Demonstrating Flexible Support for Knowledge-Intensive Processes with proCollab

Nicolas Mundbrod and Manfred Reichert

Institute of Databases and Information Systems
Ulm University, Germany
{nicolas.mundbrod,manfred.reichert}@uni-ulm.de
http://www.uni-ulm.de/dbis

Abstract. Knowledge-intensive processes (KiPs) are driven by knowledge workers utilizing their skills, experiences, and expertise. As these processes are emergent and unpredictable, their support constitutes a big challenge. For coordinating and synchronizing the various tasks of a KiPs, knowledge workers still rely on simple task lists like, e.g., to-do list or checklists. Though task lists are intuitive, their current implementations are very ineffective: tasks are neither made explicit nor are they personalized or synchronized. In addition, no task management lifecycle support is provided and media disruptions frequently occur. This tool demonstration presents the proCollab framework supporting a wide range of KiPs (e.g., projects and cases) through integrated, lifecycle-based task management. In particular, proCollab task trees support the provision of task list templates and to constitute digital task lists of any kind. For customization, it further allows integrating domain-specific methodologies as well as configuring task lists at design and run time. Overall, the proCollab framework shall improve coordination among knowledge workers, increase work awareness, and record valuable coordination efforts.

Keywords: knowledge-intensive processes, task management, task list, checklist, to-do list, knowledge worker

1 Introduction

Residing in sensitive key business areas, such as research, healthcare, or engineering, knowledge-intensive processes (KiPs) have become the centerpiece for creating value in many companies [3]. While driving KiPs, knowledge workers utilize their skills, experiences, and expertise to cope with sophisticated tasks. Thus, the systematic and sustainable support of KiPs constitutes a prerequisite for achieving business goals. KiP support, however, still poses a big challenge. KiPs can be characterized as *non-predictable*, *emergent*, *goal-oriented*, and *knowledge-creating* processes [1, 3]. Consequently, KiPs have not been fully supported by contemporary process-aware information systems so far. Instead, knowledge workers still rely on simple task lists (e.g., to-do lists, checklists) to coordinate their work [6]. But these instruments are error-prone and ineffective. Moreover, knowledge workers suffer from media disruptions and a lack of lifecycle-based task management preventing them from reusing existing artifacts (e.g., task lists) in similar KiPs [4].

The *proCollab*[1] framework aims at the systematic and sustainable support of KiPs. As tasks constitute the key objects regarding planning and quality assurance for knowledge workers, proCollab provides process- and lifecycle-based task management [3], empowering knowledge workers to collaborate more effectively. To make use of best practices and knowledge gained in the context of comparable KiPs, proCollab provides *process* and *task list templates*, which knowledge workers may *instantiate* on demand. To foster the reuse of these templates, a context-aware approach for configuring task list templates is included as well. Knowledge workers may then easily configure task lists either at design or run time. To increase work awareness in addition, proCollab encompasses a flexible state management concept. The latter may be customized to optimally align KiPs, task lists, and tasks with domain-specific requirements. Finally, the proCollab framework aims to improve coordination among knowledge workers, prevent media disruptions, and record valuable coordination efforts and knowledge.

2 proCollab and its Significance to the BPM field

To enable a generic, but customizable support for KiPs, the proCollab framework relies on *processes*, *task trees*, and *tasks* as the conceptual pillars for representing a wide range of KiPs as well as task-based artifacts used by knowledge workers. To establish and provide lifecycle-based task management support in the context of KiPs [2], processes are refined to *process templates* and *process instances*, whereas task trees are refined to *task tree templates* (with *task templates*) and *task tree instances* (with *task instances*) respectively. Figure 1 illustrates the relationship and interplay of those components.

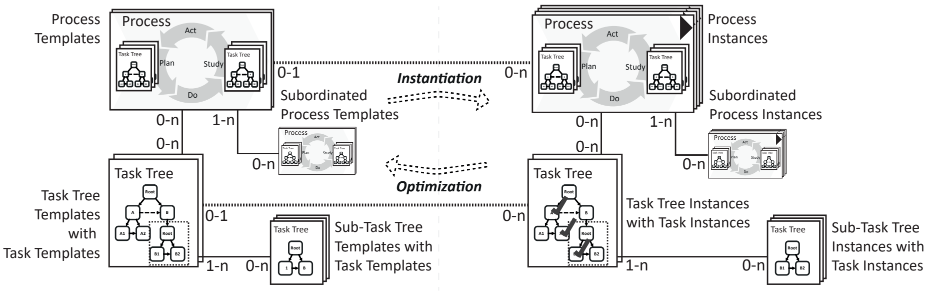

Process Templates
Process
Act
Plan
Study
Do
Task Tree
0-1
Subordinated Process Templates
Instantiation
0-n
Process Instances
Subordinated Process Instances
0-n
0-n
1-n
0-n
Optimization
Task Tree Templates with Task Templates
0-1
Sub-Task Tree Templates with Task Templates
1-n
0-n
Task Tree Instances with Task Instances
Sub-Task Tree Instances with Task Instances
Root
A
B
A1
A2
B1
B2

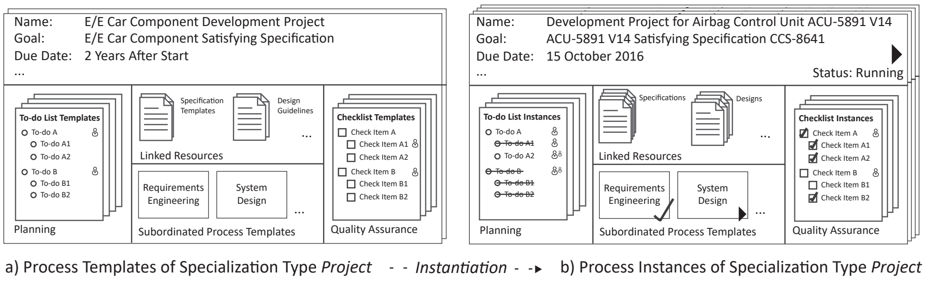
Fig. 1. Overview of proCollab Components

Process templates and *task tree templates* enable knowledge workers to accelerate planning of their tasks based on best practices. Every process template may have several subordinated process templates and feature various properties,

[1] **Pro**cess-aware Support for **Collab**orative Knowledge Workers

conditions, and linked resources. Every process template may be linked to an arbitrary number of task tree templates. A task tree template, in turn, contains *task templates* and, optionally, subordinated task tree templates. Thereby, it reflects best practices for planning (to-do list) or quality assurance (checklist) in the context of KiPs. For example, process template for clinical surgeries may contain a task tree template for ensuring patient safety during surgeries.

At run time, knowledge workers collaborate in the context of *process instances*. A process instance may represent a running *project*, a *case*, or any *kind of knowledge worker collaboration*. Moreover, it has properties like a goals and it features resources (e.g., documents). A process instance may have subordinated process instances to let knowledge workers focus on specialized sub-goals. Every process instance may comprise multiple *task tree instances* with corresponding *task instances*. A task tree instance constitutes the generic representation of common task-based artifacts (e.g., a to-do list) in use. For example, an automotive engineering project with to-do lists for planning and checklists for quality assurance can be supported by a respective proCollab process instance with corresponding task tree instances (of type "to-do list" and "checklist") (cf. Figure 2).



a) Process Templates of Specialization Type *Project* - - *Instantiation* - -▸ b) Process Instances of Specialization Type *Project*

Fig. 2. Visualization of Process Templates and Instances from the Automotive Domain

In general, knowledge workers may create a process instance based on a process template or starting without any template at all. If a process template is instantiated, the linked task tree templates are instantiated, too. Further, the created task tree instances are linked to the corresponding process instance. In the context of a particular process instance, knowledge workers may instantiate further task tree templates or add "blank" task tree instances on demand.

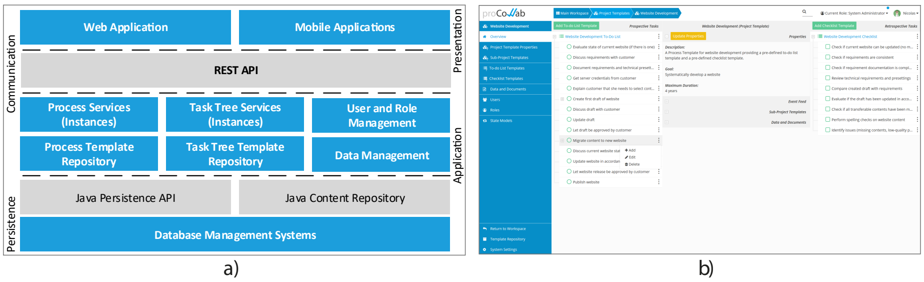
To support a wide range of application scenarios, proCollab employs *specialization types* enhancing the generic data structures of processes and task trees. Depending on the chosen specializations, *processes* (e.g., projects) and *task trees* (e.g., to-do lists) may feature additional properties (e.g., states), conditions, constraints, or assignments. For example, a proCollab process may be adapted to a specific automotive project (cf. Fig. 2). If a task tree instance is linked to the specialization type *to-do list*, it will be interpreted as a *to-do list instance* with corresponding user interface representations and properties. To ensure a coherent use of specialization types, the latter may be linked to each other. For example, the specialization types *to-do list* and *to-do item* are interlinked and, hence, task trees of type *to-do list* may only contain tasks of type *to-do item*. To further increase domain-specific support, proCollab employs a generic state management for its stateful key components. This enables us to integrate domain-specific methodologies as well as to manage different types of proCollab components in a controlled manner. Additionally, proCollab provides a configuration concept for task tree templates. The latter enables the efficient configuration of task list templates in accordance to the given application context. Due to a lack of space, both the state management and the configuration concept are presented in the screencast of this demonstration in more detail (cf. Section 3).

Concerning the significance of proCollab to the BPM field, the design of a systematic KiP support still constitutes a big challenge. While predictable business processes can be well supported by process-aware information systems based on pre-specified process models (*doing-by-design*), unpredictable and emergent KiPs require a degree of flexibility traditional systems are not able to provide due to the limitations of model-driven approaches. In turn, the proCollab framework relies on the *design-by-doing* approach necessitating different concepts and functions to support a wide-range of KiPs in a generic, but still domain-specific way. In comparison to declarative process execution systems like Declare or DCR Graphs [7] and approaches based on CMMN [5], proCollab focuses on the stateful and flexible task (list) management support for KiPs and knowledge workers. In particular, knowledge workers themselves shall drive proCollab processes and, as a consequence, the management of tasks and task lists, too.

3 The proCollab Tool and its Maturity

To prepare empirical studies and to evaluate the technical feasibility of the proCollab framework, we developed the proCollab tool. The latter is realized with Java EE 7 and relies on a MVC-based architecture (cf. Fig. 3). In particular, the application logic layer represents the core of the tool realizing the key services of the proCollab framework. The REST-based interface enables web and mobile applications to communicate with these key services. Finally, the web application based on AngularJS provides state-of-the art user interfaces (cf. Fig. 3) to the knowledge workers interacting with the proCollab tool.

The current version of the proCollab tool enables knowledge workers to manage the proCollab key components, i.e., they may manage KiPs in the shape of common projects or cases (i.e., proCollab processes) including task trees embodied as, e.g., to-do lists or checklists. Moreover, knowledge workers may configure task tree templates or instantiate process/task tree templates on demand (see screencast). Furthermore, the proCollab tool allows integrating domain-specific state models for templates and instances of processes, task trees, and tasks. In turn, the assignment of users to tasks, the integration of process resources (e.g., documents, structured data), and the synchronization of tasks and processes are currently in development to be integrated in the web application. Based on its architecture and the technologies in use, the proCollab tool can scale up well and even be deployed in cloud-based environments to serve a significant amount of knowledge workers concurrently.



Website Development To-Do List

○ Evaluate state of current website (if there is one)
○ Discuss requirements with customer
○ Document requirements and technical presett...
○ Get server credentials from customer
○ Explain customer that she needs to select cont...
○ Create first draft of website
○ Discuss draft with customer
○ Update draft
○ Let draft be approved by customer
○ Migrate content to new website
○ Discuss current website stat
○ Update website in accordan
○ Let website release be approved by customer
○ Publish website

Description:
A Process Template for website development providing a pre-defined to-do list template and a pre-defined checklist template.

Goal:
Systematically develop a website

Maximum Duration:
4 years

Website Development Checklist

□ Check if current website can be updated (no m...
□ Check if requirements are consistent
□ Check if requirement documentation is compl...
□ Review technical requirements and presettings
□ Compare created draft with requirements
□ Evaluate if the draft has been updated in acco...
□ Check if all transferable contents have been m...
□ Perform spelling checks on website content
□ Identify issues (missing contents, low-quality p...

Fig. 3. Architecture and Screenshot of the proCollab Tool

Finally, we prepared a screencast available on `bpm2017demo.procollab.de` to demonstrate proCollab. The screencast illustrates the key aspects of the proCollab framework using the scenarios of developing a website and conducting a surgery.

4 Conclusion

This demo presented the proCollab framework and its proof-of-concept tool. The latter is implemented as a scalable, cloud-ready, and web-based application platform. This allows users to collaborate in the scope of KiPs and to simultaneously manage proCollab key components to receive optimal KiP support. Based on the tool, user experiments will be conducted to systematically analyze whether KiPs can be successfully evolved when using the proCollab framework. Overall, we believe that the proCollab framework will improve coordination among knowledge workers, increase work awareness, and record valuable coordination efforts.

References

1. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-Intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches. J on Data Semantics 4(1), 29–57 (2014)
2. Mundbrod, N., Beuter, F., Reichert, M.: Supporting Knowledge-Intensive Processes through Integrated Task Lifecycle Support. In: Proc. EDOC 2015. pp. 19–28 (2015)
3. Mundbrod, N., Kolb, J., Reichert, M.: Towards a System Support of Collaborative Knowledge Work. In: BPM 2012 Workshops. LNBIP 132 (2013)
4. Mundbrod, N., Reichert, M.: Process-Aware Task Management Support for Knowledge-Intensive Business Processes: Findings, Challenges, Requirements. In: Proc. EDOCW'14. pp. 116–125 (2014)
5. Object Management Group: Case management modeling and notation (cmmn) 1.1 (2016), `http://www.omg.org/spec/CMMN/1.1/`
6. Pryss, R., Mundbrod, N., Langer, D., Reichert, M.: Supporting medical ward rounds through mobile task and process management. Inf Sys and e-Business Management 13(1), 107–146 (2015)
7. Reijers, H.A., Slaats, T., Stahl, C.: Declarative Modeling—An Academic Dream or the Future for BPM? In: Proc. BPM 2013, pp. 307–322. Springer (2013)