Subject-Oriented Adaptive Case Management

Extending Subject-Oriented Business Process Management to Knowledge-Intensive Cross-Enterprise Business Processes

Matthias Kurz[1] and Matthias Lederer[2]

[1] DATEV eG, Paumgartnerstr. 6-14, Nürnberg
[2] Universität Erlangen Nürnberg, Lehrstuhl Wirtschaftsinformatik II, Lange Gasse 20, Nürnberg

Abstract. Adaptive Case Management is a Business Process Management approach that is quickly gaining the attention of practitioners and scientists. In an effort to examine how Subject-oriented Business Process Management relates to Adaptive Case Management, this contribution proposes extending an existing ACM approach inspired by multi-agent systems with the capability of defining temporal-logical dependencies between tasks using Subject-oriented Business Process Management.

Keywords: adaptive case management; knowledge work; metamodel; subject-oriented process management; multi-agent systems.

1 Introduction

Classical Business Process Management (BPM) offers a rich and tested set of methods for well-structured and routine processes. The Adaptive Case Management (ACM) concept is an increasingly popular concept which promises to bring the benefits of BPM to the area of weakly structured knowledge-intensive business processes [1, 2]. Contrary to classical BPM systems which focus on automating business processes, ACM provides capabilities to adapt processes during runtime. "This form of runtime flexibility allows process participants to respond to challenges or new requirements that were not considered during designing the business processes." [3]

However, both classical BPM as well as ACM assume that all process participants work towards the same goal. This assumption is increasingly becoming unsustainable as processes are more and more spanning multiple enterprises. In order to address this issue, [3] extends the ACM-approach of [4] based upon the paradigm of Multi-agent Systems (MAS). [3] proposes a distributed approach for managing and supporting knowledge-intensive cross-enterprise processes (KXBP) as well as a corresponding case metamodel. Although being a first step towards a comprehensive KXBP methodology for ACM, [3] currently focusses on defining and breaking down the work within a case. Therefore, the KXBP methodology offers no way to specify the temporal-logic dependencies of tasks.

This manuscript addresses this gap from a conceptual perspective by using the Subject-oriented Business Process Management (S-BPM) approach as a method for

A. Nanopoulos and W. Schmidt (Eds.): S-BPM ONE 2014, LNBIP 170, pp. 123–132, 2014.

defining the temporal-logical dependencies of a KXBP case. As this Subject-oriented ACM (S-ACM) approach relies on the S-BPM methodology, S-ACM not only extends the ACM-based KXBP approach of [3] but also demonstrates how S-BPM and ACM can be combined. Thus, this contribution demonstrates how S-BPM can be applied in knowledge-intensive cross-company business processes that may be changed while executing these processes.

2 Adaptive Case Management

ACM was first made popular by the well-renown book [1]. [2] complements this book with practice reports and case studies. Although major principles of ACM are outlined in [1], it provides no concrete method for using ACM in a real-world environment. This gap has been closed by the method described in [4] that integrates classical BPM and Case Management with the Enterprise 2.0 paradigm. As the method from [4] serves as the foundation of this contribution, this section presents an excerpt from [4].

In ACM, knowledge workers are no longer expected to follow strictly defined business processes regardless of their suitability for a given problem. Instead, they are empowered and encouraged to adapt the case behavior if necessary. According to [4], the case behavior is described using a case process. Therefore, changing the case behavior means changing a case process. This includes changing the processes of running cases. This kind of runtime flexibility is a key characteristic of ACM. It allows case workers to respond to new challenges which arise after a case has been started.

Each case is represented by a case workspace and is assigned objectives the case is expected to achieve. The workspace contains a process which is constituted by a hierarchy of tasks. Tasks assist in coordinating knowledge work between multiple case participants. [4]

Tasks are not the only object type of a case workspace. Workspaces also contain artifacts like documents or hyperlinks. These artifacts may be added to, removed from, or modified in a case workspace. While automation is not the primary objective of ACM, workflows may be linked to tasks in order to provide automation to those parts of a case that are unlikely to change and, therefore, can be automated efficiently. [4]

Every case object may be created from scratch. For improved efficiency, the method advises to use object libraries for storing and retrieving commonly used case objects. Similarly, case workspaces can be instantiated from predefined templates stored in the template library. This instrument allows to standardize and manage similar cases while retaining a high degree of flexibility for the knowledge worker. [4]

Once several instances (cases) of a case template have been completed, it is advisable to review the respective cases for common changes which should be integrated back into the templates. This adaptation of case processes is a vital instrument for the continuous improvement of case templates. [4]

3 Cross-Enterprise Adaptive Case Management

The ACM method described in section 2 provides little guidance how to support the autonomy of the participants from multiple organizations. This section briefly introduces the KXBP approach described in [3] which supports this kind of autonomy.

With "many competing, mutually inconsistent [definitions]" [5] for the terms *multi-agent systems* and *agents* circulating, the KXBP approach is based on the following definition: "Multi-agent systems are those systems that include multiple autonomous entities with either diverging information or diverging interests, or both" [12]. Therefore, an agent in the context of this contribution is an autonomous knowledge worker who pursues both his or her own interests as well as those of his organization while participating in a case along with other agents from his or her own organizations or other organizations. [3]

When multiple organizations participate in one case, there are essentially two types of case objects. *Common case* objects (cco_c) refer to objectives, tasks, and artifacts that are visible to other organizations involved in a case. While common objectives are used to manage and measure the output of the case, common tasks are used to coordinate the case work. The case output or intermediate results are stored in artifacts like documents. *Private case objects* (pco_c) are used by single organizations ($pco_{o,c}$) or their agents ($pco_{a,c}$) in order to efficiently follow their own agenda within the contractual or legal boundaries and to define the internal behavior of an organization within the case. Private case objects are not visible to members of other organizations. [3]

The different types of case objects constitute perspectives on a case. There are three possible perspective types in a case:

- Every case has a common perspective (COP). The COP is constituted by the set of all common case objects of a case (CCO_c). This perspective contains all common objectives and the basic case structure which is shared by all case participants. [3]
- Each organization may optionally have its private organizational perspective (POP) for every case its members contribute to. POPs introduce elements from the organization's own agenda and proprietary best practices that are useful for achieving the objectives. [3]
- Similar to organizations, agents may optionally have a private agent perspective (PAP) for each case they participate in. [3]

These three perspectives are combined when determining the view for an individual participant. For any given agent a of the organization o, the view $v_{c,a}$ on the case c is calculated as follows:

$$v_{c,a \in membersOf(o)} = CCO_c \cup PCO_{o,c} \cup PCO_{a,c}$$

By adding private case objects like objectives, tasks, or temporary documents to the POP / PAP, private perspectives allow defining behavior that is proprietary to and only visible to the respective organization / agent. Common case objects are used to coordinate cross-organizational case behavior and to define common artifacts like documents. The different perspectives may each reside on separate computers in order

to avoid the necessity of a central computer that has to be trusted by all participating organizations. This way, each organization can be sure that only common perspectives leave the control of the organization. [3]

4 Subject-Oriented Adaptive Case Management

4.1 Case Behavior

The following sections extend the KXBP ACM approach summarized in section 3 by leveraging the S-BPM method for defining the case behavior which refers to the temporal-logical dependencies between tasks. Therefore, the resulting approach is referred to as *Subject-oriented ACM* (S-ACM) in this paper.

The case behavior comprises (1) the *proposed case process* and (2) *the executed case process*. The suggested way for achieving the case objectives is defined in the *proposed case process* which provides a temporal-logic relation between the tasks of a case and thereby constitutes the case behavior. Proposed case processes are introduced in section 4.3. As emphasized in section 4.2, agents are free to deviate from the proposed case process whenever deemed applicable. The *executed case process* records which tasks have been executed during the case lifetime. While the proposed case process may be modified before[1] and during the entire lifetime of a case, the executed case process is automatically being created and extended during the lifetime of a case. Section 4.4 discusses executed case processes in more detail.

Tasks are only started by agents or by external events. If an external event occurs, a task associated with the event may be started and processed by an automated program agent.

4.2 Allowing Ad-hoc Changes

Contrary to Workflow Management Systems (WFMS) and BPMN 2.0 ad-hoc sub processes [6], proposed case processes only suggest the order of and the dependencies between tasks. In order to allow for run-time flexibility, agents may deviate from the case process and introduce ad-hoc changes. That way, knowledge workers are empowered to adapt the case to the individual requirements of particular cases which were not known when defining a case template.

While ad-hoc changes are vital for ensuring run-time flexibility, they can lead to deadlock situations if the process model is changed ad-hoc. E.g., the task *wait for approval* will never be completed if the preceding task *approve* is deleted by an ad-hoc change. Deadlocks may happen due to dependencies that are less obvious like input preconditions of tasks that expect a certain process state. If agents are be empowered to make ad-hoc-changes to the proposed case processes without having to consider deadlock situations, tasks cannot assume a specific process state. This can be

[1] The proposed case process may be defined in a case template which is used for instantiating a particular case. Thus, the proposed case process of a given case may be modified / defined before instantiating this case.

achieved by adding a test to each task. Such tests are invoked automatically whenever a task is executed and check whether the respective task can be executed in the current process state. If not, the task execution is stopped with a message explaining the reason.

4.3 Proposed Case Process

There is a vast number of notations and paradigms for defining the temporal-logic dependencies between tasks in a process (e.g., flow-oriented [7], object-oriented [8], and subject-oriented [9]). With ACM relying upon the cooperation of multiple individuals, S-ACM uses the subject-oriented paradigm and notation proposed by [9] which is "[…] a stakeholder- and communication-oriented paradigm that roots in the observation that humans usually use standard semantics of natural language […] when they describe what they are doing in a business process" [10].

Subject-oriented business processes comprise two types of models. (1) *Interaction diagrams* (ID) show the communication between the roles in a process (subjects in S-BPM) of a process. (2) *Subject-behavior diagrams* (SBD) define the sequence of activities within a subject. This concept of distinguishing between the externally observable behavior of a process role (the communication) and the internal behavior of a process role is highly compatible for defining the case behavior, as the case process is typically used for coordinating the work between multiple agents. By focusing on the communication between subjects, the case process allows coordinating the dependencies between the tasks of the participating agents while ensuring the agents may decide how to complete their tasks.

With multiple individual agents potentially being able to play the same role, roles are assigned to subjects in the proposed case process. Using the S-BPM notation makes introducing ad-hoc changes to the parts of the proposed process affecting the own role simpler for agents playing this role, as these agents typically focus on the process behavior of their own role. This maps well with the concept of autonomous agents introduced in section 1.

Fig. 1 shows the corresponding SBD for the involved roles *project manager* and *requirements engineer*. These SBD only contain the tasks (function states in S-BPM) of the roles and the corresponding communication primitives (send and receive states in S-BPM). Assuming the communication between the roles is necessary for ensuring the temporal-logical dependencies between tasks, these figures depict the minimal possible process defining the case behavior.

Introducing proposed case processes as an instrument for specifying the temporal-logical relationships between tasks means that there are two views on the tasks. This has the following implications.

- The task hierarchy and the proposed case process are two perspectives on the same process and not two synchronized yet distinct models. E.g., whenever a task is added or removed in the proposed case process, this change is immediately visible in the task hierarchy as well.

- Tasks may be created without being assigned or proposed to a role / subject. In the proposed case process, these tasks are associated with the role *unknown*.
- Not all tasks of the task hierarchy must occur explicitly in the proposed case process, as they implicitly are part of the respective subject's SBD. These tasks may be executed at the agents' discretion.

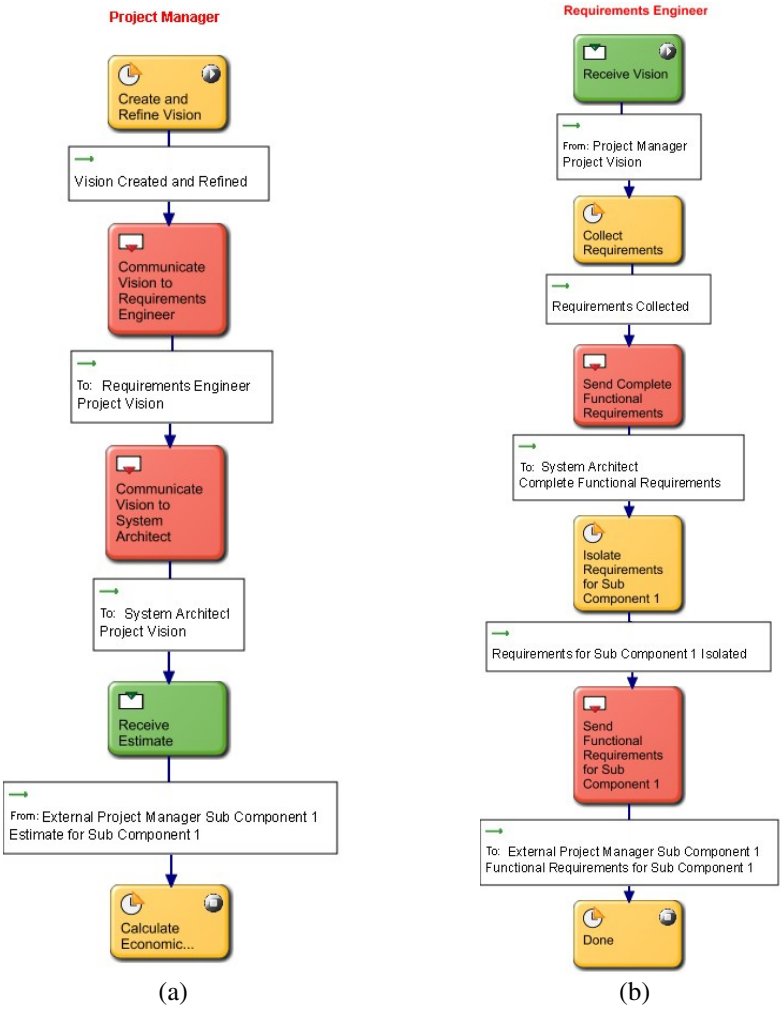


Fig. 1. Proposed case process subject behavior diagram of the roles *Project Manager* and *Requirements Engineer*

4.4 Executed Case Process

The executed case process stores the sequence of tasks which have been executed during a case. It serves three primary purposes:

- During case execution, it represents the current state of the case processes. With the activity stream of the metamodel described in [3] already recording all interactions with a case, the executed case process essentially is a view on the activity stream which filters out all activities that are not directly related to task execution.
- Once a single case has been completed, it serves as an instrument for checking whether the compliance rules have been adhered to. Checking the compliance of tasks is particularly important, as ACM intentionally provides a high degree of flexibility due to the focus on ad-hoc changes. While necessary for knowledge-intensive business processes, this makes violations of compliance rules more likely.
- After multiple cases originating from the same case template have been completed, it is possible to compare the executed case process of the individual cases with the proposed case process of the case template. This allows identifying issues and potential improvements and thereby substantially simplifies continuously improving case templates. [4] refers to this as *cross-case adaptation*.

The executed behavior of a case is amended whenever an agent executes a task. The subject interaction diagram (SID) of Fig. 2 and the SBD of Fig. 3a and Fig. 3b show the executed case process for the proposed case process introduced in section 4.3. As the executed case process contains the executed tasks only, the SBD encompasses no conditional behavior. Similarly, the SID contains only those subjects which already executed tasks. Furthermore, while subjects are assigned to roles in the proposed case process, the agents playing these roles are known in the case process.

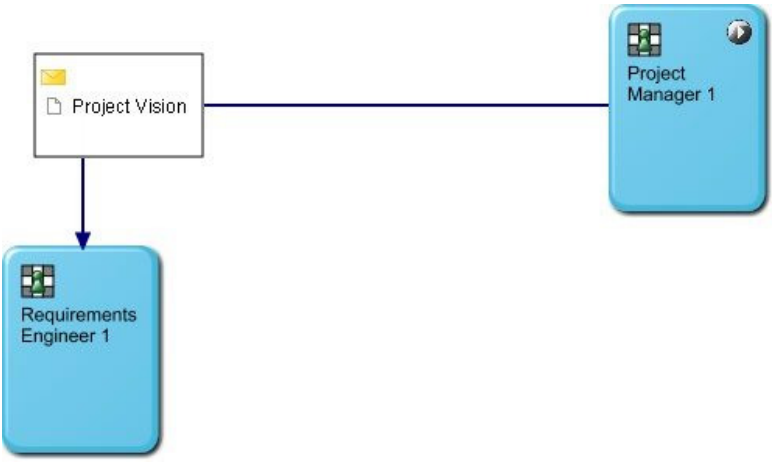
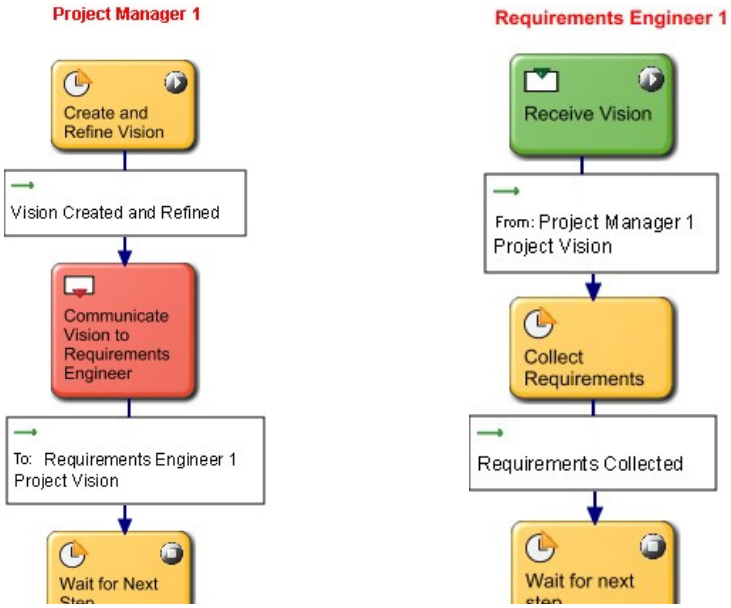


Fig. 2. Executed case process interaction diagram

The task *wait for next step* in Fig. 3a and Fig. 3b is a placeholder which ensures that the SBD is syntactically correct. It indicates that the next action is determined by the agents of the case.

Essentially, the executed case process of S-ACM fulfils a similar function as the ModelAsYouGo approach which allows agents to incrementally build a case process by modifying the internal behavior of subjects representing the respective agents [11].



(a) (b)

Fig. 3. Executed case process subject behavior diagram of the agents *Project Manager 1* and *Requirements Engineer 1*

4.5 Extended Metamodel

Considering the temporal-logical relationships between tasks in the S-ACM approach requires extending the metamodel described in [3] which provides no means for defining these relationships. Fig. 4 shows how the UML metamodel of [3] needs to be extended. New classes and associations are highlighted with a gray background. The following extensions have been made:

- As external events may start associated tasks of program agents with process the respective tasks (cf. section 4.1), it is necessary to include the *external event* class along with the corresponding relations *associatedWith* and *processedBy*.
- Tasks can have sub tasks refining other tasks. Therefore, the *subTaskOf* association has been added to the *Task* class.
- The proposed case process is added to the metamodel. The *Proposed Case Process* class is associated directly with *Case State*, as (1) associating it with *Phase* would require separate processes for each phase and (2) a case-level process would be required for orchestrating the processes of the individual phases.
- Section 4.3 shows that tasks correspond to S-BPM function states. Therefore, *Proposed Case Process* has an aggregation association with *Task*.

The executed case process is a special view on the activity stream. Therefore, no new elements are introduced to the metamodel for representing the executed case process.

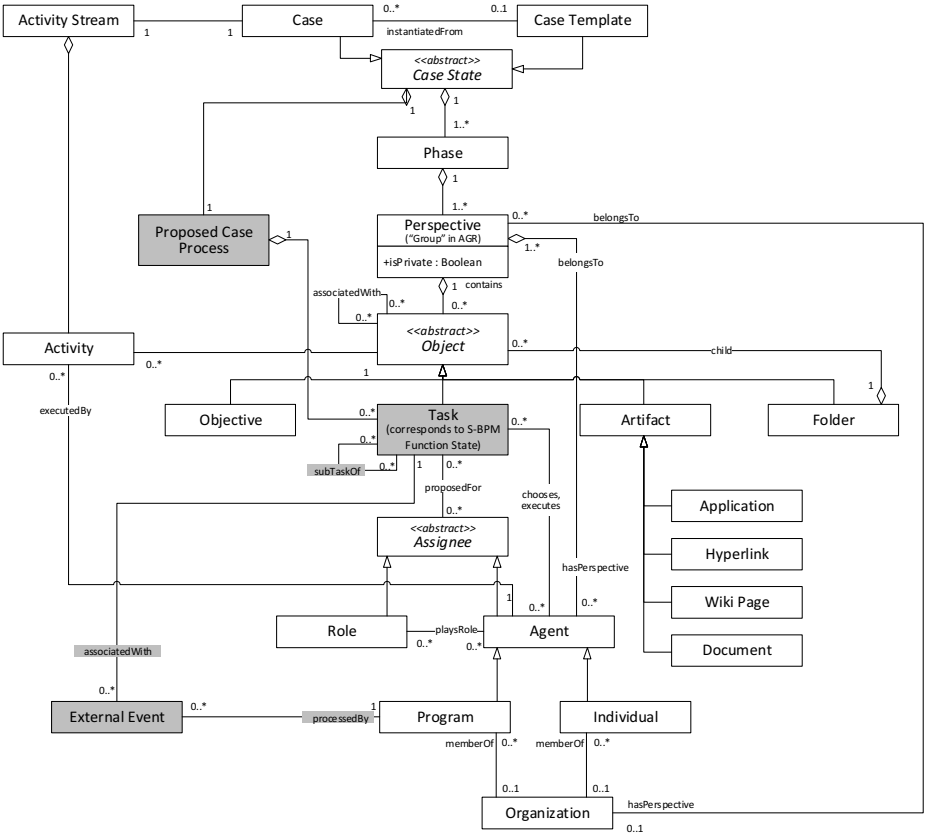


Fig. 4. Extended ACM metamodel

5 Conclusion and Outlook

This paper demonstrates how S-BPM can be applied for defining the temporal-logical dependencies between tasks of knowledge-intensive cross-enterprise processes. S-BPM process models propose an order in which to execute the tasks of the respective case (the proposed case process). In S-ACM, Agents are free to deviate from this proposal at their discretion. With the sequence of executed tasks automatically being recorded in an executed case process, it is possible to check completed cases for compliance with legal requirements or company-specific regulations. Analyzing multiple executed case processes originating from the same case template assists in further improving the proposed case processes embedded in the respective case template.

On the other hand, the proposed approach still has a number of limitations. First, although S-BPM has a graphical notation, this notation needs to be extended in order to address the various aspects of the metamodel. E.g., there currently is no way for visually expressing the relationship between a proposed case process and other case artifacts like documents. The upcoming *Case Management Model And Notation*

(CMMN) of the Object Management Group [11] may give some guidance to the effort of developing such a notation.

Second, a typical operation in ACM is refining tasks by splitting a single task into multiple tasks. While the S-ACM approach provides provisions for this scenario, it is not discussed in this paper due to space constraints. The authors plan to publish an extended version of this publication in the near future. In general the KXBP approach serving as the foundation for S-ACM offers a rich set of concepts that have only been briefly mentioned. In particular, the implications of having multiple perspectives, which may constitute multiple different proposed case processes for different agents, have been covered only briefly due to space constraints.

Finally, a comprehensive set of real-world examples will help analyzing the practicality of the proposed approach. The authors plan to conduct such case studies soon.

References

1. Swenson, K.D. (ed.): Mastering the Unpredictable. How adaptive case management will revolutionize the way that knowledge workers get things done. Meghan-Kiffer Press, Tampa (2010)
2. Layna, F. (ed.): Taming the unpredictable. Real World Adaptive Case Management Future Strategies, London (2011)
3. Kurz, M.: Taming Diversity. A distributed ACM-based approach for cross-enterprise knowledge work. In: Raghavan, V., Hu, X., Liau, C.-J., Treur, J. (eds.) The 2013 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology Workshops, pp. 87–91. IEEE Computer Society (2013)
4. Herrmann, C., Kurz, M.: Adaptive case management. Supporting knowledge intensive processes with IT systems. In: Schmidt, W. (ed.) S-BPM ONE 2011. CCIS, vol. 213, pp. 80–97. Springer, Heidelberg (2011)
5. Shoham, Y., Leyton-Brown, K.: Multiagent systems. Algorithmic, game-theoretic, and logical foundations. Cambridge University Press, Cambridge (2009)
6. Object Management Group: Business Process Model and Notation, BPMN (2011), http://www.omg.org/spec/BPMN/2.0/PDF/
7. Keller, G., Nüttgens, M., Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)" (1992)
8. Dori, D., Goodman, M.: From object-process analysis to object-process design. Annals of Software Engineering 2, 25–50 (1996)
9. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subject-oriented business process management. Springer, Heidelberg (2012)
10. Fleischmann, A., Schmidt, W., Stary, C.: Subject-oriented BPM = Socially Executable BPM. In: Proceedings of the 15th IEEE Conference on Business Informatics, CBI 2013, pp. 399–406. IEEE Computer Society (2013)
11. Object Management Group: Case Management Model and Notation, CMMN (2013), http://www.omg.org/spec/CMMN/1.0/Beta1/