USING CONTEXT INFORMATION AND CMMN TO MODEL KNOWLEDGE-INTENSIVE BUSINESS PROCESSES

Siniša Nešković*, Kathrin Kirchner**
*Faculty of Organizational Sciences, University of Belgrade, Jove Ilića 154, Belgrade, Serbia
**Berlin School of Economics and Law, Alt-Friedrichsfelde 60, Berlin, Germany
sinisa.neskovic@fon.bg.ac.rs, kathrin.kirchner@hwr-berlin.de

***Abstract*—Knowledge-intensive business processes are characterized by flexibility and dynamism. Traditional business process modeling languages like UML Activity Diagrams and BPMN are notoriously inadequate to model such type of processes due to their rigidness. In 2014, the OMG standard CMMN was introduced to support flexible processes. This paper discusses the main benefits of CMMN over BPMN. Furthermore, we investigate how context information of process instances can be used in CMMN to allow runtime flexibility during execution. The proposed technique is illustrated by an example from the healthcare domain.**

I. INTRODUCTION

Business Process Management (BPM) often deals with well-structured routine processes that can be automated. But in last years, the number of employees that intensively use non-routine analytic and interactive tasks increased [1]. Knowledge workers like managers, lawyers and doctors need experience with details of the situation to carry out their work processes. While software is provided to support routine tasks, this is less the case for knowledge work [2]. Knowledge-intensive processes have a need of human expertise in order to be completed. They integrate data in the execution of processes and require substantial amount of flexibility at run-time [3]. Thus, activities might require a different approach during each process execution. Case-based representations of knowledge-intensive processes provide a higher flexibility for knowledge workers. The central concept for case handling is the case and not the activities and their sequence. A case can be seen as a product which is manufactured. Examples of cases are the evaluation of a job application, the examination of a patient or the ruling for an insurance claim. The state and structure of any case is based on a collection of data objects [4].

Standards for business process modeling, e.g., UML Acitivity Diagrams [5] or BPMN 2.0 [6], usually abstract from flexibility issues [7]. BPMN is used for modeling well-structured processes. However, in BPMN 2.0, ad-hoc processes can be used to model unstructured models. Elements in an ad-hoc sub-process can be executed in any order, executed several times or are even omitted. No rules are defined for the task execution in the sub-process, so the person who executes the process is in charge to make the decision.

Case Management Model and Notation (CMMN) is an OMG modeling standard for case modeling introduced in 2014 [8]. Although a new version 1.1 in December 2015 increased the clarity of the specification and corrected features to improve the implementability and adoption of CMMN, the language still needs to be adapted by process modelers and prove that it is capable to support different knowledge-intensive processes with lots of flexibility.

One important aspect of flexible business processes is the context in which a process is executed. Many practical situations exist where the workflow of activities depends heavily on surroundings of the particular process instance. For example, in healthcare domain patients are treated depending on their particular health conditions, and routine fixed treatment cannot be prescribed in advance.

The context plays an important role in several application areas such as natural languages, artificial intelligence, knowledge management, and web systems engineering, ubiquitous computing and Internet of things computing paradigm. In the domain of business process modelling, context awareness is a relatively new field of research. Context aware self-adaptable applications change their behavior according to changes in their surroundings [9]. CMMN has no direct support to model context aware processes.

However, certain mechanisms exist which can be used to model a context based flexible process. In this paper we propose a technique how this can be achieved.

Our paper is structured as follows: Section II introduces CMMN as a means to support flexible processes. In Section III, we give a motivational example from healthcare. Section IV provides our idea to integrate contextual information into CMMN models. We discuss related work in section V and summarize our findings in section VI.

II. CMMN AS A MEANS TO SUPPORT FLEXIBLE PROCESSES

Case management requires models that can express the flexibility of a knowledge worker. This can be covered by CMMN [8]. CMMN provides less symbols than BPMN, and might be therefore easier to learn. Since CMMN is a relatively new standard, a brief introduction is given. CMNN is a graphical language and its basic notational symbols are shown in Figure 1.

A Case in CMMN represents a flexible business process, which has two main distinct phases: the design-

time phase and the run-time phase. In the design-time phase, business analysts define a so called Case model consisting of two types of tasks: tasks which are always part of pre-defined segments in the Case model (represented by rounded rectangular), and "discretionary" (i.e. optional, marked as rectangular with a dotted line) tasks which are available to the Caseworker, to be performed in addition, to his/her discretion. In the run-time phase, Caseworkers execute the plan by performing tasks as planned, while the plan may dynamically change, as discretionary tasks can be included by the Caseworker to the plan of the Case instance in run-time.

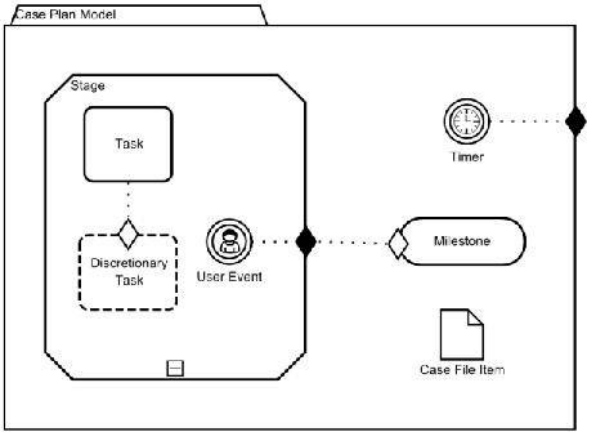

Figure 1. Basic CMMN 1.0 notational symbols

A Stage (rectangles with beveled edges) groups tasks and can be considered as episodes of a case with same pre- or postconditions. Stages can be also planned in parallel.

Sentries (diamond shapes) define criteria as pre- or postconditions to enter or exit tasks or stages. Entry conditions are represented by the white diamond, whereas exit conditions are designated by black diamond. Conditions are defined by combinations of events and/or Boolean expressions.

Milestones (rounded rectangles) describe states during process execution. Thus, a milestone describes an achievable target, defined to enable evaluation of the progress of a case.

Other important symbols are events (circles) that can happen during a course of a case. Events can trigger the activation and termination of stages or the achievements of milestones. Every model is described as case plan model, that implicitly also describes all necessary data. In order to make the data more explicit, a so called `CaseFileItem`, represented as document symbol, can be used.

Additional conditions for the execution can be described in a planning table. Connections are used to express dependencies but no sequential flows. Connections are optional visual elements and do not have execution semantics.

III. MOTIVATIONAL EXAMPLE

In this paper, we use an example for a CMMN model in a health care scenario [10]. During a patient treatment, or especially in an emergency scenario, medical doctors and nurses need to be free to react and make decisions based on the health state of the patient. Deviations in a treatment process are therefore frequent. Figure 2 shows a CMMN model for the evaluation of a living liver donor.

A person that is considering donating a part of her liver is first medically evaluated to ensure that such a surgery can be carried out. Each evaluation case needs to be started with performing the required task of *Draw blood for lab tests and perform physical examination*. Afterwards, *Perform psychological evaluation* must be performed before the milestone *Initial examination performed* is reached. Thus, the execution of the two stages *Med/Tech investigations* and *Mandatory referrals* is enabled.

Examinations that are performed only sometimes according to medical requirements (depending on the decision of the medical staff) are modeled as discretionary tasks, e.g., *Perform lung function test*. The tasks within a stage can be executed in any sequence.

The stages *Med/Tech Investigations* and *Mandatory referrals* are prerequisites for the milestone *Results available.* If the milestone is reached, the task *Analyze Results* can be executed. According to this analysis, further investigations might be conducted in the stage *Further Investigations*. In this stage, all tasks are discretionary and can be planned according to their need during case execution for a specific patient. The `CaseFileItems` *Patient Record* and *Patient Analysis Result* contain important information for the decision about executing tasks. A potential donor can be considered as non-suitable at every stage of the evaluation, as shown in the model by the described event.

IV. EMBEDDING CONTEXTUAL INFORMATION INTO CMMN MODELS

Business processes are almost never executed isolated, but instead have an interaction with other processes or the external environment outside of their business system. In other words, processes usually execute within a context. Hence, modeling flexible processes requires modeling of the context as well as expressing how contextual information influence the process execution.

In the literature there exist many different approaches for context modeling [11] and how context-aware and self-adaptable applications can be developed [9]. However, these approaches are not possible directly to employ since CMMN is strictly defined standardized modelling language.

However, in CMMN exist several mechanisms for expressing flexibility, which can be used for modeling context aware processes. Here we propose an approach which is based on concepts of `CaseFileItems` and `ApplicabilityRules` of `PlanningTables`.

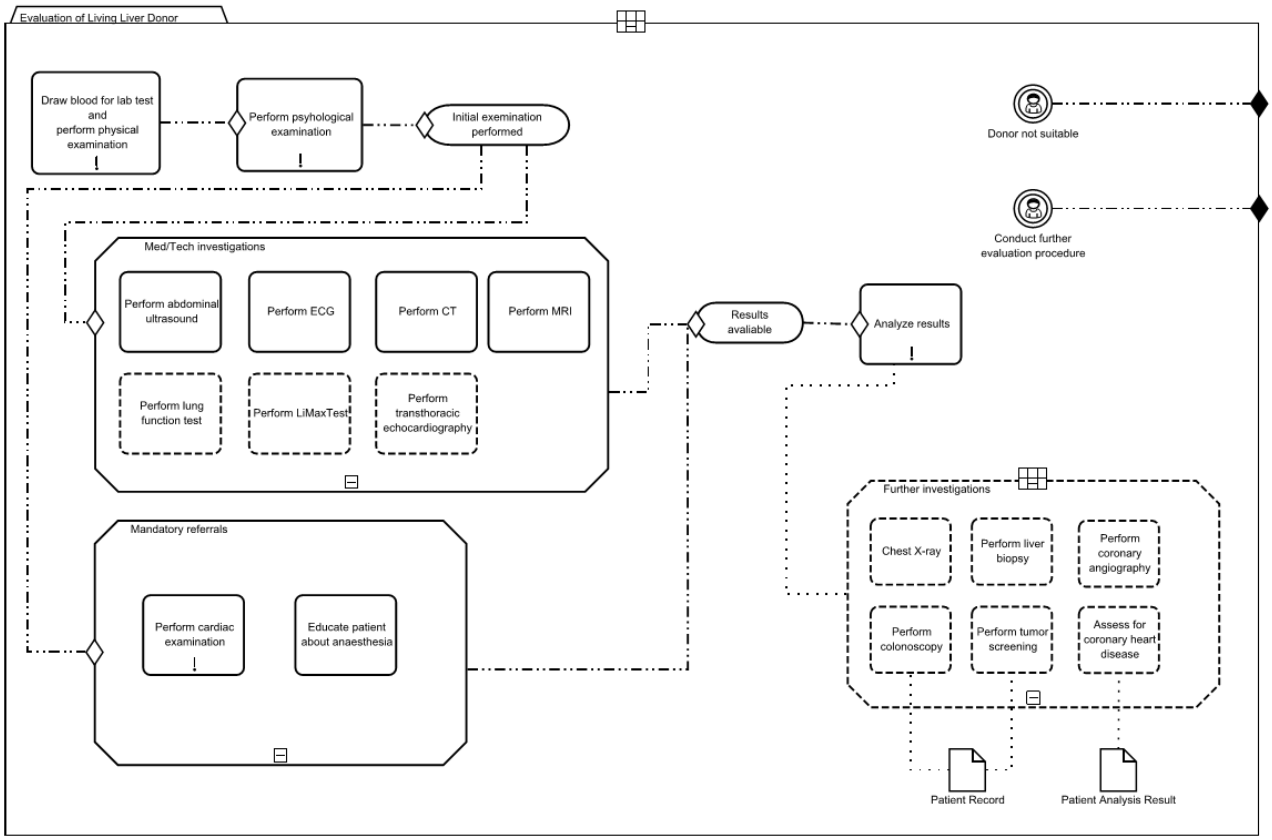

Figure 2. CMMN example for a living liver donor evaluation (adapted from [10])

`CaseFileItem` is essentially a container of information (i.e. document), which is mainly used in CMMN to represent inputs and outputs of tasks. But, it can be also used to represent the context of a whole process or particular process stage. According to the CMMN standard, information in the `CaseFileItems` are also used for evaluating `Expressions` that occur throughout the process model. This allows that contextual information stored in a `CaseFileItem` can be directly used in declarative logic for changing behavior of the process.

CMMN does not prescribe the modelling language for data structures and expressions. Here we propose to use UML class diagrams [12] for modelling `CaseFileItem` structures, and Object Constraint Language (OCL) as a language for modelling expressions [13], [14].

In Figure 3 a simplified example of patient records is shown which can be used as a context for the stage *Further investigations* of the living liver donor evaluation process from Figure 2.

In addition to the basic personal data about a patient (class *Patient Record*), the context also includes analysis results (class *Patient Analysis Result*), each consisting of concrete values for various analysis parameters (class *Analysis Parameter*).

How contextual information influence the process execution is expressed using `Planning tables` and `Applicability rules`. A `Planning Table` is used to define a scope of the context, i.e. which parts (i.e. process stages and tasks) of the case model are dependent on the context. Stages and tasks which have a planning table are decorated in the CMMN graphical notation with a special table symbol (⊞).

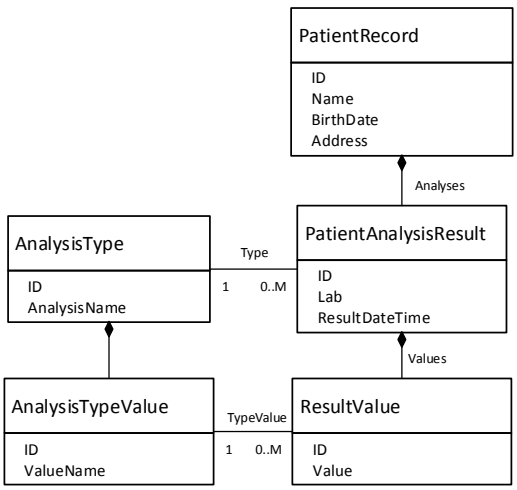

Figure 3. Patient Record - Context model for patients

Each table consists of so called Discretionary Items, which represent tasks or other (sub)stages which are context dependent. Each Discretionary Item can be associated with an Applicability Rule. Rules are used to specify whether a specific Discretionary Item is available (i.e. allowed to perform) in the current moment, based on conditions that are evaluated over contextual information in the corresponding Case File Item.

As an simplified illustration, in Table 1 is given a part of the planning table for the stage Further investigations, which consist of three Discretionary Items (other rules are omitted due to space limitation).

Discretionary Item	**Applicability Rule**
Perform colonoscopy	Over50
Perform tumor screening	Over50
Assess for coronary heart diseases	HighBloodPress

Table 1. Planning Table for *Further Investigation* Stage

The first two `Discretionary Items` are tasks `Perform colonoscopy` and `Perform tumor screening`. These two tasks should be available to the caseworker for execution, when a particular patient case comes to the stage for further medical investigation, only if rule `Over50` is evaluated to true. In other words, these two types of further investigations of a patient should be done only when he/she is older than 50. The rule is specified over context using OCL syntax as follows:

```
Rule Over50:
context: PatientRecord
inv: self.Age >= 50
```

The rule is based on a very simple OCL expression which examine the value of age attribute of patient record.

The third `Discretionary Item` is the task `Assess for coronary heart diseases` which should be available only if rule `HighBloodPressure` is evaluated to true. This rule is based on somewhat more complex OCL expression, which is specified as follows:

```
Rule HighBloodPressure:
context: PatientAnalysisResult
inv: self.Values()->exist(v|
     v.Parameter.Name = ”systolic” and
     v.Value> 140)
```

This rule expression evaluates to true if there exist a patient analysis containing parameter for systolic blood pressure with value bigger than 140. In other words, `Assess for coronary heart diseases` of a patient should be done only when he/she have a higher blood pressure.

V. RELATED WORK

Process models and process-aware information systems need to be configurable, be able to deal with exceptions, allow for changing the execution of cases during runtime, but also support the evolution of processes over time [15].

In the literature, several approaches exist already to support the modeling and execution of flexible processes. Declarative Process Modeling is activity-centered [16]. Constraints define allowed behavior. During runtime, only allowed activities are presented to the knowledge worker, and he decides about the next executed activity. Provop [17] allows the configuration of process variants by applying a set of defined change operations to a reference process model. Configurable Event Process Chains similarly allow the explicit specification of configurations [18].

Regarding execution of flexible processes, Proclets allow the division of a process into parts [19]. These parts can be later executed in a sequence or interactively. ADEPT allows to make changes during the execution time of a process [20].

The research into context-awareness is a well-known research area in computer science. Context-awareness is related to ability of a system to react to the changes in its environment. Many researchers have proposed definitions of context and explanations of different aspects of context-aware computing, according their specific needs [21], [9], [22]. One of the most accepted definition is given in [9]: "*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*"

There exist many different context modelling approaches, which are based on the data model and structures used for representing and exchanging contextual information. Strang and Linnhoff-Popien [11] categorized the most relevant context modelling approaches into six main categories:

- *Key-value*: context information are modeled as key-value pairs in different formats such as text files or binary files
- *Markup schemes*: Markup schemas such as XML are used
- *Graphical modeling*: Various data models such as UML class diagrams or ER data models are used
- *Object based modelling*: Objects are used to represent contextual data.
- *Logic based modelling*: Logical facts, expressions and rules are used to model contextual data.
- *Ontology based modelling*: Ontologies and semantic technologies are used.

Since CMMN is open regarding representation language used to model Case File Items, many of the above categories can be used. The approach used in this paper belongs to the graphical approach.

Context-awareness systems typically should support acquisition, representation, delivery, and reaction according to [23]. From this point of view, except for reaction, CMMN has no adequate concepts to support this typical functions. Developers must rely on some external mechanism in order to support these functions.

In addition, according to [19], there are three main abstract levels for modeling and building context-aware applications:

1. No application-level context model: All needed logic needed to perform functions such as context acquisition, pre-processing, storing, and reasoning must be explicitly modeled within the application development.
2. Implicit context model: Applications can be modeled and built using existing libraries, frameworks, and toolkits to perform contextual functions.
3. Explicit context model: Applications can be modeled and built using a separate context management infrastructure or middleware solutions. Therefore, context-aware functions are done outside the application boundaries. Context management and application are clearly separated and can be developed and extend independently.

From this point of view, CMNN supports the second level only partially. Namely, functions for reasoning and reaction are supported with the concepts of `Planning Table` and `Applicability Rules`, while all others must be explicitly modeled and developed each time.

VI. CONCLUSION

Context aware self-adaptable applications, also in the field of business process management, are becoming very popular in recent years [24]. Especially for knowledge-intensive processes, that are typically very flexible, the context plays an important role during the execution of a process instance. In this paper, we investigated how context information of process instances can be used in CMMN to allow runtime flexibility during process execution.

We have shown that using the specified planning table and applicability rules, it is possible to model a very often situation in the health care domain, where a specific workflow of patient treatment is heavily influenced by results of patient state and effects of already performed (i.e. applied to the patient) activities within the same process case. Similarly, this approach is applicable to any other business domain which requires process flexibility based on contextual information.

However, CMNN is lacking support for many typical functions of context-aware systems. Like other business process modeling languages, CMMN is intended to model only high level of business logic, whereas other details are left to be specified and developed at the level of specific applications. However, due to importance of context-awareness in today's business processes, an extension of CMMN with concepts for an explicit support for modelling context-aware functions as well as integration with external context management infrastructures would be needed for modeling context-aware knowledge-intensive processes.

REFERENCES

[1] D. Auer, S. Hinterholzer, J. Kubovy, and J. Küng, "Business process management for knowledge work: Considerations on current needs, basic concepts and models," in *Novel Methods and Technologies for Enterprise Information Systems. ERP Future 2013 Conference, Vienna, Austria, November 2013, Revised Papers*, 2014, pp. 79–95.

[2] H. R. Motahari-Nezhad and K. D. Swenson, "Adaptive Case Management: Overview and Research Challenges," in *2013 IEEE 15th Conference on Business Informatics (CBI)*, 2013, pp. 264–269.

[3] C. Di Ciccio, A. Marrella, and A. Russo, "Knowledge-Intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches," *J. Data Semant.*, vol. 4, no. 1, pp. 29–57, Apr. 2014.

[4] W. M. P. Van Der Aalst, M. Weske, and D. Grünbauer, "Case handling: A new paradigm for business process support," *Data Knowl. Eng.*, vol. 53, no. 2, pp. 129–162, 2005.

[5] Object Management Group, *OMG Unified Modeling Language (OMG UML) Version 2.5*. 2015.

[6] Object Management Group, *Business Process Model and Notation 2.0*. 2011.

[7] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, and W. Van Der Aalst, "Process flexibility: A survey of contemporary approaches," *Lect. Notes Bus. Inf. Process.*, vol. 10 LNBIP, pp. 16–30, 2008.

[8] Object Management Group, *Case Management Model and Notation 1.1 (Beta 2)*. 2015.

[9] A. K. Dey and G. D. Abowd, "Towards a better understanding of context and contextawareness," in *Workshop on the What, Who, Where, When and How of Context-Awareness, affiliated with the ACM Conference on Human Factors in Computer Systems*, 2000.

[10] N. Herzberg, K. Kirchner, and M. Weske, "Modeling and Monitoring Variability in Hospital Treatments: A Scenario Using CMMN," in *Business Process Management Workshops*, 2014, pp. 3–15.

[11] T. Strang and C. Linnhoff-Popien, "A Context Modeling Survey," in *First International Workshop on Advanced Context Modelling, Reasoning and Management*, 2004.

[12] Object Management Group, *UML 2.4.1 Superstructure Specification*. 2011.

[13] Object Management Group, *OCL 2.3.1 Specification*. 2010.

[14] J. Warmer and A. Kleppe, *The Object Constraint Language: Getting Your Models Ready for MDA*. Addison Wesley, 2003.

[15] M. Reichert and B. Weber, *Enabing Flexibility in Process-Aware Information Systems*. Springer Berlin Heidelberg, 2012.

[16] M. Pesic and W. M. P. van der Aalst, "A Declarative Approach for Flexible Business Processes Management," *Bus. Process Manag. Work.*, pp. 169 – 180, 2006.

[17] A. Hallerbach, T. Bauer, and M. Reichert, "Capturing variability in business process models: the Provop approach," *J. Softw. Maint. Evol. Res. Pract.*, vol. 22, no. 6–7, pp. 519–546, 2010.

[18] M. Rosemann and W. M. P. van der Aalst, "A configurable reference modelling language," *Inf. Syst.*, vol. 32, no. 1, pp. 1–23, 2007.

[19] W. M. van der Aalst, R. S. Mans, and N. C. Russell, "Workflow Support Using Proclets: Divide, Interact, and Conquer," *IEEE Data Eng. Bull.*, vol. 32, no. 3, pp. 16–22, 2009.

[20] M. Reichert and P. Dadam, "Enabling adaptive process-aware information systems with ADEPT2," in *Handbook of Research on Business Process Modeling*, 2009, pp. 173–203.

[21] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[22] O. Saidani and S. Nurcan, "Towards context aware business process modelling," in *8th Workshop on Business Process Modeling, Development, and Support (BPMDS'07), CAiSE*, 2007.

[23] J. Andersson, L. Baresi, N. Bencomo, R. de Lemos, A. Gorla, P. Inverardi, and T. Vogel, "Software engineering processes for selfadaptive systems," in *Software Engineering for Self-Adaptive Systems II*, 2013.

[24] U. Kannengiesser, A. Totter, and D. Bonaldi, "An interactional view of context in business processes," in *S-BPM ONE-Application Studies and Work in Progress*, 2014, pp. 42–54.