

The Old Therapy for the New Problem: Declarative Configurable Process Specifications for the Adaptive Case Management Support

Irina Rychkova¹, Selmin Nurcan^{1,2},

¹ CRI, Université Paris 1, Pantheon-Sorbonne, 90, rue Tolbiac, 75013, Paris, France

² IAE de Paris Sorbonne Graduate Business School; 21, rue Broca 75005 Paris, France
{Irina.Rychkova, [Selmin.Nurcan](mailto:Selmin.Nurcan@univ-paris1.fr)}@univ-paris1.fr

Abstract. The Case Management Process Modeling RFP released by OMG in 2009 expresses the particular demand of practitioners in the case management solutions. The case is defined as “A situation, set of circumstances or initiative that requires a set of actions to achieve an acceptable outcome or objective.” In this paper we consider an example of the case management process - the mortgage approval process. We formulate 5 challenges encountered while modeling this process using a traditional, activity-oriented modeling formalism, i.e. BPMN. We argue that the research methodologies developed during the past decade can be successfully applied to case management modeling. We propose the use of declarative specifications, variability modeling, and FOL-based semantics for modeling descriptive processes and, in particular, case management processes. We assemble these theoretical concepts in the form of DeCo process specifications that extend the BPMN notation.

Keywords: Adaptive Case Management, Declarative process specifications, Configurable processes, First Order Logic.

1 Introduction

The Case Management or Adaptive Case management is an emerging topic that has been extensively discussed during the last months by the BPM community. In a legislative system or health care, the notion of case has been known for many years: here by a case we understand the set of circumstances or facts related to a criminal act or a patient condition that requires a decision making and a treatment with respect to some norms or regulations. Whereas the norms are well defined, the case-related conditions can vary widely and evolve with time, preventing the agent responsible for the case treatment from applying a standard predefined template or model. The similar notion of the case has been recognized in business process management.

The Case Management Process Modeling (CMPM) RFP released by OMG on September 2009 expresses the particular demand of practitioners in the case management solutions [1]. OMG defines case management as “A *coordinative and goal-oriented discipline, to handle cases from opening to closure, interactively between persons involved with the subject of the case and a case manager or case team.*” Systematic improvement of this process based on the user experience is one of the main objectives of adaptive case management approaches. This objective can be achieved by

following the *social production* principle defined by the social software [26]: “*Social production is the creation of artifacts, by combining the input from independent contributors without predetermining the way to do this.*” Case management is an example of knowledge-intensive process that cannot be fully determined at design-time. Mechanisms of the process evolution and actors contributing in this evolution cannot be specified in advance either: for example, a manager is not any longer a passive process user, it is by aggregation, analysis, and mining [2] of case stories and associated user/designer/manager/customer experience the case management process can be improved. Therefore a mechanism to integrate the “*innovative contributions not identified or planned in advance*” should be provided.

In order to benefit from the social software principles in business process management and case management in particular, the possibility to communicate, negotiate, and change a process definition based on the aggregated knowledge should be provided at all levels of the process lifecycle starting from the design. Thus, an appropriate formalism for business process modeling has to be selected.

For the traditional formalisms accepted in the industry, such as BPMN or EPC, this represents a real challenge. Being almost systematically imperative and activity-driven, these formalisms encourage the early specification of the explicit order in which the activities of the process will be executed. On the other hand, these formalisms are often implicit in specifying data that is circulating throughout the process. Therefore, while presenting powerful techniques for modeling and control for the prescriptive business processes, these formalisms fail in specifying knowledge-intensive processes and case management processes in particular. Does that mean that to address the Adaptive Case Management a brand-new formalism is needed?

We argue that the research methodologies and techniques developed during the past decade can be successfully tuned and then applied for the case management modeling. In this paper we discuss the use of Declarative specifications [3], Variability modeling [4], and FOL-based Formal semantics for modeling descriptive processes [5] and, in particular, case management processes. We assemble these theoretical concepts in the form of a modeling approach that we call DeCo – for Declarative Configurable process specifications. DeCo process specifications extend the BPMN notation (a de-facto standard for process modeling) providing a mechanism for descriptive process modeling, formal analysis, and step-wise evolution.

The reminder of this paper is organized as follows. In Section 2 we provide the business process models taxonomy, position the case management in this taxonomy and discuss the existing business process modeling formalisms. In Section 3 we consider a mortgage approval process and attempt to model this process in BPMN-BizAgi (www.bizagi.com). Based on the encountered challenges, we formulate 5 issues that have to be addressed by a case management process modeling technique. In Section 4 we introduce DeCo process specification and illustrate how the aforementioned issues are handled in DeCo. Section 5 presents our conclusions.

2 From Business Process to Case Management

2.1 Process model taxonomy

Business processes models can be roughly divided into two categories: *prescriptive* and *descriptive*. Whereas prescriptive process models specify how things must/should be

done, a descriptive process model aims at recording and providing a trace of what happens during the business process [6][25].

In practice, *prescriptive models* are used to specify processes with predictable sequences of simple tasks and well defined coordination rules (e.g. repetitive, highly automated production processes). Such process can be fully specified at design-time. Prescriptive modeling style, though, is inappropriate for knowledge-intensive processes based on the actor collaboration and information exchange. These processes are characterized by a weak predictability of task sequences and partially defined coordination rules; they can be only “sketched” at design-time by *descriptive models*. Fig.1 shows the examples of processes that can be specified using one or another (or both) modeling styles.

In the global economy the growing interest of organizations to explore new markets can be observed. However, even a simple business process has to be adapted to a new execution context, depending on, among other aspects, cultural and legal considerations [7]. Thus, we distinguish another two categories of business processes models: *context-specific and configurable*. Highly specialized processes, defined for a given execution environment (e.g. a research experiment in chemistry or physics) can be captured by a *context-specific model*. Such processes are hardly reusable and for each new environment (or context) a new process (and its corresponding model) has to be defined. Today, organizations are interested to consolidate their processes while keeping them customizable in order to reflect the context-specific parameters. Processes requiring a customization (such as role/task assignment, task ordering, and rule selection) upon their deployment can be specified using *configurable process models*.

Case management. The glossary of RFP for CMPM defines *case* as “A situation, set of circumstances or initiative that requires a set of actions to achieve an acceptable outcome or objective. ...” In [1], the case management process is addressed as a knowledge-driven process, where activities do not occur in a predefined order. In addition, case management processes supposed to not only anticipate the change of a business context at deployment, but also has to react in the consistent and organized manner on all the emerging case-related knowledge at run-time. Thus, we claim that case management processes shall be specified using *descriptive, configurable models* (top-right quadrant of our diagram in Fig.1).

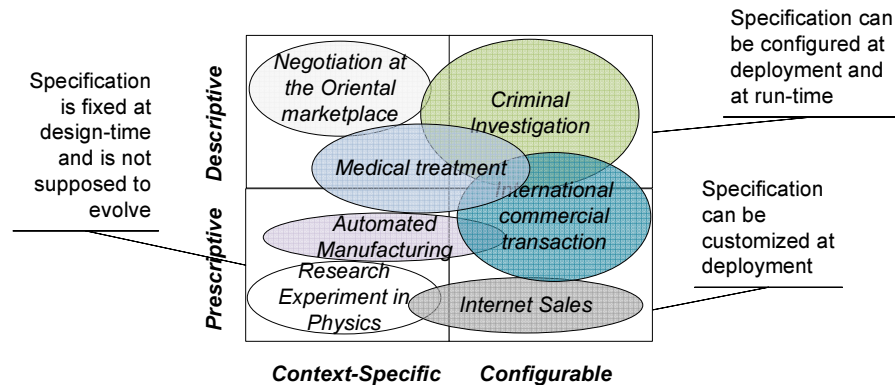


Fig.1. Business process taxonomy. Case management can be considered as a descriptive, configurable process.

2.2 Modeling formalisms

Literature provides various process modeling formalisms that we classify into four categories: activity oriented, product oriented, decision oriented and conversation oriented models [8].

The Business Process modeling formalisms defined by Unified Modeling Language, Event-Driven Process Chain (EPC), and Business Process Modeling Notation (BPMN) gain the wide recognition among practitioners today. All these formalisms are based on the activity-oriented and/or product-oriented paradigm for business process modeling.

The most recent type of process models [9], [10], [11], [12] is based on the *decision-oriented paradigm* according to which the successive transformations of the product are looked upon as consequences of decisions. Conversation models are based on the speech act theory and on the principle that each sentence expressed by someone represents an intention, a commitment.

BPMN 2.0 beta specification [13] published on August 2009 by OMG supports the increasing demand to the modeling collaborations, communication, and human actors involvement. Though, it incorporates the elements of the decision-oriented and conversational paradigms.

A possibility to customize a business process taking into account an environment where this business process is instantiated is a part of the more general problem of flexibility. This problem was identified in [14] and [15] in general and in the context of WfMS respectively: Knoll and Jarvenpaa [14] introduce the term of flexibility as a form of alignment between organizations and their IT systems in turbulent environments. The authors recognize three types of flexibility in the context of IT: flexibility in functionality, in use and in modification. Heintz et al. [15] illustrate the necessity of flexibility in workflow management applications and identify two classes of flexibilities: by selection and by adaption.

Another stream of research e.g. [16][17] favors what we refer to as declarative business process modeling. In [16] the representation of a business process as a trajectory in a state space is introduced. The authors attempt to declaratively describe the dynamics of a business process by defining a notion of a valid state and planning rules that make a state valid. Van der Aalst in [17] presents a case handling paradigm to cope with business process flexibility. In contrast to workflow management, case handling aims to describe what *can* be done to achieve a business goal but not what *should* be done and *how*.

3 Example: The Mortgage Approval Process

Mortgage approval process is a typical example of a case management process. In this section, we provide a generic mortgage approval process description as defined by different financial institutions in the USA. The information provided below results from our study of multiple information sources (e.g. <http://www.homebuyinginstitute.com/>, <http://www.mortgage-resource-center.com/>, <http://homebuyereducation.bankofamerica.com/>, <http://www.homeloancenter.com/> etc.) It represents a compilation of guidelines, recommendations, and descriptions of mortgage approval process, provided by different loan consulting firms, financial advisors, and banks and available on the web.

3.1 The Process Description

A mortgage is a loan for buying a house. The terms and length of the mortgages are negotiable and can be adapted for the applicant's situation. The mortgage approval process can be divided into the following steps: Pre-qualification; Formal application; Document review; Pre-approval; Property appraisal; Final approval; Closing.

The goal of the pre-qualification step is to determine the potential mortgage amount that the applicant is eligible for. The purpose of the formal application is to provide the lender with documents characterizing in details the current financial situation of the applicant as well as his/her employment and credit history. Document review follows the formal application and may include the pre-approval step. The pre-approval letter issued as a result of this step indicates that the applicant is pre-approved by a lender for a specific loan amount. When the property is selected by the applicant, the mortgage lender initiates the property appraisal. The appraisal step defines the amount of the mortgage and a corresponding down payment. The lender makes "approve", "not approve", or "approve with conditions" final decisions based on the document review and the appraisal results. If the loan is approved, a commitment letter is issued for the applicant, and a closing date is set up. Closing (also called settlement) is a final step. During the mortgage closing, the mortgage lenders will need to purchase the house and hold the title as the applicant makes payments to them.

3.2 Specification of the Mortgage Case Management Process Using BPMN

Considering the complexity of the complete process, in this paper we will focus on the Formal Application process step. The text below describes this step in details.

Mortgage Approval: Formal Application

- 0 The applicant can request the application package by e-mail or by post. Alternatively, all the forms can be accessed on the Web.
- 1 Mortgage application can be submitted electronically or during a personal meeting with the mortgage lender.
- 2 The exact set of documents may vary depending of the financial institution and the particular situation of an applicant. These documents may include: The Social Security card; Record for past two years for residence address; Employer name, address; W-2 tax forms; federal income tax returns; Most recent pay-stubs, etc.
- 3-5 During the application, the lender provides the applicant with a Good Faith Estimate (GFE) of costs of loan closing; the applicant can be asked to make a final decision on the type of mortgage loan; also an interest rate for the loan can be locked in this phase.
- 6 Some lenders will give to the applicant an access to their website where the applicant can check on the approval status of his/her package.
- 7 Usually an application fee and the appraisal fee will have to be paid by the applicant during the mortgage application submission.

The main purpose of this description is to illustrate the diversity of activities, actors, and information involved and to stress the unpredictable nature and variability of this process – the characteristics that make it's modeling a challenging task. Figure 2 presents the model of the Mortgage approval process specified using BizAgi modeling tool.

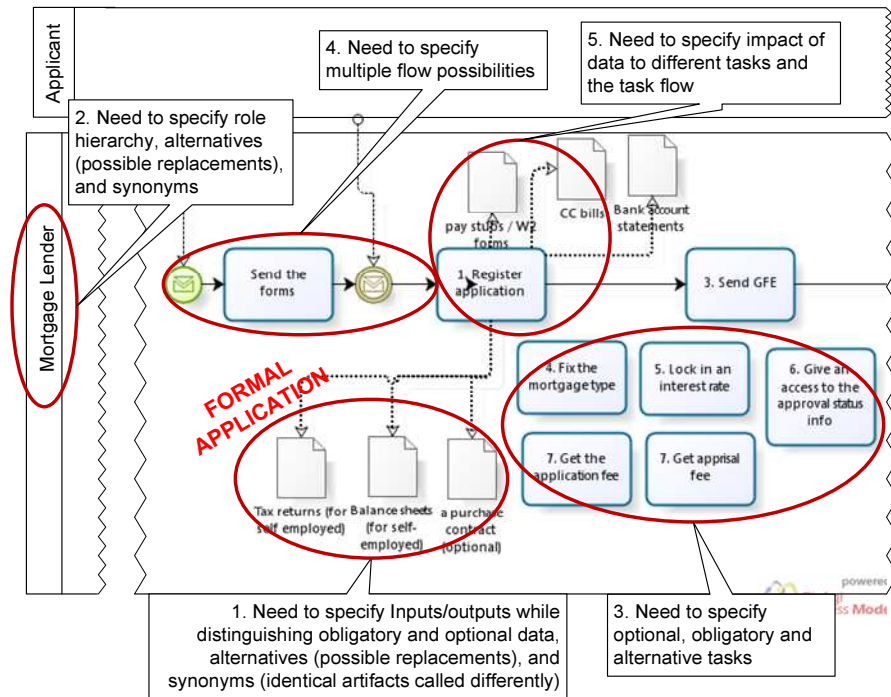


Fig. 2. BPMN specification of the Formal Application in the Mortgage Approval process.

3.3 Discussion

Modeling the mortgage approval process in BPMN, we have encountered the following challenges:

Optional tasks. Modeling optional tasks represents one of the major challenges. For example, <Send the forms> task is not needed if an applicant has already downloaded forms from the Web or if he/she fills in the application on-line.

Task Ordering. Many tasks, being obligatory for the process, cannot be preordered at design time. Based on the norms and policies of the concrete place/institution, their order may be specified at deployment (e.g. <Get an access to the approval status info>, <Get application fee>). However, some actions (e.g. <Lock in an interest rate>, <Get appraisal fee>) can be ordered only at run-time as they will be executed based on the data availability.

Within the traditional BPMN formalism, task ordering can be expressed using a gateway mechanism; however, considering the large number of such ordering options, this mechanism is not efficient. An explicit specification of conditions under which one or another ordering should be applied is also challenging. In our BPMN diagram, the appropriate solution found was to leave the tasks that could not be explicitly ordered at design time outside of the process flow.

Optional/alternative data objects and synonyms. The mortgage approval process consumes and produces a vast amount of data artifacts. Some of these artifacts may be *obligatory*, whereas the others can emerge (or be requested) in specific situations only.

For example, the mortgage loan applicant can be asked to provide the mortgage lender with one or multiple different tax forms. The form W-2 (Wage and Tax Statement) is typically requested from all the applicants in USA (obligatory); the form 2555 (entitled Foreign Earned Income) should be provided by taxpayers who have earned income from sources outside the United States. In order to adapt to different applicant's situations, *the alternative data artifacts* should also be specified. For example, the form 1040 is a commonly accepted tax return form, whereas the form 1040EZ is a simplified version of this form: for single and joint filers with no dependents the form 1040EZ can replace the form 1040 [source: wikipedia]. Some artifacts also can be called differently from one organization (or country) to another: the tax forms (or IRS forms) defined for USA corresponds to another tax forms in UK. For a mortgage lender working in both countries, having these *synonyms* explicit can be beneficial.

Explicit modeling of data impact on decision making/task ordering. Data artifacts impact the task ordering and decision making at run-time. For example, if the purchase contract is provided during the application, the mortgage lender may want to immediately initiate the appraisal process. We have used BPMN artifacts to specify the relations between data and tasks where these data is required. The way the data is used in general can be indicated using annotations. However, neither one nor another mechanism is formalized in BPMN. Implicit impact of data on the decision making stays out of the process modeling scope within the existing formalism.

Role assignment. Whereas abstract process participants – Mortgage lender and Applicant – are easily recognizable, concrete role assignment may depend on the financial institution: small banks have only several (2-4) roles associated with the process, whereas in the big agencies the application processing and decision making is more likely to be assigned to a number of different roles with different responsibilities. These roles can be assigned only upon deployment. As our study shows, the same task can also be performed by several roles – this postpones the role assignment until the run-time.

In this section, we have listed only several challenges related to the case management process modeling. The modeling of negotiation and communication, internal decision making, modeling data/actor/task ranges are the problems that will be addressed in the future. To conclude our discussion, we formulate five requirements to a descriptive process modeling approach that the traditional formalisms can hardly meet and that we are going to address in further sections:

1. *Need to specify inputs/outputs while distinguishing obligatory and optional data, alternatives (possible replacements), and synonyms (identical artifacts called differently).*
2. *Need to specify role hierarchy, alternative roles, and synonyms.*
3. *Need to specify optional, obligatory and alternative task.*
4. *Need to specify multiple flow possibilities.*
5. *Need to specify impact of data to different tasks and the task flow.*

4 Declarative Configurable Process Modeling Notation (DeCo)

In this work, we present the process modeling approach based on Declarative Configurable (DeCo) process specifications that extend the BPMN notation (a widely adopted standard for process modeling with more than 60 current implementations).

4.1 Theoretical Foundations

The theoretical foundations of DeCo are grounded on Declarative modeling [3], Variability modeling [4], Refinement theory, and formal methods - the paradigms, proven in the research community. This work is largely based on the authors' research reported in [5][18][19][12] and is inspired by the results presented in [20][21].

The declarative specifications for modeling business processes have been presented in [19]: this approach is based on the systematic modeling of process-related data. This allows us to introduce the notion of *state*. Each process task then can be associated with a set of *pre-states* - the states where this task can (but not necessary will!) be executed - and a set of *post-states* - the states resulting from the task execution respectively. Consequently, the process specification represents a set of activities (tasks to be done) with no predefined execution scenario.

The declarative modeling principles allow one to postpone the decision making about the process control flow until its deployment or even execution. As soon as declarative process specification evolves - the mechanism to validate and to control this evolution is required. In the work presented in [19] the evolution from the declarative, nondeterministic process specification at design time towards precise (imperative) process specification at deployment is represented as a set of refinement steps. The notion of refinement for graphical specifications, adopted from software engineering [3], is presented in [5]. In this work, the formal semantics for graphical specifications is defined based on first order logic and set theory. These semantics allows us to reduce the problem of refinement verification to the validation of the first-order logic formula and provides the means for automated process analysis and control using the tools for automated analysis of program specifications defined in software engineering.

The technique presented in [19] allows one to demonstrate that different control flow configurations of the process are valid with respect to a high-level declarative design specification. However, to deal with descriptive processes (and the case management in particular) the process configurability should not be limited by a control flow.

In the literature, several major perspectives of the process models are specified [23]: *the control flow perspective* that captures the temporal ordering of process tasks, events, and decision points; *the data perspective* that captures the lifecycle of data objects (creation, usage, modification, deletion) within the process; *the resource perspective* that describes how the process is carried out within the organization and deals with roles and resource assignments; *the operational perspective* that addresses the technical aspects of process execution and specifies the elementary process tasks and their assignment to concrete applications or application components of the organizations; *the context perspective* that describes the attributes related to the process execution context; *the performance perspective*, addressing the process cost effectiveness.

In [5][20][21][22] the concept of configurable process has been presented and the modeling formalism to deal with process configurability at multiple perspectives is defined. Namely the authors present the Configurable Integrated EPC (C-iEPC) modeling notation that extends the well known Event Process Chain (EPC) notation and addresses the process configurability along the control-flow, data, and resource perspectives. According to this approach, "Given a configurable process model, analysts are able to define a configuration of this model by assigning values to its variation points based on a set of requirements. Once a configuration is defined, the model can be individualized automatically.." Individualization process can be considered as

automated synchronization of the process model perspectives in respond to each configuration decision. This guarantees the correctness of individualized process models by construction. To introduce the notion of correctness and to reason about individualized and configurable models, authors define formal semantics for C-iEPC based on FOL [20].

4.2 Multi-perspective configurability

Inspired by the approach presented in [20]-[22], we propose to address the 3 out of 5 requirements defined in the previous section using the multi-perspective configurability as follows:

1. *Need to specify inputs/outputs while distinguishing obligatory and optional data, alternatives (possible replacements), and synonyms (identical artifacts called differently).* - This requirement can be addressed by providing configurability along the data perspective.
2. *Need to specify role hierarchy, alternatives (possible replacements), and synonyms.* - This requirement can be addressed by providing configurability along the resource perspective.
3. *Need to specify optional, obligatory and alternative tasks.* This requirement can be addressed by providing configurability along the operational perspective.

Due to space limitations, we consider in more details only the DeCo implementation of the data configurability. Figure 3 illustrates the data object hierarchy models for Tax forms and Tax return forms required for formal mortgage application in USA. Vast amount of data artifacts consumed and produced by a process represents a challenge for modeling. The hierarchy model of data objects describes:

- generalization-specialization relations between data objects;
- alternatives – data objects that can replace the data object originally required by the task; For example, the Form 1040EZ can be considered as an alternative of the obligatory 1040 form. The rule 3.2 specifies the condition where this alternative is applicable. Such rule can be described as a text or formalized in FOL.
- synonyms – different terms referred to the same data object. For example, in USA, both tax forms and tax return forms can be called Internal Revenue Service (IRS) forms [source: wikipedia]

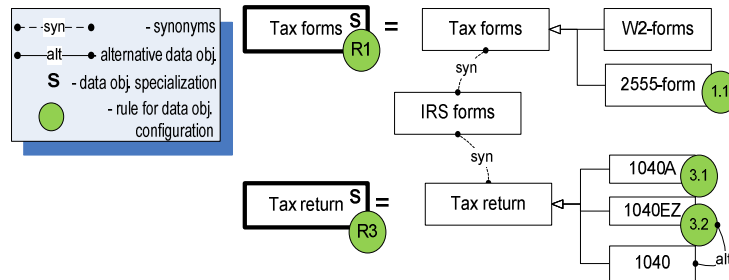


Fig. 3 The data object hierarchy model for the Formal Application

4.3 Declarative specification of tasks

The last two requirements:

4. *Need to specify multiple flow possibilities*

5. *Need to specify impact of data to different tasks and the task flow*

can be addressed applying the declarative modeling principles as explained in [19][5]. Formal semantics permits to express the action contracts in terms of pre-condition, post-condition and invariants. We define a formal semantics for DeCo specifications based on first-order logic (FOL). Similarly to results presented in [5], it can be used for the mapping of a DeCo specification to the Alloy specification language [24] for further validation.

At run-time, at any moment a process enactment is characterized by its **state** (a case state). We define a case state in DeCo as a vector $\bar{X} = (p_1, p_2, \dots, p_n)$. The components p_1, p_2, \dots, p_n are values of data objects related to this case at a given moment of time. A state space Σ is a set of all possible states of a case.

For every process task A we define a precondition and a postcondition. Postcondition A_{post} is a condition that a case meets after the task termination. Precondition A_{pre} specifies a condition that must hold upon the task invocation: If A is started in a state satisfying A_{pre} , it is guaranteed to terminate in a state satisfying A_{post} .

Precondition and postcondition are modeled as predicates over state space Σ :

$$A_{pre} : \Sigma \rightarrow \{true, false\},$$

$$A_{post} : \Sigma \times \Sigma \rightarrow \{true, false\}$$

In addition to pre- and postcondition, invariants for process tasks can be specified.

Invariant A_{inv} is a condition that holds before and after the task execution.

Task A defines a transition of the case from state \bar{X} to state \bar{X}' (pre- and post-states respectively). We define a task in DeCo as a binary FOL-formula $A : \Sigma \times \Sigma \rightarrow \{true, false\}$. We specify the task using logical implication between precondition and postcondition:

$$A(\bar{X}, \bar{X}') \stackrel{def}{=} A_{pre}(\bar{X}) \rightarrow A_{post}(\bar{X}, \bar{X}')$$

If at a given state \bar{X} the precondition A_{pre} of the task A holds, then the case will be transitioned to a state \bar{X}' , for which the postcondition of A - A_{post} - holds.

Preconditions, postconditions and invariants explicitly relate tasks with data objects within a case.

Example:

To specify the contract for the <Send the Forms> task from Fig.2, we define 2 supplementary predicates:

`isReceived(d: DataType, a: Applicant) { $\exists m: \text{ReceivedMessage} \mid m.\text{content} = d \wedge m.\text{src} = a$ }`

`isSent(d: DataType, a: Applicant) { $\exists m: \text{SentMessage} \mid m.\text{content} = d \wedge m.\text{dst} = a$ }`

The former predicate evaluates to True if in the process there exists a message m received from an applicant a that contains a data object d . Along those lines, the second predicate evaluates to True when the message m with a content d is sent to an applicant a . We define the precondition and the postcondition for the <Send the Forms> task as follows: Pre: `isReceived(FormRequest, applicant)`; Post: `isSent(AppForms, applicant)`

There are no invariants specified for this task. The task itself can be denoted as follows:

$\forall \text{apl:Applicant} \mid \text{isReceived}(\text{FormRequest}, \text{apl}) \rightarrow \text{isSent}(\text{AppForms}, \text{apl}) :$

Whenever the process receives a form request from an applicant apl, then the execution of this task will move the process in the state where the application forms are sent to this applicant.

5 Conclusion and Future Work

OMG RFP demonstrates the increasing interest and the particular needs of the practitioners in the methodologies and tool support for case management process [1]. According to OMG RFP, the objectives of the Case Management methodologies or tools are (i) to accommodate and structure the knowledge of participants about the current case, (ii) to provide the information of the previous similar cases and help the practitioners to learn from best practices, and (iii) to progressively improve case management processes to make them “*evolve toward prescribed processes*.”

DeCo defines the concepts for systematic gathering, and structuring the information about the data (consumed or produced), the roles, and the tasks of a process. Namely, based on the configurability modeling along multiple process perspectives, we specify generalization/specialization, synonym, and alternative relations.

Compared to traditional business process modeling formalisms, DeCo provides the means to support the prescriptive nature of the case management process: the declarative modeling principles allow the designer to specify the set of activities that could (but not necessary will) be executed during a process invocation together with the conditions under which this execution can be possible: {pre, post, inv}. Though extending it, DeCo is based on the BPMN graphical notation. Our main objective with DeCo is not to provide yet another set of concepts and shapes representing them, but to reuse as much as possible the notation that (a) has already been known and used by practitioners for years and (b) has a significant developers’ support. This can help us to get a useful feedback from the BPM practitioners’ community while conceptualizing DeCo.

Social software is software that supports the interaction of human beings. Thus, being largely dependent on such interactions, case management process can benefit from supporting tools based on the social software principles. The following three important missions can be successfully fulfilled by social software: (i) collection of process requirements for their further incorporation into the process design model; (ii) collection of the information related to the process context for the further process model customization at deployment; (iii) collection of the content (case subject) – related information at the process run-time for the further process model analysis and evolution (e.g. implementing process mining [2]).

References

1. Object Management Group, Case Management Process Modeling (CMPM) Request For Proposal OMG Document: Bmi/2009-09-23 (2009)
2. W.M.P. van der Aalst. Challenges in Business Process Mining. BPM Center Report BPM-10-01, BPMcenter.org, (2010)
3. N. Wirth, “Program development by stepwise refinement”, Communications of the ACM, 14:221–227. (1971)
4. M. Rosemann and W. M. P van der Aalst. A Configurable Reference Modelling Language. Information Systems, (2007)

5. I. Rychkova, Formal semantics for refinement verification of enterprise models. Dir.: Alain Wegmann. Thèse EPFL, no 4210 (2008). <http://library.epfl.ch/theses/?nr=4210>
6. Van der Aalst W.M.P., Stoffele, M. and Wamelink, J.W.F. Case Handling in Construction, Automation in Construction, 12(3), 303-320. (2003)
7. G. Regev, and A. Wegmann, "Regulation Based Linking of Strategic Goals and Business Processes", Proceedings of the 3rd BPMDS Workshop on Goal-Oriented Business Process Modeling, GBPM'02, London, September 2002.
8. Nurcan S., A Survey on the Flexibility Requirements Related to Business Processes and Modeling Artifacts. Proceedings of the 41st Annual Hawaii International Conference on System Sciences, Big Island, Hawaii, USA, 7-10 January 2008.
9. Dellen, B., Maurer, F., and Pews, G. Knowledge Based Techniques to Increase the Flexibility of Workflow Management. *Data and Knowledge Engineering*, 23(3) North Holland. (1997)
10. Faustmann, G. Enforcement vs. Freedom of Action-An Integrated Approach to Flexible Workflow Enactment. Workshop on Adaptive Workflow Systems. Conference (1998).
11. Nurcan, S., Etien, A., Kaabi, A., Zoukar, I. and Rolland, C. A Strategy Driven Business Process Modelling Approach. Special issue of the Business Process Management Journal on "Goal-oriented business process modeling", Emerald, (2005)
12. Nurcan, S. Business Process Modeling for developing Process Oriented IT Systems, The "Business Process Management Tools and Technologies" track of the Int. IRMA Conf. (2004)
13. Object Management Group, Business Process Model and Notation (BPMN) FTF Beta 1 for Version 2.0 OMG (2009)
14. K. Knoll, S.L. Jarvenpaa, Information technology alignment or "fit" in highly turbulent environments: the concept of flexibility, proceedings of the computer personnel research conference on Reinventing IS.(1994)
15. P. Heintz, S. Horn, S. Jablonski, J. Neeb, K. Stein, and M. Teschke, A Comprehensive Approach to Flexibility in Workflow Management Systems, proceedings of the international joint conference on work activities coordination and collaboration, (1999)
16. M. Khomyakov, and I. Bider, "Achieving Workflow Flexibility through Taming the Chaos". OOIS 2000 - 6th international conference on object oriented information systems. Springer, 2000, pp.85-92. Reprinted in the Journal of Conceptual Modeling, August 2001:
17. W.M.P. van der Aalst, M. Weske, D. Grünbauer, "Case Handling: A New Paradigm for Business Process Support", *Data Knowl. Eng.* 53(2) (2005) 129-162
18. I. Rychkova, G. Regev and A. Wegmann. Declarative Specification and Alignment Verification of Services in ITIL. First International Workshop on Dynamic and Declarative Business Processes (DDBP 2008), Munich, Germany. (2008)
19. I. Rychkova, G. Regev and A. Wegmann, Using Declarative Specifications In Business Process Design.. *International Journal of Computer Science & Applications.* (2008)
20. La Rosa, Marcello and Dumas, Marlon and ter Hofstede, Arthur H.M. and Mendling, Jan Configurable multi-perspective business process models. Preprint submitted to Information Systems. (2009) Available at QUT Digital Repository: <http://eprints.qut.edu.au/>
21. La Rosa, Marcello and Dumas, Marlon and ter Hofstede, Arthur H.M. and Mendling, Jan and Gottschalk, Florian (2008) Beyond Control-Flow: Extending Business Process Configuration to Roles and Objects. In Li, Qing and Spaccapietra, Stefano and Yu, Eric, Eds. Proceedings 27th International Conference on Conceptual Modeling (ER 2008), Barcelona, Spain.
22. Mendling, Jan and La Rosa, Marcello and ter Hofstede, Arthur H.M. Correctness of Business Process Models with Roles and Objects. Elsevier (2008)
23. S. Jablonski and C. Bussler. *Workow Management: Modeling Concepts, Architecture, and Implementation.* International Thomson Computer Press (1996)
24. D. Jackson, "Software Abstractions: Logic, Language, and Analysis", MIT Press.(2006)
25. C. Rolland, S. Nurcan, G. Grosz. Enterprise Knowledge Development: the process view. *Information and Management Journal*, Elsevier, 36:3 (1999), p. 165-184.
26. S. Nurcan, R. Schmidt. Service-oriented Enterprise Architecture for Enterprise Engineering: Introduction. Proceedings of EDOC 2009 IEEE Workshops and Short papers.