Tasks and Assignments in Case Management Models

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 100 (2016) 156 – 163

Procedia Computer Science

Conference on ENTERprise Information Systems / International Conference on Project MANagement / Conference on Health and Social Care Information Systems and Technologies, CENTERIS / ProjMAN / HCist 2016, October 5-7, 2016

Tasks and assignments in case management models

Giorgio Bruno*

Politecnico di Torino, Torino, Italy

Abstract

The recent standard CMMN presents an approach to deal with situations, referred to as cases, that call for more flexibility than that offered by traditional control-flow based approaches. The information (documents and data) needed by a case is collected in a hierarchical structure called case file. The advancement of a case is based on the execution of the tasks shown in the case process model; the determination of the tasks to be carried out takes place through human decisions or events (e.g., those related to changes in the case file). However, the standard does not explain how the actual assignments of tasks to the participants in the case are generated. This paper tries to fill the gap by presenting a more detailed treatment of assignments, and by proposing an extension to the CMMN notation. This extension enables the features of assignments to be inferred from the case process model; moreover, it allows for the representation of the dataflow between tasks. There are two major benefits: one is to show the types of the inputs of the assignments, and the other is the possibility of determining the assignees of tasks as a function of the input entities. The notation is illustrated with the help of an example that concerns the handling of the papers submitted to a conference.

Peer-review under responsibility of the organizing committee of CENTERIS 2016

Keywords: case management; case process models; human tasks; assignments; dataflow.

1. Introduction

In the realm of Business Process Management, the recent publication of the CMMN standard[5] is a clear sign of the interest that is growing around the notion of case. In this context, a case is a situation that requires a

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: giorgio.bruno.@polito.it

Peer-review under responsibility of the organizing committee of CENTERIS 2016
doi:10.1016/j.procs.2016.09.135

customized treatment and the customization is carried out by the people involved in the case. On the contrary, there are many other situations in which a standardized treatment (also called routine) fits the needs of the business.

Routines provide an efficient way of handling repetitive situations: they orchestrate work by distributing tasks (i.e., work units) to automatic services or to (human) participants through their work lists. Participants are seen as resources to be entrusted with activities that are not automatable; their visibility on the situations in which they are involved is narrow (as pointed out in case handling[18]) and limited to the assignments contained in their work lists. The choice of the paths in the process is carried out automatically on the basis of predefined rules. The standard notation for routines is BPMN[1].

On the contrary, in the treatment of a case the participants are entitled to decide the activities to be carried out on the basis of the circumstances related to the current situation. The notion of case is well known in the medical and legal realms. In the business domain, the examples that are usually presented address the processing of complaints, loan requests, and various kinds of applications.

A case encompasses three categories of elements: the information that matters, the tasks that may be carried out, and the participants (also called case workers) involved.

CMMN assumes that the information (documents and data) needed by a case is collected in a hierarchical structure called case file; the access to the components of the case file (called case file items) is granted to the participants in the case on the basis of the roles they play.

CMMN provides a notation to graphically define the process governing the evolution of a case. There is no rigid control flow as in routines because the execution of tasks can be decided by the participants themselves as well as by the process.

CMMN stresses the notion of plan: the plan of a case encompasses all the tasks that have to be performed. A new task can be added to the plan by the process as a result of an event, such as the completion of a task, the achievement of a milestone, a change in the case file, or a time event. In addition, case workers can add discretionary tasks (called planned tasks) to the plan at run-time; they do so by selecting such tasks from planning tables that may be associated with tasks or stages (a stage is a grouping of tasks and of lower-level stages).

However, the analysis of the CMMN standard reveals that a number of points need to be clarified:

- the correspondence between the plan and the work list of the participants;
- the determination of the assignees of the tasks;
- the definition of the inputs and outputs of the tasks in terms of the case file items affected.

This paper tries to fill the gap by presenting a more detailed treatment of these points on the basis of the notion of assignment.

An assignment is related to a case file and to a task of the case, is directed to an assignee and may refer to one or more case file items (the inputs of the assignment). Assignments form the work lists of case workers and the assignments related to the same case form the case plan.

Assignments have two execution features: the active period and the execution mode. The former specifies when the task must or may be performed; the latter establishes whether the task is mandatory or optional and whether it may be repeated or not.

Assignments are generated in three ways: on the basis of a time constraint, of an event, or of a planning action.

This paper also proposes an extension to the CMMN notation: this extension enables the features of assignments (and the ways they are generated) to be inferred from the case process model.

The notation allows for the representation of the dataflow between tasks. There are two major benefits: one is to show the types of the inputs and outputs of the assignments, and the other is the possibility of determining the assignees of tasks as a function of the input entities.

The representation of the dataflow calls for an information model that defines the case file in terms of entity types, relationships and attributes; a UML class model[16] can fit this purpose.

This paper is organized as follows. Section 2 is about the related work. Section 3 presents an example that concerns the handling of the papers submitted to a conference. This example will be used to explain the approach proposed. Section 4 illustrates the features of assignments and how they are generated; it also provides the process model of the example in order to clarify the analysis of the relationships between tasks and assignments. Section 5 contains the conclusion.

2. Related work

The literature on case management includes a number of recent surveys[6,13,14]. Case management takes several flavors, i.e., Adaptive Case Management[8] (ACM) and Production Case Management (PCM): they can be compared[12] on the basis of eight key performance indicators[7].

In the years, various definitions of those variants have been proposed[12]. On the basis of such definitions it turns out that the major distinction between ACM and PCM is as follows: in PCM the design of the case and its execution take place in two sequential stages, while in ACM they are carried out at the same time by the same persons.

The relationships between case management and social networks as well as task management have also been investigated[15].

The standard CMMN[5] follows the PCM approach; some extensions to make it compliant with ACM features have also been suggested[11].

CMMN has roots in the Guard-Stage-Milestone approach[10] (GSM) and in research on flexible processes.

GSM is data-centric and as such it emphasizes the business entities involved in the process and their life cycles[9]. Compared to the activity-centric approaches such as BMPN[1], data-centric ones facilitate communication among the stakeholders and help them focus on the primary purposes of the business[4]. CMMN considers the case as a whole and then it handles one life cycle; moreover, it stresses human tasks, which are not given particular importance in GSM.

Flexibility means that the sequence of execution of tasks is not rigidly predetermined in the process model but, to a certain extent, it can be decided at run time. The ad-hoc sub-processes provided by BPMN give the maximum freedom in that their tasks may be carried out in any order and may be repeated. Constraints can be defined in Declare[17]: any sequence of task is allowed provided that the constraints are not violated. Flexibility is achieved in CMMN by means of planning tables, through which discretionary tasks can be assigned to suitable participants.

The explicit representation of the dataflow in business process models has also been discussed in previous papers[2,3] of the author.

3. Description of the example

This section presents the case that will be used to illustrate the proposed approach. It concerns the handling of the papers submitted to a conference; for simplicity it is referred to as conference case although only some of the actual issues of a conference are addressed.

As mentioned in the introduction, a case encompasses three categories of elements: the information that matters, the tasks that may be carried out, and the participants involved.

Participants (also called case workers) are indicated by the role that plays: in the conference case there are three roles, Chair, Author and Reviewer. Chair is the case manager role.

The information that matters is made up of information entities whose structure is defined by means of an information model, which is a UML class model. One of the classes represents the primary entity, i.e., the

entity which is generated first and is connected either directly or indirectly to all the other entities. Such an entity is called primary entity and its class is called primary class. The information model of the conference case is shown in Fig.1: Conference is the primary class and Paper and Review are the other information classes. The primary class is denoted by the <<p>> stereotype.

A short description of the case is as follows.

The chair is assumed to provide the basic information on the conference through the primary entity, and then the process is started by the run-time system. The basic information includes the conference title and five dates, d1 to d5: d1 and d2 are the beginning and the end of the submission period, respectively, d3 is the date by which the reviews of the papers must be available, d4 is the date at which the results will be published, and d5 is the deadline for the provision of the final versions of the papers accepted. Therefore, there are four sequential period: the submission period (d1 to d2), the reviewing period (d2 to d3), the assessment period (d3 to d4), and the finalization period (d4 to d5).

During the submission period, authors may submit papers (up to 2); they may also withdraw their papers.

In the reviewing period, the chair will assign each paper to three reviewers. The job of reviewers is to complete their reviews by the due dates, which have been set in their assignments by the chair.

The chair is entitled to make additional assignments in order to get at least three reviews for each paper; this occurs, for example, if some expected reviews are not provided in due time. The chair is to set the due dates so that the deadline (d4) for the publication of the results can be met.

After the chair has decided the outcome of each paper (either accepted or rejected), the authors are notified by email; the final versions of the accepted papers may be provided by d4.

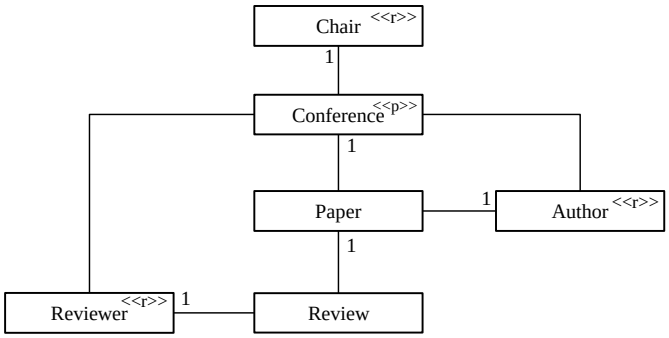

Figure 1. The information model of the conference process

The information model also includes role classes, i.e., the classes representing the participants in the case on the basis of their roles. The role classes are denoted by the <<r>> stereotype. All the participants are connected to the primary entity; this is the reason of the relationships between each role class and the primary class. Such relationships are called participation relationships.

The participants are users of the underlying case management system (CMS): they may join the case over time and then they become participants in the case. A user that is entitled to start a conference case takes the chair role; the users who want to participate in a conference case as authors or reviewers ask the CMS to be

enrolled in the case with those roles. A user can participate with different roles in the same case: for example, a reviewer may also be an author. The details of the enrolment of users in cases are not worked out in this paper.

The relationships between role classes and information ones, such as Author - Paper and Reviewer - Review, usually take the meaning of authorship relationships.

4. Assignments and tasks

This section explains the features of assignments and illustrates how they are produced and how the assignees are determined. It does so with the help of the conference process model shown in Fig. 2.

The process model is made up of three sections, named Submissions, Reviews and Finalization, which correspond to the three sequential phases of the conference case: the first is about the submission of papers, the second deals with the reviews and the third includes the final actions. The basic components are tasks and events: the graphic symbols of tasks and events are the rectangle and the circle, respectively.

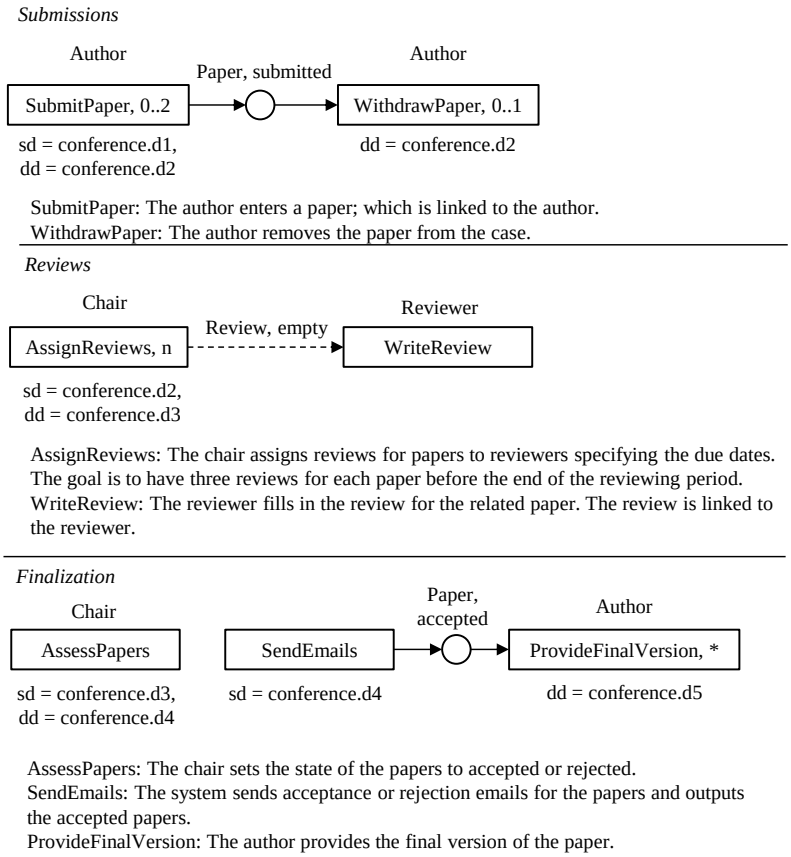

Figure 2: The process model of the example

A human task is accompanied by the role of the participants who can perform it. An automatic task has no role associated with it. Task descriptions may be included in the model.

4.1. Features of assignments

The execution of tasks takes place through assignments. An assignment is related to the primary entity and to a task of the case, is directed to an assignee and may refer to a number of input entities. If there are no input entities the task is meant to be able to retrieve those needed by navigating through the case entities.

The assignments form the work lists of the participants in the process. The participants may look at their work lists and when they click on an assignment they can perform the corresponding task with the help of a graphical interface.

Assignments have two execution features: the active period and the execution mode. The active period determines when the corresponding task can be performed and consists of two dates, the start date and the due date. The execution mode establishes whether the task is mandatory or optional and whether it may be repeated or not.

The execution mode takes one of these three forms: a range (l..h), the character 'n' and the character '*'. The lower limit of the range may be 1(if the task is mandatory) or 0 (if it is optional); the higher limit is an integer number >= l. The character "n" ("*") means that the task is mandatory (optional) and may be repeated a number of times that is not predetermined.

The execution features may be specified in the task attributes and may appear in the process model as follows. The active period is shown by means of a task annotation having the form "sd = date" and "dd = date", where sd introduces the start date and dd the due date, and "date" stands for a reference to a date attribute of a case entity. For example, the active period of task SubmitPaper is the submission period of the conference; therefore the start date and the due date are "d1 and "d2", where d1 and d2 denotes the attributes d1 and d2 of the primary case entity.

The execution mode is shown after the task name and takes one of the above mentioned forms. If it is missing, the implicit execution mode is "1..1"; in this case, the task is mandatory and it is not repeatable.

Assignments are generated in three ways: on the basis of a time constraint, of an event, or of a planning action. These ways and the related execution features can be inferred from the process model: in particular they are inferred from the attributes and the context of tasks as follows.

4.2. Time-based assignments

Time-based assignments take place for tasks having no input links, such as SubmitPaper, AssignReviews, AssessPapers and SendEmails. The start date and the due date are both specified but for automatic tasks, such as SendEmails, where only the start date needs to be indicated.

On the basis of the execution mode, it results that SubmitPaper may be performed in the submission period, while AssignPapers and AssessPapers will be performed in the reviews period and in the finalization one, respectively.

Tasks having no input links are assumed to retrieve their input entities by navigating through the case entities starting from the primary case entity, which is associated with their assignments.

4.3. Event-driven assignments

The completion of an assignment for a certain task may bring about an assignment for another task. This situation is represented by an event symbol which is the output of the first task (the source task) and the input

of the other task (the destination task). The event may carry a payload which is a case entity. The event is called simple if it has no payload; otherwise it is a specific event. In the second case, the event icon has a label containing the type and the name of the event, e.g., “Paper, submitted”.

When an event occurs, the run-time system generates an assignment for the destination task: the start date is the current date and the due date is the one associated with the task. If the event contains a payload, the case entity is linked to the assignment. For example, when task SubmitPaper is completed, the run-time system makes an assignment for task WithdrawPaper: the input entity of the assignment is the paper emitted by task SubmitPaper. If there is no payload the destination task is meant to get its input entities by searching through the case entities.

4.4. Planned assignments

A planned assignment is made during the execution of a task, called planning task. This situation is represented by means of a dashed link (called planning link) from the planning task to the planned one. For example, task AssignPapers is meant to plan a number of assignments for task WriteReview. Each assignment refers to an empty review as shown by the label of the link. The active periods of the assignments are decided by the performer of the planning task. If the planning link has no label, the planned task is meant to get its input entities by searching through the case entities.

4.5. The determination of assignees

While in planned assignments, the assignees are decided by the performers of planning tasks, for event-driven assignments and time-based ones they are determined by the run-time system on the basis of assignment rules.

Two such rules, called role assignment and payload assignment, are suitable for the conference case. Role assignment is applied to time-based tasks and event-driven tasks if the events are simple; payload assignment is applied to event-driven tasks if the events are specific. With the first rule, the assignees are all the participants who play the role of the task and participate in the case; with the second rule, the assignees are the participants who play the role of the task and are connected with the input entity.

Therefore, the assignees of SubmitPaper are all the authors participating in the conference, the assignees of WithdrawPaper and of ProvideFinalVersion are the authors of the input papers.

5. Conclusion

CMMN provides a notation to graphically define the process governing the evolution of a case. There is no rigid control flow as in routines in that the execution of tasks can be determined by the participants through planning tables or by the process through various kinds of events, such as the completion of a task or a change in the information base of the case.

This paper has elaborated on the notion of assignment, which has been left aside in CMMN. The execution of tasks takes place through assignments, which form the work lists of the participants in the case.

Assignments have a number of features that can be inferred from the case process model: for this reason, an extension to the CMMN notation has been proposed. The notation allows for the representation of the dataflow between tasks. There are two major benefits: one is to show the types of the inputs of the assignments, and the other is the possibility of determining the assignees of tasks as a function of the input entities.

The representation of the dataflow calls for an information model that defines the case file in terms of entity types, relationships and attributes.

Further work is developing in two directions. One direction is concerned with the association of pre-conditions and post-conditions with tasks so as to express the constraints and the effects of the tasks. The other direction is to elaborate on the interactions between planners and assignees of planned tasks in order to provide greater flexibility: for example, the assignee of a planned task, such as WriteReview, might decline the assignment and if so the performer of the planning task AssignReviews will be informed so that he or she can take appropriate actions.

References

1. BPMN, Business Process Model and Notation, V.2.0.2. Retrieved February 4, 2016, from http://www.omg.org/spec/BPMN/2.0.2/
2. Bruno, G.: A data-flow language for business process models. Procedia Technology, vol. 16, pp. 128–137 (2014)
3. Bruno, G.: Data flow and human tasks in business process models. Procedia Computer Science, vol. 64, pp. 379–386 (2015)
4. Chao, T., et al.: Artifact-based transformation of IBM Global Financing. LNCS, vol. 5701, pp. 261–277. Springer, Heidelberg (2009)
5. CMMN. Case Management Model and Notation, V.1.0. Retrieved February 4, 2016, from http://www.omg.org/spec/CMMN
6. de Man, H.: Case management: a review of modeling approaches. BPTrends, January (2009).
7. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. Journal on Data Semantics, 4, 29–57 (2015)
8. Hauder, M., Pigat, S., Matthes, F.: Research challenges in adaptive case management: a literature review. In IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), pp. 98–107 (2014)
9. Hull, R.: Artifact-centric business process models: brief survey of research results and challenges. LNCS, vol. 5332, pp. 1152–1163. Heidelberg: Springer (2008)
10. Hull, R., et al.: Introducing the Guard-Stage-Milestone approach for specifying business entity lifecycles. LNCS, vol. 6551, pp. 1–24. Springer, Heidelberg (2011)
11. Kurz, M., Fleischmann, A., Schmidt, W., Lederer, M.: Leveraging CMMN for ACM. In 7th International Conference on Subject-Oriented Business Process Management. ACM New York (2015)
12. Marin, M., Hauder, M.: Case Management: a data set of definitions. Cornell University Library: arXiv:1507.04004v1 (2015)
13. Marin, M., Hauder, M., Matthes, F.: Case management: an evaluation of existing approaches for knowledge-intensive processes. In 4th International Workshop on Adaptive Case Management and other non-workflow approaches to BPM. Springer, Heidelberg (2015)
14. Marin, M., Hull, R., Vaculín, R.: Data centric BPM and the emerging case management standard: a short survey. LNBIP, vol. 132, pp 24–30. Springer, Heidelberg (2013)
15. Motahari-Nezhad, H. R., Swenson, K. D.: Adaptive case management: overview and research challenges. In IEEE 15th Conference on Business Informatics, pp. 264–269 (2013).
16. UML. Unified Modeling Language, V.2.4.1. Retrieved February 4, 2016, from http://www.omg.org/spec/UML/2.4.1/
17. van der Aalst, W. M. P., Pesic, M., Schonenberg, H.: Declarative workflows: balancing between flexibility and support. Computer Science - Research and Development, 23 (2), 99–113 (2009).
18. van der Aalst, W.M.P., Weske, M., Grünbauer, D.: Case handling: a new paradigm for business process support. Data & Knowledge Engineering, vol. 53 (2), pp 129–162 (2005)