Setup and Maintenance Factors of ACM Systems

Thanh Tran Thi Kim[1], Max J. Pucher[1], Jan Mendling[2], and Christoph Ruhsam[1]

[1] ISIS Papyrus Europe AG
{thanh.tran,max.pucher,christoph.ruhsam}@isis-papyrus.com
[2] Wirtschaftsuniversität Wien, Institute for Information Business
jan.mendling@wu.ac.at

Abstract. Adaptive Case Management (ACM) is information technology for the secure and transparent management of structured and unstructured business processes, consisting of data, content, related work tasks and rules executed towards well-defined process goals. Thus, it goes beyond combining benefits of workflow diagrams with ad-hoc task mechanisms. One of the notorious weaknesses of classical workflow technology is the experts' effort for getting a sufficiently complete specification of the process to create an executable which typically takes several months. In contrast, ACM provides goal-oriented mechanisms to enable performers to define and execute work tasks ad-hoc. In this paper, based on the definition of the ACM concepts, we analyze which setup steps have to be conducted for an ACM system in a typical scenario from the service industry. Our contribution is an identification of major factors that influence the setup initiated by experts and the maintenance performed by business users.

Keywords: Adaptive Case Management, Business Process Management, ISIS Papyrus ACM.

1 Introduction

Adaptive Case Management (ACM) is information technology for the secure and transparent management of structured and unstructured business processes representing work tasks linked to process goals with data and content [1, 2]. ACM provides mechanisms to enable performers to define and execute work tasks ad-hoc without preliminary process analysis and design. The tasks are always linked to at least one goal with completion rules or need to achieve a customer outcome and are monitored through operational targets. In this way, it embraces flexibility requirements such as variability, adaptation and evolution [3] which classical workflow systems often do not support to the full extent [4].

One of the weaknesses of classical workflow technology is the considerable effort of analyzing a process for creating a workflow implementation. For instance, Herbst and Karagiannis observe that the acquisition of workflow knowledge typically consumes three times more than the actual implementation [5]. While there are notable benefits reported of implementing a process [6, 7], the setup costs often become a roadblock for supporting a process with a dedicated workflow implementation.

Y.T. Demey and H. Panetto (Eds.): OTM 2013 Workshops, LNCS 8186, pp. 172–177, 2013.

In this paper, we aim to conceptually investigate how the presumed benefits of ACM in terms of reduced setup effort can be substantiated. The remainder of this paper is structured as follows. First, we analyze the setup and maintenance factors of an ACM system (ACMS). Then, we illustrate the implementation of these factors and point out the adaptability of ACMS in the context of service contract management. Our contribution is an identification of major factors that influence the setup initiated by experts and maintenance performed by knowledge workers (KWs).

2 ACM Setup and Maintenance Factors

ACM methodology takes advantages of the process definition features of BPM to cover specific process fragments, which must be executed in a specific way. This enables ACM to utilize predefined process models like in BPM as a guideline for users but not necessary force them to follow the existing steps from the models unless this is required. However, to deal with unpredictable events happening while working with a predefined process, ACM allows users to create and modify process models at runtime (based on access rights) or create their own step by step performance. Once the well-defined goals are reached the process instance can be converted to an abstract (without the instance data) goal template. Therefore, it eliminates the need for fully sketching out all details of the process beforehand.

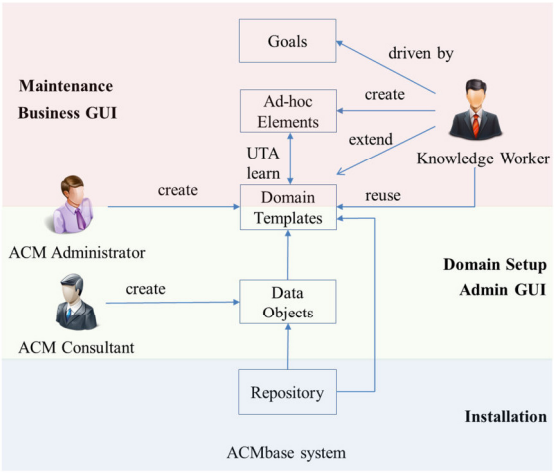

Fig. 1. The ACM environment

Figure 1 depicts the overview of the setup and maintenance of ACM. A process implemented by ACM requires a set of explicitly defined goals. The definition needs access to data for the rule definitions. The rules are performed in natural language and supported with interactive syntax guidance and data attribute validation. Goals can be linked within a case or more importantly also across cases. There are several types of goals: process goals, operational targets, and customer outcomes on the case level and

statistically calculated key performance indicators. Process goals are rule definitions using process data or checklists. Operational targets can be financial or service level targets that are monitored at the case/process level and not just at the department or capability level. That allows a drill down analysis when targets are missed. The factors relating to an adaptive process are goals (business view), outcomes (customer view), skill or resources (capability view), work (task types, dependencies, checklists), data (forms and silo interface view), rules (for data and content, resources, work), and content (inbound, outbound, social, email, rules).

In implementation, the ACMbase system is installed by IT people and can be applied in many business domains. Depending on each business domain, specific data objects and templates are created by ACM consultants and Administrators (Admins in short) in the setup phase. The maintenance is operated by KWs who handle instances and ad hoc events in a particular business case. The performance of KWs is driven by goals which are expected when a case closes. The setup and maintenance factors of ACM are investigated with the ISIS Papyrus ACM system (ACMS). The ACMS is illustrated in the context of service contract management in which the rigidity of BPM and the flexibility of ACM are both applied simultaneously.

The setup an ACM is operated in the GUI for Admins. Admins are assumed to have the knowledge of business operation in service contract management and is able to administrate the ACMS. This is similar to other management systems where administrators need to have the knowledge of the system applied in a specific business domain, e.g. database management systems.

The setup deals with the preparation of the base ACM system for use in a specific business application in terms of the five elements of ACM: content, goals, cases, data entities and GUIs. The setup is operated by Admins with the support from consultants who have knowledge about ACM. Admins construct the templates from classes: data entity, document (i.e., content), goals, rules, reusable processes, participants, and the case itself holding all these elements. Without creating code, Admins configure the setup with Admin GUIs based on the existing foundation classes in the ACMS. Thus, comparing with a BPM setup, the ACM setup needs no technical support from IT experts.

Figure 2 is a mash-up screenshot illustrating the implementation of the setup and maintenance factors of the Papyrus ACMS. Within the limit of a short paper, we represent the main interfaces illustrating the flexibility and ease of the ACMS in setup and maintenance. As seen in Figure 2, Part 1 is the tree of objects which are setup by Admins without coding. The Admins work on the Admin GUI to create business objects or templates. Depending on the business application domain, in this case service contract management, the Admins create the suitable templates for KWs to operate their daily business.

Data entities are the essential elements in a data centric system like ACMS. Customer and contract data objects are created by Admins and ACM consultants. Another key element of ACMS is goals comprised of rules as conditions to decide whether the goals are achieved. Rules are defined in Papyrus Natural Language Rule [9] within the Papyrus ACMS. In service contract management, there are several goals such as Service Contract Acquisition, Service Execution, Contract Audition, etc. (see Part 1).

Each main goal can have sub-goals, for instance, the Service Contract Acquisition has sub-goals which are Contract Composition, Contract Negotiation, and Contract Approval.

Each sub-goal can refer to pre-defined processes (i.e. sub-processes) which are executed to achieve the goal. This element enables ACM to operate the BPM-like processes. Although sub-processes are pre-defined to provide work support for the KWs, they can be edited on the execution level (instances) by KWs. However, it is necessary to follow rigid steps in the approval contract process. In this case, the pre-defined sub-process called Approval Contract is operated rigidly.

Case templates contain all the necessary elements to achieve their goals. However, a case in the ACM is driven by goals. Therefore, Admins only define the data objects containing the goals of the case. Based on the templates created by Admins, the KW performs their daily works in business GUI. However using the templates is not obligate for KWs. KWs are assumed to have capability to handle the unpredictable event in their daily business by their own way. The templates created by Admins are used as guidelines for KWs and they can choose or create by themselves the instances based on the current situation they are working on.

When the setup is accomplished, KWs are able to use the system with the business GUI of the ACM system as seen partly in the right parts of Figure 2. We demonstrate how the system supports KWs with the flexibility under the influence of content and goal orientation.

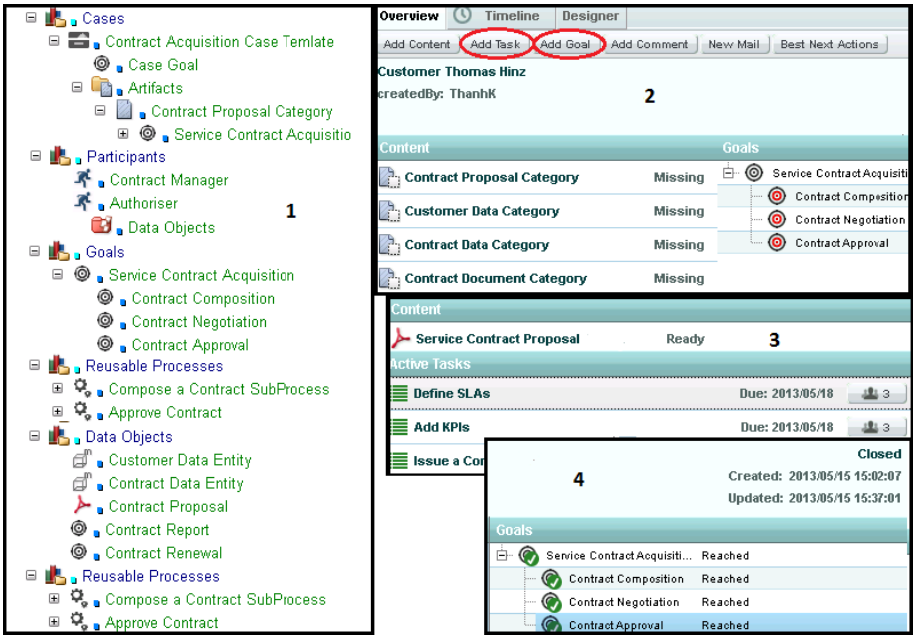

Fig. 2. The implementation of ACM setup and maintenance factors

Part 2 of Figure 2 shows the overview of a case for customer Thomas Hinz GmbH based on the case templates defined by Admins. The case is started with the main goal Service Contract Acquisition. The content related to the case is still missing and needs to be uploaded as Contract Proposal to start the case. In other words, the goal is triggered by the uploading data Contract Proposal. Note that the buttons Add Task and Add Goal enable KWs to add new instances on the fly without the use of templates.

Part 3 depicts the uploaded Contract Proposal and three associated tasks become available for KWs. The KWs choose the task Define SLAs to start compose the service contract definition and the condition contained in the contract document. Since the SLAs and KPIs are completed, the contract is issued. Thus the goal Contract Composition is reached as seen in Part 4. The goal Contract Negotiation does initially not contain any tasks defined by Admins. The KW can add Tasks depending on the current business situation by using the function Add Task as shown in Part 2. Although ACM supports the KW by adding tasks on the fly, the KW cannot edit the predefined processes which are important to meet compliance regulations. The goal Contract Approval contains the sub-process Approve Contract which is essential in service contract management. Therefore, the KW must operate this sub-process in the way defined by Admins. This represents the rigidity of BPM applied in ACM.

When all the goals are reached, the main goal Service Contract Acquisition is reached and the case is closed as seen in Part 4. The demonstration of ACMS GUI shows that the system supports the flexibility for KWs to handle the unpredictable tasks in the practical situation. Moreover, with the goal orientation, the system is reliable to satisfy the legal situation.

The maintenance of the system happens during its operating time and proclaims the adaptability of ACMS. The reusable processes category contains the sub-processes templates. When a process is created by KWs and used frequently, it can be added to the reusable processes category by Admins. Moreover, the system supports KWs by suggesting the most proper next step. This is performed by the User Trained Agent (UTA) which examines the data pattern in a certain KW context to suggest future steps based on pattern matching. Thus, the system improves itself by learning from KWs and their activities which lead to successful closing of goals.

Table 1. Setup and maintenance factors

Factors	Setup	Maintenance	Learning
Goals	Template (Tmp)	Instance (Ins)	Tmp derived from Ins
Data objects	Data structure	Data value	
Reusable Processes	Tmp	Ins	Tmp derived from Ins
Cases	Tmp	Ins	Tmp derived from Ins
GUI	Admin GUI	Business GUI	
Participants	Admin	KW	Admin

Table 1 contains the summary of setup and maintenance factors. The minimum effort for setting up a process in an ACM system requires the definition of goals and essential data objects. This is in contrast the factors that drive the effort of a classical

workflow implementation. The work by Aysolmaz et al. investigates workflow setup effort and finds that control flow complexity and number of different outputs of the activities in the process model are the significant factors in this context [10]. This means that processes with high control flow complexity and a high variation in outputs might potentially be much easier set up using ACM systems. This hypothesis requires an empirical approach in future research.

3 Conclusion

In this paper we discuss the effort of setting up, maintaining and learning a process using an ACM system. Based on a typical case of service contract management we investigate the steps of setting up the system and compare it to the effort of setting up a classical workflow. Our contribution is an identification of major factors that influence the process creation and reuse. We aim to investigate setup effort further in future research.

References

1. Pucher, M.: The Strategic Business Benefits of Adaptive Case Management. In: Fischer, L. (ed.) How Knowledge Workers Get Things Done - Real-World Adaptive Case Management. Future Strategies Inc. (2012)
2. Pucher, M.: Adaptive Case Management - ISIS Papyrus and ACM. ISIS Papyrus, http://acmisis.wordpress.com/what-is-adaptive-case-management-acm/
3. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies, pp. I-XVIII, 1–515. Springer (2012) ISBN 978-3-642-30408-8
4. Reijers, H.A.: Workflow Flexibility: The Forlorn Promise. In: 15th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2006), pp. 271–272 (2006) ISBN 0-7695-2623-3
5. Herbst, J., Karagiannis, D.: Integrating machine learning and workflow management to support acquisition and adaptation of workflow models. In: Proceedings of the 9th International Workshop on Database and Expert Systems Applications, p. 745 (1998) 0-8186-8353-8
6. Reijers, H.A., Van Der Aalst, W.M.P.: The effectiveness of workflow management systems. Predictions and lessons learned. Int. J. Inf. Manag. 25(0268-4012), 458–472 (2005)
7. Gruber, H., Huemer, C.: Profitability Analysis of Workflow Management Systems. In: Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing, pp. 233–238 (2009) 978-0-7695-3755-9
8. Pucher, M.: The Elements of Adaptive Case Management. In: Mastering the Unpredictable, pp. 89–135. Meghan-Kiffer Press, Florida (2010)
9. ISIS Papyrus: WebRepository Adaptive Case Management Framework - Developer and Administrator Guide. In: ISIS Papyrus Technical Documentation (2013)
10. Aysolmaz, B., İren, D., Demirörs, O.: An Effort Prediction Model based on BPM Measures for Process Automation. In: Nurcan, S., Proper, H.A., Soffer, P., Krogstie, J., Schmidt, R., Halpin, T., Bider, I. (eds.) BPMDS 2013 and EMMSAD 2013. LNBIP, vol. 147, pp. 154–167. Springer, Heidelberg (2013)