

Pertuniti: Subprocess Modeling and Hierarchic Case Management

Johannes Tenschert and Sophia Marmaridis

Pertuniti GmbH,
Martin-Behaim-Str. 10, 90489 Nuremberg, Germany
{johannes,sophia}@pertuniti.de

Abstract. Knowledge-intensive processes are typically scattered across many systems with no clear system of record. Support systems have to consider informational relations of process instances on different granularities, and different aspects of the work require different characteristics in regard to flexibility. Today, adaptive case management is typically either tailored towards flexible process models or ad-hoc activities for a lot size of 1. Pertuniti connects both approaches by implementing cases as a set primarily of groupware artifacts for complete flexibility, subprocess modeling to automate aspects of the work, and flexible case data management considering hierarchies of cases. This way, we can reduce routine work or guide through regulated activities, while deviations and ad-hoc activities remain traceable.

Keywords: adaptive case management · subprocess modeling · data modeling · knowledge-intensive business process · process flexibility

1 Background and Significance to BPM

Adaptive case management systems (ACMSs) support knowledge workers in performing emergent, unstructured work. Keith Swenson introduced the term in *Mastering the Unpredictable* [4]. There is no clear definition, which led to many different types of ACMS from modeling in BPMN or CMMN [3] to systems similar to social software [6], e. g. Pertuniti. However, these differences are not driven by a lacking definition, but by different requirements even within knowledge work [7]. We focus on complex, communication-oriented work, i. e. on the collaboration model in Davenport's classification of knowledge-intensive processes [2].

Pertuniti is a commercial ACMS that targets knowledge-intensive processes, and initially had a clear emphasis on ad-hoc processes which we currently extend to modeled, flexible processes. It resulted from a PhD thesis [5] on how to transparently combine structured and ad-hoc processes with a focus on communication. Pertuniti shows characteristics of groupware and social software to capture the expectations of supporting ad-hoc activities, and provides subprocess modeling for routine activities to be performed within the same environment.

Activity streams show all activities performed within a case regardless of whether they were manual or automated.

While capturing the whole process in a model is often desirable for traceability, automation, and continuous improvement, it is typically not sensible for a knowledge-intensive process. However, knowledge work rarely contains no frequent or repeatable characteristics. Our approach allows to import well-known best practices of the BPM community, and to combine different modeling paradigms and ad-hoc activities within the same overall process¹. The focus on ad-hoc activities combined with activity streams allows to transparently extend and deviate from modeled routines. This is especially useful for processes with regulated process aspects around a set of flexible project management core activities, e. g. medical quality management. This nature is also present for example in automotive, aviation, insurance, or finance.

2 Subprocess modeling and hierarchic case management

We introduce subprocess modeling and hierarchic case management based on the running example of a research coordinator. These activities are highly knowledge-intensive and typically not known in advance. The focus is more on knowledge than on process management. A screencast of the features outlined in this paper is available at <https://www.pertuniti.com/events/bpm-2020>.

2.1 Adaptive case management

Pertuniti provides artifact types and features similar to popular groupware, e. g. integration with external calendar and contacts clients. Cases can capture all master data, contacts, tasks, events, documents, interactions, and relations between all case artifacts. Cases can be hierarchical, and may contain an arbitrary amount of subprocess instances. This way, we want to facilitate that all case artifacts can be captured or referenced within one system of record.

We emphasize on collaboration by adopting activity streams that are prevalent in social software. These are available on different levels: all visible cases, all case activities, all subprocess activities, and all artifact activities. Knowledge workers can track their own and team member actions for coordination, but activity streams can also pose as an audit trail.

All artifacts provide an EAV schema for adding attributes on demand. To facilitate discussions and capture additional information, notes can be added on artifact and case level.

For automation, we provide integration via web standards like CalDAV and CardDAV, user-defined document templates, and subprocess models. Document templates can be created by end users in a low-code designer. They can be multilingual and apply case and contact data either from the case context or user-defined inputs. Subprocess models allow modeling small routine aspects or full processes within the context of a case.

¹ Subprocess instances happen within the context of a case, but their execution is isolated. BPMN models and DCR graphs may capture different aspects of the case.

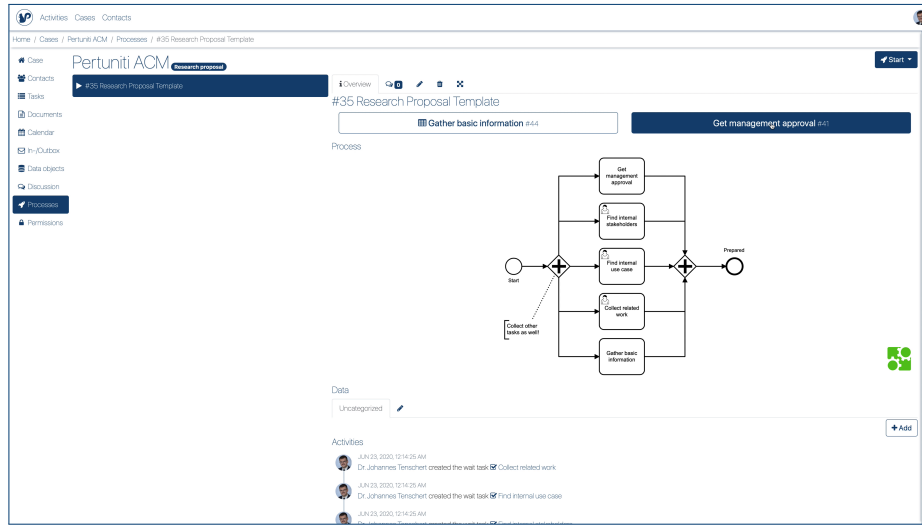


Fig. 1: Running subprocess of a case with different types of tasks

2.2 Subprocess modeling

All modeled processes in Pertuniti are subprocesses. These have a state and their execution is loosely isolated, i. e. they could use case data to communicate, but each subprocess instance also has its local context. In the screencast, you see a subset of BPMN. Currently, we also finalize support for accepting petri nets and DCR graphs, on the client side based on [diagram-js](https://github.com/bpmn-io/diagram-js)².

This approach allows small low-code automation of routine aspects of an overall flexible process. As ad-hoc activities are always possible and subprocess models are performed within the context and activity stream of a case, it also allows to transparently deviate from and to extend more elaborate processes. Moreover, it allows to combine structured or regulated processes with flexible project management. Process models may merely, and for auditors transparently, guide through automation, while unique, knowledge-intensive work is performed in a traceable way within the same context.

Figure 1 shows a running subprocess instance consisting of plain activities, where performance is intended primarily for guidance and implicit documentation, form activities to gather user input, and case task activities that are managed outside of the process instance. Case task activities are displayed in the same views as plain case tasks, and finalizing them advances the corresponding process instance as well.

² <https://github.com/bpmn-io/diagram-js>

2.3 Hierarchic case data management

Hierarchic case structures are prevalent in BPM, e.g. subprocess activities in BPMN, and in project management [1]. In project management, these are typically task-level hierarchies or hierarchically managed tasks.

In process instance hierarchies, instances typically share information via message passing, e.g. on initiating and finalizing an instance. In Pertuniti, we allow a shared context, e.g. via inheriting attributes from parent cases, and gathering information like tasks from subcases in a consolidated view. Inheritance is different to modeled cases or processes: it is defined in an ad-hoc fashion via a flag on individual attributes.

Currently, the implementation allows inheriting parent case data to subcases via flags on attributes, and aggregating subcase data in the parent case (tasks). Moreover, child subprocesses can export information to the parent case, and import information from the parent scope. Hence, we capture all directions and sources of definition of data exchange in case hierarchies.

2.4 Technical details

Pertuniti is a single-page web application written in Angular and Typescript on the client side, and primarily Java on the server side. We store information in a database, and larger blobs in more scalable storage, e.g. an S3 back end³. On the server side, it has an internal message passing system to route changes immediately to client side subscribers that are notified via web sockets. While this approach allows that all clients typically see the same and up-to-date information, this also lead to additional implementations of BPM engines, e.g. a subset of BPMN and petri nets. In the future, we intend to provide an interface for external BPM engines to exchange context information, activity streams, and update notifications.

3 Maturity

The basic approach of Pertuniti stems from a PhD thesis [5], and has been evaluated in regard to traceability [5] and features for different types of knowledge workers [7]. It is used professionally by a set of users that guide further development. The new aspects of notation-agnostic subprocess modeling and hierarchic case data management are still prototypical. However, all features are implemented multi-lingual and in an event-driven environment, i.e. all changes are pushed immediately to web clients and connected devices. Already, all activities of the BPM engines are completely traceable.

The aspects of subprocess modeling and hierarchic data management are primarily intended to support regulated processes. As a first domain, we focus on medical quality management of SMEs. Here, companies already need a large set of documented processes that typically are not yet automated or supported, and

³ Blobs are decoupled from individual storage providers.

the most value-adding activities are creative product and service development, i. e. knowledge-intensive work with regulated aspects. We will further evolve usability and implementation details with representative customers.

4 Conclusion

The goal of Pertuniti is to introduce methods and best practices of BPM into knowledge-intensive processes without restricting them to predefined models. We want to support processes that contain aspects with varying flexibility requirements in a single system of record.

For that, we support completely ad-hoc activities and managing groupware artifacts next to subprocess modeling. Subprocess modeling is not restricted to a single notation or modeling paradigm. Subprocess instances can be used for automation or to guide through processes. Hierarchic case data management facilitates sharing and synchronizing context for different aspects of an overall process. Exchanging context information between different levels in the case hierarchy does not require predefined models. All generated and ad-hoc artifacts are displayed within the same views. Hence, documents may be generated, and manually edited, while all changes remain traceable on the same artifact. We provide traceability via activity streams on different granularities.

References

1. Guidance on project management. ISO Standard 21500:2012 (Sep 2012)
2. Davenport, T.H.: Thinking for a Living: How to Get Better Performances and Results from Knowledge Workers, chap. How Knowledge Workers Differ, and the Difference It Makes, pp. 25–38. Harvard Business Press (2005)
3. Object Management Group: Case Management Model and Notation (CMMN). Tech. rep. (2016), <https://www.omg.org/spec/CMMN/1.1/PDF>
4. Swenson, K.D. (ed.): Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done. Meghan-Kiffer Press (2010)
5. Tenschert, J.: Speech-Act-Based Adaptive Case Management. Ph.D. thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen (2019)
6. Tenschert, J., Lenz, R.: Agora - Speech-Act-Based Adaptive Case Management. In: BPM Demo track, CEUR Workshop Proceedings (2016)
7. Tenschert, J.C., Lenz, R.: Evaluation of WfMC Awards for Case Management: Features, Knowledge Workers, Systems. In: Daniel, F., Sheng, Q.Z., Motahari, H. (eds.) Business Process Management Workshops. pp. 107–120. Springer International Publishing, Cham (2019)