

# Towards Adaptability and Control for Knowledge-Intensive Business Processes: Declarative Configurable Process Specifications

Irina Rychkova

CRI, Université Paris 1, Pantheon-Sorbonne  
[Irina.Rychkova@univ-paris1.fr](mailto:Irina.Rychkova@univ-paris1.fr)

Selmin Nurcan

CRI, Université Paris 1, Pantheon-Sorbonne,  
IAE de Paris Sorbonne Graduate Business School  
[Selmin.Nurcan@univ-paris1.fr](mailto:Selmin.Nurcan@univ-paris1.fr)

## Abstract

*Recent discussions raised by the BPM community illustrate the increasing demand of practitioners in the solutions for descriptive, knowledge-intensive processes. As a matter of fact, a majority of modeling formalisms presented on the market today fails in providing an appropriate level of adaptability while ensuring validation and control for such processes.*

*In this work we present a modeling approach based on Declarative Configurable specifications that capture the processes at three abstraction levels: design, deployment, and execution. As a result, DeCo allows for separation of the process goals from the process means and, finally, from the process realization details improving adaptability and control.*

## 1. Introduction

Recent publications illustrate the increasing interest of BPM practitioners in both methodology and tools for unstructured, knowledge-intensive processes such as case management processes (CMP) [1][2]. In its report from November 2009 [1], Gartner Inc. has published 5 Business Process Management predictions for 2010 and beyond that reflect the challenges and emerging BPM trends. Namely, they acknowledge the need for support of unstructured, dynamic, knowledge-intensive processes. According to Gartner, “By 2013, dynamic BPM will be an imperative for companies seeking process efficiencies in increasingly chaotic environments.”

Another prediction highlights the importance of developing a business process modeling culture in the organizations: “By 2014, 40% of business managers and knowledge workers in Global 2000 enterprises will use comprehensive business process models to support their daily work, up from six percent in 2009.” Janelle Hills, a vice president in Gartner Research, comments on this prediction in [3]: “Comprehensive, graphical (rather than textual) and explicit process

*models that capture and represent organizational knowledge will create a lingua franca for business and IT roles.”*

The Case Management Process Modeling (CMPM) Request For Proposal released by OMG on September 2009[2] expresses the particular demand of practitioners in the case management solutions. 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.” Case management processes have multiple applications, including “licensing and permitting in government, insurance application and claim processing in insurance, patient care and medical diagnosis in healthcare, mortgage processing in banking...” [2]. The main resource of a CMP is knowledge obtained as a result of communication between multiple actors/users and stored in a “case file”. There are two types of knowledge CMP deals with: *explicit* knowledge about the case subject (e.g. a patient’s body temperature) and *tacit* knowledge required to manage the case (e.g. experience of a medical team). Whereas explicit knowledge can be stored, managed, and shared easily between the actors, tacit knowledge is hard to represent and to transfer – it exists in the form of “experience” or “intuition”.

Being largely based on tacit knowledge, CMPs can be hardly analyzed, improved, and replicated by the organizations. This problematic is extensively explored by researchers in knowledge management discipline [4][5][6]. Nonaka and Takeuchi, in [6], describe conversion of different types of knowledge within organizations as an iterative and spiral process that includes four phases: socialization, externalization, combination and internalization (SECI). This process explains how individuals’ tacit knowledge is getting transformed into, first, tacit, and then explicit knowledge of an organization. According to [6], SECI stimulates the new knowledge creation and application

resulting in the improved processes. Along those lines, experience is one of the main objectives of adaptive case management approaches.

The dominant business process and workflow supporting formalisms are almost systematically activity oriented: the main advantage of these formalisms is a possibility to generate executable process specifications and also to simulate and validate these specifications prior to the process deployment. This technique guarantees better control over processes and helps to avoid costly errors. However, providing a high degree of control, aforementioned formalisms suffer from the lack of adaptability: once the process is designed, it becomes difficult (if at all possible) to adjust it with respect to a changing execution context or emerging knowledge. Thus, being well suited for prescriptive, context-specific business processes, these modeling formalisms are not appropriate for the knowledge-intensive processes and the case management process modeling (CMPM) in particular.

Inspired by the OMG request for proposal, we formulate the following question: *Can we provide a formalism that would support the process run-time adaptability while providing control and ensuring process correctness?*

During the last decade, process adaptability and evolution support remains the central area of interest for many researchers: among others, numerous contributions of the groups at the University of Ulm (Dadam, Reichert et al) and the Eindhoven University of Technology (van der Aalst et al) can be emphasized.

In this work we introduce the modeling approach for CMP based on Declarative Configurable (DeCo) specifications. The originality of this approach is in its attempt to distill different ideas explored in different periods of time by different research groups in BPM and software engineering: declarative modeling, model verification, variability modeling.

Compared to traditional modeling approaches, DeCo explicitly defines three modeling levels that capture knowledge-intensive processes (i) *at design* and result in a configurable, non-deterministic design specification that focuses on the process goals; (ii) *at deployment* and result in a customized design specification that reflects the process deployment environment; (iii) *at execution* and result in a fully deterministic process trace that reflects post-factum a single process enactment.

To enable the process control, we formalize the relationships between the process specifications at these three levels as refinement relationships [7]. Using formal methods of refinement validation, we can ensure that the process customizations made at the deployment level and the process execution are aligned with the process goals specified at design.

systematic improvement of CMP based on the user

In this paper, we focus on the DeCo modeling levels definition. The formalization of refinement relations between specifications at these levels and refinement validation will be addressed in our future publications. We illustrate DeCo specifications on the example of a mortgage approval process.

The reminder of this paper is organized as follows: In Section 2 we discuss the business process taxonomy and position CMPs within this taxonomy. We also discuss the existing trends in process modeling, highlighting challenges and perspectives in modeling knowledge-intensive processes and CMPs. Section 3 introduces a mortgage approval process – a CMP example. In Section 4 we present DeCo process specifications: we discuss the theoretical foundations, define the three modeling levels for our specifications, and introduce the DeCo process lifecycle. In Section 5 we illustrate the graphical notation of DeCo on the example of mortgage approval process. This notation extends the BPMN notation providing the concepts for declarative process modeling, formal analysis, and step-wise evolution. In section 6 we present our conclusions.

## 2. Case management process modeling: a new challenge for BPM

### 2.1. Process taxonomy

Business Processes can be roughly divided into two categories [8]:

1. *Prescriptive processes*: processes with predictable sequences of simple tasks and well defined coordination rules. Such process can be fully specified at design-time.
2. *Descriptive processes*: knowledge intensive processes, strongly based on the actor collaboration and information exchange. They can be also characterized by a weak predictability of task sequences and only partially defined coordination rules. Such process can be only “sketched” at design-time; at run-time this process should reflect the emerging knowledge and adapt accordingly.

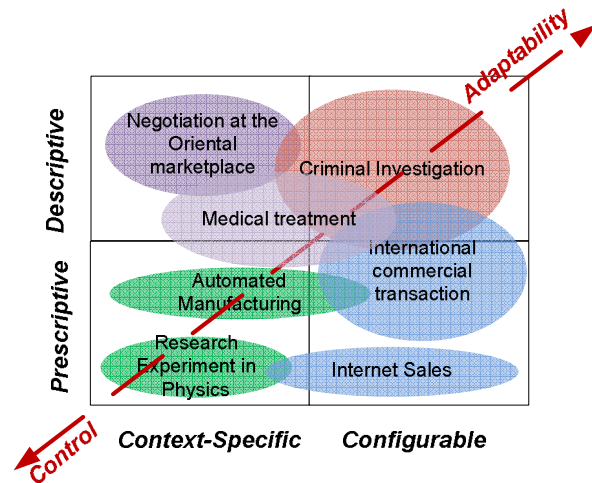
In the global economy the growing interest of organizations to explore new markets can be observed. However, even a simple business process has to be adapted to a new execution context, depending on, among other aspects, cultural and legal considerations [9]. We distinguish another two categories of business processes:

1. *Context-specific processes*: highly specialized processes, defined for a given execution context (e.g. a research experiment in chemistry or physics).
2. *Configurable processes*: processes requiring a customization (e.g. role/task assignment, task ordering,

and rule selection) upon its deployment in order to reflect the context-specific parameters. For such processes the “deployment-time” adaptability is required.

Figure 1 illustrates our process taxonomy and provides the reader with several examples.

**Case management.** The glossary of RFP for CMPM in [2] defines *case* as “A situation, set of circumstances or initiative that requires a set of actions to achieve an acceptable outcome or objective...”. Case management processes (CMP) can be characterized by the following: it is driven by emergent knowledge about the case subject or the environment; largely based on tacit knowledge (e.g. human expertise); highly unpredictable; difficult to replicate; hard to analyze and improve as no HOWTOs available. According to the definition provided in [13], CMPs can be considered as knowledge-intensive business processes. According to our classification in Fig. 1, CMP 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. Therefore, we identify CMP as descriptive, configurable processes and position them in the top-right quadrant of our diagram in Fig.1.



**Figure 1: Business process taxonomy: CMPs can be considered as descriptive configurable processes.**

## 2.2. Modeling formalisms

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. According to the classification provided in [8], these formalisms are based on the activity-oriented and/or product-oriented paradigm for business process modeling.

BPMN 2.0 beta specification [10] extends BPMN 1.0 distinguishing Private (internal) business processes, Abstract (public) processes, and Collaboration (global) processes. It enables model orchestrations and choreographies as stand-alone or integrated models and, therefore, provides better support for modeling collaborations, communication, and human actors’ involvement.

The process models presented in [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.

Van der Aalst in [15] 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*.

In [16], an approach for mining change logs in adaptive process management systems is presented. Since adaptive processes provide the information about process changes – these changes can be used for organizational learning and process improvement.

In [8], the taxonomy of flexible/adaptive workflow approaches is presented. Approaches, providing a flexibility by selection (a priori) and by adaptation (a posteriori) are distinguished. *The flexibility by adaptation* allows for evolution (adaptation) of the process definitions (design models) as well as process instances during their execution. The modeling formalisms supporting *the flexibility by selection* offer the capacity to deal with the change of the process deployment environment on the instance level but do not result in an evolution of process definitions (design models).

In [14], the authors distinguish three classes of business processes: operational, practice-oriented and case-handling. They use the term *organizational practices* to address the result of externalization of tacit knowledge by organizations. The question of how to develop and use these practices in order to provide systematic analysis and improvement of CMP is one of the main objectives of adaptive case management.

In this work, we present the process modeling approach based on Declarative Configurable (DeCo) process specifications that extend the BPMN notation with the constructs for declarative and variability modeling support. This approach explores flexibility by adaptation and assumes an eventual evolution of the initial (descriptive) process definition via enactment analysis or process mining [17].

## 3. Example: the mortgage approval process

Mortgage approval process is a typical example of a case management process. In this paper, 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.

A mortgage is a loan for buying a house. The mortgage approval process can be divided into the following steps: Pre-qualification; Formal Application; Document Review; Pre-approval; Property Appraisal; Final Approval; Closing. Whereas the order of these main steps varies rarely, the documents required, the sequences of tasks, the participants and the length of each step can be different depending on the place, the financial institution's policies, and the applicant's situation and requirements. In this paper we will focus on the Formal Application process step:

#### **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 formal application, the mortgage 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 variability of activities, actors, and information involved and to emphasize the unpredictability and diversity of the process scenarios – the characteristics that make it's modeling a challenging task.

## **4. Declarative Configurable process specifications: DeCo**

### **4.1. Theoretical foundations**

While designing DeCo modeling approach, the following guiding principles have been considered:

1. **Fixed constraint often means lost opportunities:** DeCo formalism should not limit but guide the modeler, encouraging him/her to think in terms of possibilities, alternatives, and variations instead of rules and constraints.

2. **Very little is certain at design-time:** DeCo specifications should separate the process goals (what should be achieved?) from the process means (what should be done to achieve it?) and from the process realizations (how?). There is a proper time to address each of these aspects which is not necessary a design-time.

3. **Controlled anarchy:** DeCo formalism should support the mechanisms for assembling *meaningful* process specifications from the predefined process parts or variants and for controlling this meaningfulness.

In order to follow the guidelines above, DeCo formalism is grounded on:

- The declarative modeling paradigm [18] that supports nondeterminism and, thus, releases a designer from the obligation to determine the exact process scenarios upfront;
- The variability modeling [19] that provides the mechanism to specify multiple variants for different process components (tasks, data objects, actors, etc), while hiding the complexity and keeping the model reasonably readable;
- The refinement theory and formal methods [18][20] - the paradigms, defined in software engineering and providing the methodology and tool support for program specifications analysis and validation. In our context, we use these methodology and tools to ensure the correctness of process customization and process realization with respect to the design specification.

This work is largely based on the research reported in [7][21][22][12] and is inspired by the results presented in [23][24]. DeCo approach enhance the results presented in [22] providing an explicit notion of modeling levels and extending the graphical notation.

**4.1.1 The declarative modeling principles** allow one to postpone the decision making about the process control flow until its deployment or even execution. The declarative specifications for modeling business processes have been presented in [22][21]: this approach is based on the systematic modeling of process-related data. This allows us to introduce the notion of process *state* (or case

state) 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. For every process task  $A$  we define a precondition and a postcondition. Postcondition  $A_{post}$  is a condition that the case meets upon the task termination. It specifies all possible case states after the task completion, including success and failure. Precondition  $A_{pre}$  specifies a condition that must hold upon the task invocation. It specifies all possible case states at which the task can (but not necessary will!) be executed. We specify the task using logical implication between precondition and postcondition as follows:

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

The expression above can be interpreted as follows: *If at a given state  $\bar{X}$  the precondition  $A_{pre}$  of the task  $A$  holds, then the case will be transited to a state  $\bar{X}'$ , for which the postcondition of  $A$  -  $A_{post}$  - holds.*

The process specification in DeCo represents a set of tasks with no explicit execution scenario: at run-time, each task can be dynamically selected from the list of tasks enabled at a given case state.

#### 4.1.2 The refinement theory and formal methods.

As soon as process specification evolves based on the user experience or emerging knowledge, *the mechanism to validate and to control this evolution* is required. In the work presented in [22] the evolution from the declarative, non-deterministic process specification at design time towards precise (imperative) process specification at instantiation is represented as a set of refinement steps. The notion of refinement for graphical specifications, adopted from software engineering [18], is presented in [7]. 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.

**4.1.3 The variability modeling.** The technique presented in [22] 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 [26]: *the control flow perspective, the data perspective, the resource perspective, the operational perspective, the context perspective, and the performance perspective*. In [19][23][24][25] the concept of configurable process has been presented and the modeling formalism to deal with process configurability at multiple perspectives is defined. In their work, the authors address 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. To introduce the notion of correctness and to reason about individualized and configurable models, authors define formal semantics for C-iEPC based on FOL[23].

## 4.2. Multi-level process specifications with DeCo

According to [27], a process definition obtained at design time includes a network of activities, relationships between them, information about individual activities including participants, data, etc.

While modeling knowledge-intensive or descriptive processes, the major challenge is related to the fact that multiple process aspects cannot be identified at design-time and, thus, cannot be included into the process definition. In DeCo, we identify three modeling levels for descriptive processes and, accordingly, define the following process specification types:

1. Configurable design specification;
2. Customized design specification;
3. Realization specification (or process trace).

Below we explain the reasoning behind these three levels.

**4.2.1. Design vs. Deployment.** In our example, the deployment environment of the mortgage approval process can strongly affect the way the process constructed: being defined for the US financial organizations, this process still should be adjusted taking into account legislative norms and regulation of different States. For example, in the western part of US the mortgage closing is done “in escrow.” In the rest of the country the closing is done “face-to-face”. Thus, an organization specialized in mortgage business in the entire US market and interested

to consolidate its processes (instead of keeping different process definition for different States), should anticipate those variations upfront.

One option for this organization can be to specify all possible execution paths as alternative branches within a single process model. However, this often results in tangled processes containing many exceptions. Moreover, if this organization decides to enter, for example, the European market, integration of the new process scenarios will require a lot of efforts. The second possibility is to identify and support several modeling (or abstraction) levels for the process while maintaining the relations between those levels [22].

In DeCo, as discussed in [24], we make a distinction between a *configurable process definition* (in [24] is called a configurable reference process model) resulted from the process design, and a *customized process design specification* (according to [24], model configuration or individualization) reflecting the concrete environment where the process enactment supposed to be observed.

**4.2.2. Deployment vs. Execution.** For descriptive processes, multiple process details often cannot be specified during the customization either. The emerging case-related knowledge and the customer-organization communication patterns can vary widely within the same predefined deployment environment affecting the concrete process scenario. On the other hand, they can be identified only during the execution. For example, during the mortgage approval process, a concrete applicant situation may affect the list of required documents and, consequently, the required revision activities: the form W-2 (Wage and Tax Statement) is typically requested from all the applicants in USA, whereas the form 2555 (Foreign Earned Income) should be provided by taxpayers who have earned income from sources outside the United States. The applicant may prefer to communicate with his/her loan consultant by e-mail or by setting up personal meetings – this can also affect the whole process scenario. Along those lines, a current situation on the market, a property appraisal report, or other emerging data may require a special treatment and result in various courses of actions and decisions.

Therefore, in our case, the customized design specification should remain *configurable* and allow for the “fine tuning” at execution. Thus, we make a distinction between a customized design specification of the process and its occurred realization that reflects the decision making and dynamic process configuration during its single invocation. To document such process realizations we define the third type of DeCo specifications – the *process realization (or trace)*. The main purpose of the process traces is to construct the

knowledge base and to contribute into further process improvement (e.g. using process mining [16]).

### 4.3. DeCo process life cycle

We define 4 process lifecycle phases in DeCo: Design, Customization, Execution, and Analysis. Figure 2 illustrates the lifecycle and its phases.

**4.3.1. Design Phase.** During this phase, the process goal, a set of activities/tasks that *supposed* to lead to this goal, the contracts for each individual activity (pre, post, inv), and data required to fulfill these contracts have to be identified. The following questions can characterize the design phase:

- What can characterize the process success or failure?
- What are the tasks that must/should/could be done to achieve the desired result?
- What data artifacts are required (mandatory) for each task? Are there any alternatives?
- What are the available solution patterns?
- Who can make a decision? Are there any alternatives?

Design phase is terminated with a non-deterministic design specification that defines process invariants. It focuses on what *can be done* to accomplish the process and *what is needed* for it.

**4.3.2. Deployment/Customization Phase.** At deployment, the process environment is selected. Therefore, the initial design specification can be customized reflecting the emerging information. This customization can involve: specification of the set of available roles, role/tasks assignments, partial ordering of tasks, definition of business rules etc. In order to respect the emerging constraints, new tasks can be added whereas some activities/tasks defined at design can be removed or some contracts can be precised. The following questions can characterize the customization phase:

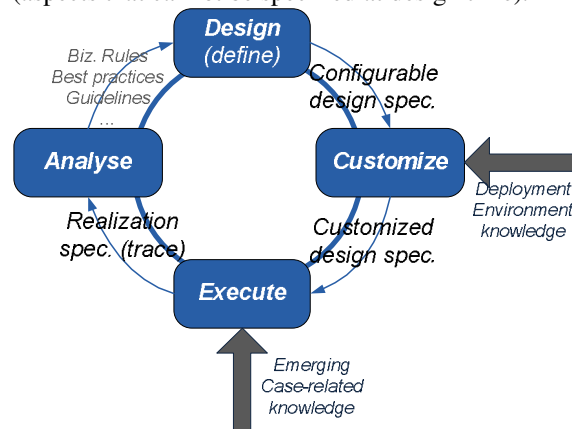
- What are the constraints/norms to comply with? Their nature (e.g. organizational, geo-political, physical, temporal, cultural, etc.)?
- What are the tasks that must/should/could be done to ensure the compliance?
- What data artifacts are required?
- What data artifacts are available to carry out the process?
- What are the accepted/recommended solution patterns? Are there any alternatives?
- Who should make a decision? Are there any alternatives?

Customized design specification focuses on *what the organization has to do* to accomplish the process in a *given environment* and what *will be available* for it to do so.

**4.3.3. Execution Phase.** Properly defined, a customized design specification enables configuration of the process enactment at run-time, supporting the



decision making about the process handling based on the emerging data. These decisions may include: triggering, canceling, re-execution, delegation of tasks, restricting or relaxing decision rules, initiating a supplementary investigations or information requests, assigning actors, etc. These decisions can be driven by the business rules and explicit regulations as well as by the actor experience, process history, best practices, etc (aspects that cannot be specified at design time).



**Figure 2. DeCo process lifecycle.**

As explained in 4.1.1, the process specification in DeCo represents a set of tasks with no explicit execution scenario: at run-time, a task can be dynamically selected from the list of tasks enabled at a given case state. As soon as the task is triggered, it becomes a part of the process enactment *trace*.

**4.3.4. Analysis Phase.** Upon each process enactment termination one possible process path (trace) is explored and can be stored in the history record. Such Case history can be studied in order to generalize the repetitive scenarios and/or derive business rules or guidelines. These guidelines and rules can be further incorporated into initial or customized design specifications making the process more deterministic.

## 5. DeCo modeling notation

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 refinement. This notation is in its infancy. With DeCo, we introduce the following features to the BPMN graphical notation:

- Action contracts, including preconditions, postconditions and invariants (denoted by 'Pre', 'Post', 'Inv' respectively);
- Distinction between cross-boundary data objects (denoted by 'IN' / 'OUT') and local data objects;

- Optional vs. obligatory tasks/data/conditions (depicted by dashed or solid lines respectively);
- Configurable vs. fixed tasks/data/actors (depicted by bold or standard lines respectively);
- Configuration rules (if explicit);
- Task/role/data object specialization (denoted by 'S');
- Task/role/data object alternatives (denoted by 'A').

### 5.1. Configurable design specification

Figure 3 illustrates the configurable design specification of the Formal Application phase of the mortgage approval process.

The **SendForms** task in Fig.3 is specified with a precondition "*request is received*" and a postcondition "*forms are sent*". These conditions can be formalized as predicates in first-order logic. This formalization is beyond the scope of this paper. Each condition is depicted as a link between the task and a data object. The link can be annotated with a condition expression in text or FOL.

**PayStubs**, **CCBills**, **Confirmation** are examples of cross boundary data objects. **TaxForms** and **TaxReturn** are configurable data objects (**R1** and **R3** are the corresponding configuration rules). **BalanceSheet** is an optional configurable data object: this form is demanded from the applicant only "*if the applicant is self-employed*". This fact is documented in the rule **R2**.

**GetApplicationFee** is an optional task that can be decided upon in each particular case. **SendForms** is an optional configurable task, i.e., if selected, it can be specialized: **SendFormsByPost** **SendFormsByFax** **SendFormsByE-mail**. Similarly one can configure the **LockInAnInterestRate** task. Apart from **SendForms** and **RegisterApplication**, other tasks are not preordered.

The **MortgageLender** role is a configurable role that can be further specialized and split into multiple roles (depending on the organization).

### 5.2. Customized design specification

At deployment, the design specification is customized with respect to the process deployment environment. Figure 4 illustrates the customized design specification of the formal mortgage application that has been obtained from the initial design specification (Fig.3) for the concrete financial institution - **Bank X**: The **SendForms** task is removed as **BankX** provides all the mortgage application forms on-line. The **RegisterApplication** task is configured as follows: the **TaxForms** data object from **Pre 1.2** is specialized: the **W-2 form** must be provided by the applicant; the **TaxReturn** data object is specialized and replaced by the **1040 form**. The 'A' in the top right corner of this object denotes that this object has alternatives (i.e. possible replacements): at run-time the form **1040** can be replaced

by the simplified version of this form – **1040EZ**. Generic actor **MortgageLender** from the design specification is specialized and represented now by two actors: **Processor** and **LoanCunsultant**. The tasks are distributed between

these actors. **LockInAnInterestRate** task is specialized to “by fax”, however this task has alternatives (e.g. it is possible to lock in the interest rate by phone).

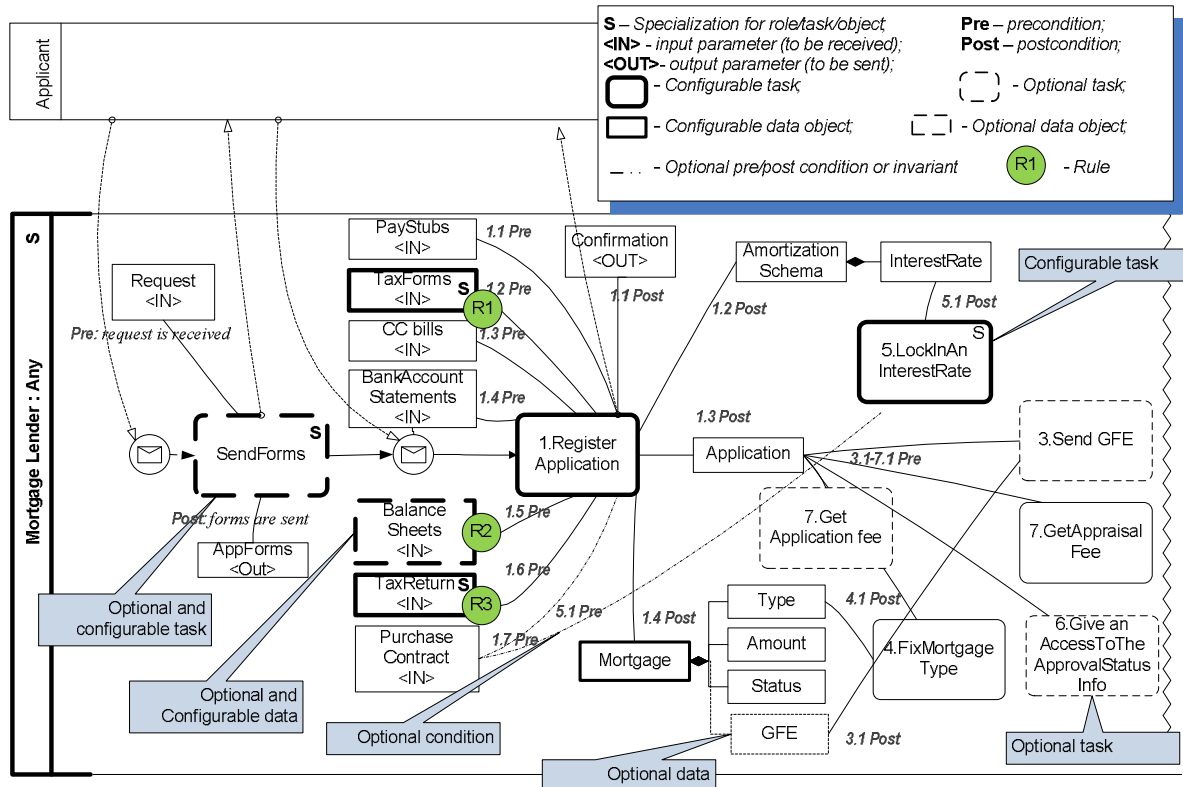


Figure 3. Configurable design specification.

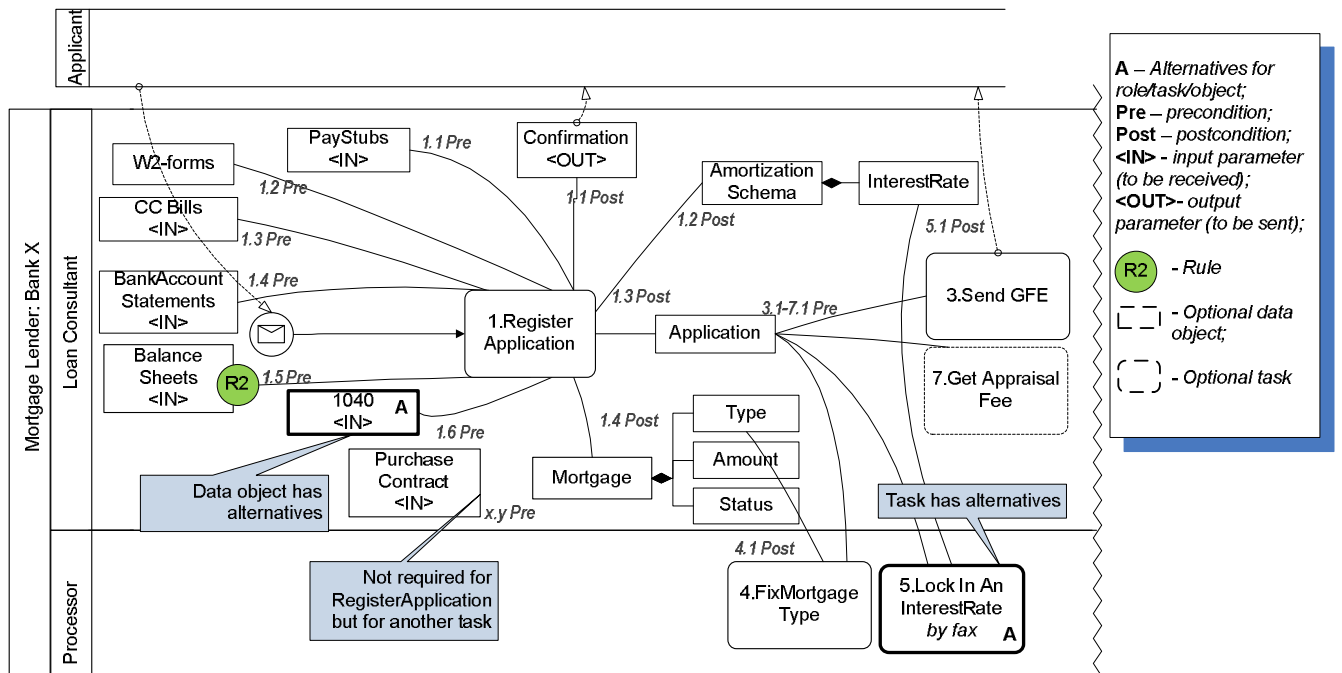
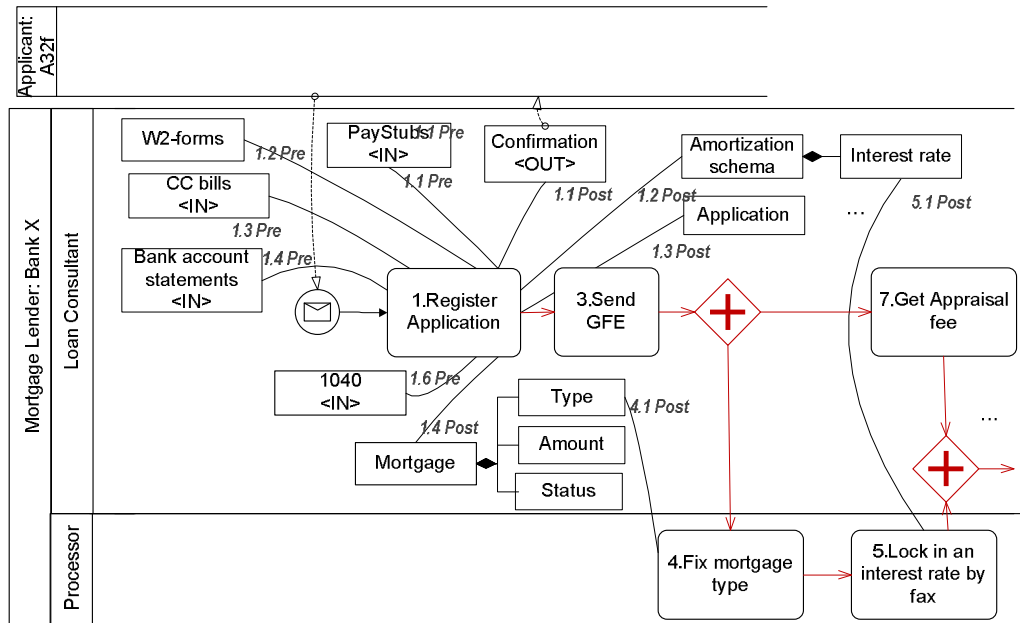


Figure 4. Customized design specification.





**Figure 5. Realization specification (trace).**

This customization of the design specification reduces the nondeterminism of the latter and provides more concrete details of the process execution as defined for the **BankX**, however, it rests configurable and non-deterministic (tasks are not preordered).

### 5.3 Trace

At the execution, the process is configured “on the fly”. Figure 5 illustrates a process trace that reflects a single invocation of the mortgage approval/formal application process customized in the previous section. This specification depicts the data objects received and the selected task ordering.

Collection of the process traces can be used for further process analysis and improvement. In order to prove that the trace is aligned with the corresponding design specifications, the relations between those specifications should be formalized as refinement relations and refinement should be validated. These formalization and process are beyond the scope of this paper.

## 6. Conclusion

In the academic classification, case management processes can be related to descriptive, knowledge-intensive, or unstructured processes. Until recently, those process types have not been recognized as a part of BPM. This partially explains why a majority of modeling

formalisms presented on the BPM market today fail in providing an appropriate level of adaptability while ensuring the process validation and control for descriptive, knowledge-intensive processes.

Traditional, activity-driven formalisms for business process modeling, such as BPMN, encourage the early specification of process details (at design-time), whereas for knowledge-intensive processes, such as case management, this is not possible. Thus, multiple modeling levels (and corresponding specifications) for knowledge-intensive processes are needed.

In this work, we presented the modeling formalism based on Declarative Configurable specifications (DeCo) that includes three types of specifications. *Configurable design specification* focuses on the process goals independently from the process deployment environment. Design specification is non-deterministic and can be configured for various environments upon the process deployment. *Customized design specification* reflects the knowledge about the process deployment environment (e.g. governmental norms and regulations, customer behavior patterns, etc). It can be obtained by refinement of the design specification. In general case, this specification is non-deterministic and supports run-time adaptability driven by emerging knowledge. *DeCo realization specification* (or process trace) reflects one actual process invocation. Aggregated history of process traces is beneficial for further process analysis and improvement.

The proposed approach allows the BPM specialists to separate the process goals from the process means and, finally, from the concrete process realization details improving the adaptability.

While exposing a process to a complex environment, it is important to ensure that the constraints imposed by this environment are feasible and that the process goal can eventually be achieved. This problem will be addressed in our future works that will be focused on the alignment between DeCo specifications.

## References

- [1] J.B. Hill, B. J. Lheureux, E. Olding, D. C. Plummer, B. Rosser, J. Sinur. Predicts 2010: Business Process Management Will Expand Beyond Traditional Boundaries, available on Gartner's website at <http://www.gartner.com/resId=1231219>
- [2] Object Management Group, Case Management Process Modeling (CMPM) Request For Proposal: Bmi/2009-09-23
- [3] J.B. Hill, Five Predictions for How BPM Will Evolve. available on-line at <http://www.documentmedia.com>
- [4] T.H. Davenport and L. Prusak. Working Knowledge: How Organisations Manage What They Know. Harvard Business Press, (1998)
- [5] T.H. Davenport and J.E. Short. The New Industrial Engineering: Information Technology and Business Process Redesign. Sloan Management Review (1990)
- [6] I. Nonaka and H. Takeuchi The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University press (1995)
- [7] I. Rychkova, Formal semantics for refinement verification of enterprise models. Dir.: Alain Wegmann. Thèse EPFL, no 4210 (2008).
- [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, (2008)
- [9] G. Regev, and A. Wegmann, Regulation Based Linking of Strategic Goals and Business Processes, Proceedings of the 3rd BPMDS Workshop, London, (2002)
- [10] Object Management Group, Business Process Model and Notation (BPMN) FTF Beta 1 for Version 2.0 (2009)
- [11] Dellen, B., Maurer, F., and Pews, G. Knowledge Based Techniques to Increase the Flexibility of Workflow Management. Data and Knowledge Engineering, North Holland. (1997)
- [12] Nurcan, S. Business Process Modeling for developing Process Oriented IT Systems, Int. IRMA Conf., USA. (2004)
- [13] N. Gronau and E. Weber, Management of Knowledge Intensive Business Processes. In Business Process Management: Second International Conference, BPM 2004 Potsdam, Germany, Springer, (2004)
- [14] O. Marjanovic and R. Seethamraju, Understanding knowledge-intensive, practice-oriented business processes, in Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS-41), (2008).
- [15] 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)
- [16] C.W. Günther, S. Rinderle, M. Reichert, and W.M.P. van der Aalst. Change Mining in Adaptive Process Management Systems. In R. Meersman and Z. Tari et al., editors, On the Move to Meaningful Internet Systems 2006, OTM Confederated International Conferences, 14th International Conference on Cooperative Information Systems. Springer-Verlag, Berlin, (2006)
- [17] W.M.P. van der Aalst. Challenges in Business Process Mining. BPM Center Report BPM-10-01, (2010)
- [18] N. Wirth, "Program development by stepwise refinement", Communications of the ACM. (1971)
- [19] M. Rosemann and W. M. P van der Aalst. A Configurable Reference Modelling Language. Information Systems, (2007)
- [20] D. Jackson, "Software Abstractions: Logic, Language, and Analysis", MIT Press. Cambridge, MA. (2006)
- [21] 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, Germany. (2008)
- [22] I. Rychkova, G. Regev and A. Wegmann, Using Declarative Specifications In Business Process Design.. International Journal of Computer Science & Applications. (2008)
- [23] 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)
- [24] 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 Proceedings 27th International Conference on Conceptual Modeling (ER 2008).
- [25] Mendling, Jan and La Rosa, Marcello and ter Hofstede, Arthur H.M. Correctness of Business Process Models with Roles and Objects. Elsevier (2008)
- [26] S. Jablonski and C. Bussler. Workow Management: Modeling Concepts, Architecture, and Implementation. International Thomson Computer Press (1996)
- [27] WfMC-TC-1011 v3 Workflow Terminology & Glossary (1999).