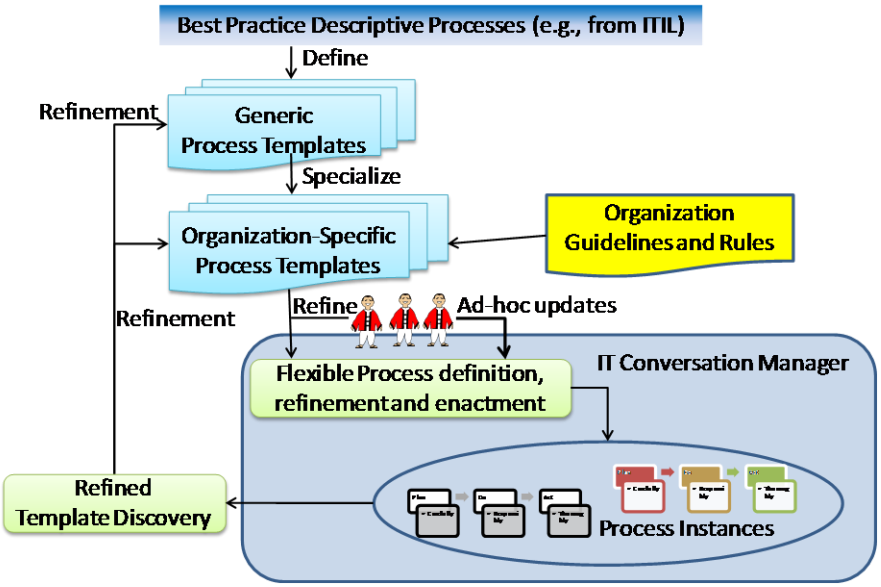

Fig 1. The proposed lifecycle of best practice processes in organizations

specifics of the organization that is adopting the framework, and people who are following the process. Fig. 1 shows the lifecycle of best practice processes for an organization. In particular, we envision creating process templates for descriptive processes. Process templates translate general textual guidelines into a structured form, which can be used by people to instantiate the process.

The current approach for creating process templates for descriptive processes is manual. It is performed by domain experts via reading the textual descriptions of the best practice processes. In order to support domain experts, we have introduced a framework that identifies the set of important concepts for people-intensive processes in [3]. Experts can use this framework as a guideline for extracting concepts and their relationships. Further research is required to enable the fully automated extraction of these process-related concepts and their relationships from the textual description in best practice processes.

In some organizations, generic process templates may be adapted to create organization-specific process templates, which then can be more specific compared to the generic process templates. People working within a best practice framework may choose to use one of the generic or one of the organization-specific templates to start a conversation (e.g., for handling an IT incident). They may also start a new conversation specifying the process in an-hoc manner. In our framework, there is no distinction between a process definition and a process instance, as processes are running as soon as they are partially defined by people in a conversation. A process definition may not exist in advance at the required level of details, and so the concrete definition emerges as the result of a collaboration among people performing a specific task (e.g., handling the incident) during the process enactment.

The concrete best practice processes live in the process instance repository, which is a repository containing all past enactments of processes. In the case of ITSCM, the repository contains the set of past conversations. Therefore, identifying how best practice processes are enacted in an organization requires the understanding of process instances in this repository. In this work, we propose to learn the process model from past process instances using a reverse engineering approach similar to process mining methods [4]. The inferred process model can be used to create new organization-specific process templates or to update existing templates. This approach does not only enable the understanding of concrete enactments of best practice processes in an organization, it also fosters reuse of organizational domain knowledge that is captured from people during past process enactments.

3 Modeling Framework for Descriptive Processes

We propose a three-level framework for descriptive processes to support people in the flexible and collaborative definition of descriptive processes, as well as to provide automated support for the enactment of these processes. The framework consists of the user level, the formal process model level and the machine representation level.

The user level. At the user level we define a set of concepts and corresponding relationships for best practice processes. Our observation shows that knowledge workers usually do not work based on formal or graphical process models. Using existing graphical modeling languages often leads to over-specification of process models. A qualitative user study [5] shows that a semi-formal modeling approach is preferred by users for modeling reference models such as ITIL processes. We, therefore, intentionally do not introduce an explicit process model with a graphical notation. We capture a descriptive process in terms of a set of concepts and associated relationships called the *process concept model*. It defines the key high-level concepts of "Process", "Task", "Item" (process resources, documents and artifacts), "Actor", "Role", and "Event". Each concept can also include a set of properties. The set of relationships includes the generic "has" and "is-a" relationships with more process-centric concepts such as "assigned-to", "receives" (inputs), "produces" (outputs), "depends-on" (tasks relationships) and "reacts-to" (events). Note that the process concept model can grow beyond the built-in concepts. For instance, users can add new tasks that are not part of the built-in processes. Users can introduce new concepts and relationships and choose to add them to a conversation-specific or to the central library of concepts. The enriched process models can include relationships between various processes in a best practice framework as well as pointers to service lifecycle phase(s) to which the process belong (refer to [3] for more details).

The formal process model level. The process concept model is still abstract and therefore not actionable. In order to provide automation support for enactment, we construct a corresponding process model based on dependency graphs from the process concept model. The process model is constructed considering the "depends-on" relationship between tasks in the process concept model. We also provide update operations on the model, such as add, delete, update (see [2,3] for details about this layer). Operations allow ad-hoc updates of the process model based on changes in the

process concept model. The resulting process model is used by an enactment engine in ITSCM to provide functionality such as sending notifications, sending reminders and enabling the tracking of progresses.

The machine representation model. We choose to store the process model with the process instances as RDF graphs (see [2] for an example of a process description using RDF). Our process concept graph and the formal process model based on dependency graphs are RDF graph models as well. The process model and process instances can be updated by adding, removing or updating the concepts and relationships in the RDF graphs. The main motivation for choosing RDF is that the process graph in RDF is extensible, and it also allows incorporating information and relationships in the process that are not necessarily related to the process enactment, but are needed to link the process to the containing project, customer information and service such as the phase of the service within which the process is enacted, etc. RDF graphs also allow applying a variety of reasoning and querying techniques on process instances such as SPARQL. At the implementation level, we use the Jena toolkit (http://jena.sourceforge.net/) which includes a variety of model stores for the repository as well as libraries for query and inference (refer to [2,3] for examples and details on this layer).

4 Learning Refined Process Templates from Process Instances

Process instances are enriched process models in our framework. They include a detailed formal process model. This process model is often refined and customized for a specific enactment. Therefore, it can include process tasks and concepts that are not part of the built-in set of concepts in the process template. For each type of process (e.g., incident management process) we want to infer the common (frequent) process template refinements from a set of past process instances. Refinements that occurred during enactment can be applied on various aspects of the process including roles that performed a process step, the actual process flows that updated the process structure or conditions that led to a particular step. The process instances are also a good source for extracting statistical information on the enactment of processes such as how long it takes to enact a process step or the process itself, in average, or how many people are involved, etc.

In particular, the refinement of process templates includes two steps of analyzing process instance traces (which are RDF graphs), and updating the process templates. In the analysis step, we learn a process model which is annotated with conditions that lead to a particular step (by looking at the attributes of previous steps), and statistical information on how many instances contain a particular step, whether this step is new or a built-in process step. Then, we define the following operations to update the original process template.

Adding and removing steps: If a certain number of instances (above a user-defined threshold) include a new step that is not included in the template, the method suggests to add the activity to the process template in the same order that has been observed in instances. If a certain step of the template has not been used in many process instances (above a threshold), it suggests removing the step from the template.

The step refinement: this operation suggests updating the details of the step in the template based on the analysis of the enactment information of the step. In particular, the runtime information of the process activity is updated including the involved roles, average enactment time, number of instance of the template that include this activity, etc. In addition, if a step is refined into more concrete steps, the concrete sub-steps are extracted and their frequency is computed. This information is included in the template to enable their reuse.

Refine Structure: this operation analyzes the process instance and identifies cases where the order of activities is different compared to that of the template. In such cases, it adds metadata to the activities in process templates providing alternative ordering of activities and the frequency of such re-ordering in the realizations.

5 Discussion and Related Work

The existing work on process mining [4] focuses on learning a process model from the set of process instances. In that context, a process instance is often a sequence of steps. In the context of descriptive processes, the process instances are richer (in our case they are represented as RDF graphs) with complex relationships. In addition, we need to learn not only the process model in terms of the graph but also information such as conditions or explanations (from textual descriptions in conversations) which lead to choosing a specific process steps. The next set of challenges is related to applying the refinement information on the original template. One issue is that not always the execution of a descriptive process agrees with the definition in the template in terms of the process structure. We have taken a first step in updating the templates as described above by annotating the template, and providing alternative enactment orders during enactments. However, more research is needed on strategies to update the template to avoid making the templates complex but rather easy to understand for people and to reuse.

References

1. L. Hendriks and M. Carr, "ITIL: Best Practice in IT Service Management," in: Van Bon, J. (Hrsg.): The Guide to IT Service Management, Band 1, London u. a. 2002, pp 131-150.
2. H.R. Motahari-Nezhad, et al: IT Support Conversation Manager: A Conversation-Centered Approach and Tool for IT Incident Management., Proc. of EDOC (2010), to appear.
3. S. Graupner, et al.: Making Processes from Best Practice Frameworks Actionable. DDBP 2009: Second International Workshop on Dynamic and Declarative Business Processes, September 1, 2009, Auckland, New Zealand (2009).
4. W. van der Aalst, et. al.: Workflow mining: a survey of issues and approaches. DKE Journal, 47(2):237–267, 2003.
5. C Taylor, C Probst, Business Process Reference Model Languages: Experiences from BPI Projects. Proceedings of INFORMATIK, 2003.
6. Orbus Software, The iServer ITIL Solution, http://www.orbussoftware.com/business-process-analysis/products/itil-solution/.