PhD thesis proposal

Exploring Formal Methods and Formal Concept Analysis for Agile Business Process Management

Funding: French Government Research Grant **(gross income: between 1850 € and 2000 € per month).**
Location : Université Paris 1 Panthéon-Sorbonne, Centre de Recherche en Informatique, 90, rue de Tolbiac 75013 Paris, France
Directors : Irina RYCHKOVA (irina.rychkova@univ-paris1.fr) and Bénédicte LE GRAND (Benedicte.Le-Grand@univ-paris1.fr)
Starting date: between September and November 2013
No application will be considered after 01/07/2013
Keywords : Formal specification, model checking, Formal Concept Analysis, Business Process Modeling

Research Subject

Capacity to timely discover and to efficiently respond to rapid changes in the environment is a major goal of an enterprise of the future. According to [4][6], a firm's ability to adapt to dynamic environments depends first on the agility of its business processes. Therefore, design and development of new process management systems enabling process adaptation at run time are essential.

The subject of this PhD thesis lies on **the intersection of three research areas**: **business process modeling, formal methods and formal concept analysis**. We propose to explore formal methods and formal concept analysis (FCA) and to build a novel approach for agile process modelling, simulation and analysis. In particular we propose to apply these techniques for unstructured processes such as case management processes (CMP).

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, etc. CMP can be characterized by the following: it is driven by emergent knowledge about the case subject or the environment; largely based on human expertise; highly unpredictable; difficult to replicate; hard to analyze and improve as no HOWTOs available.

We subdivide the problem on two complementary axes:

1. **Processes Specification for Process Agility**

A business process can be seen as a sequence of events triggered by activities of business process actors. The majority of existing methods for business process design follow *imperative* principles, implying that the order of (internal or external) process events is predefined. At run time, processes follow the configured model with limited possibilities to deviate from the predefined scenario. 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 agility [8][9][10][11].

The first challenge related to this PhD thesis is to find an appropriate (mathematical) formalism for representation and reasoning about case management processes (CMP) while ensuring an appropriate level of agility.

To ensure a greater agility, the shift of the traditional imperative paradigm for process design and exploration of declarative principles is required. In [BPMDS13][RCIS13] the abstract formalism based on finite state machine (FSM) semantics [33] has been proposed. (Other examples of models of computations include Pi-calculus[34], Lambda-calculus[35] etc.) Within this formalism, we put the notion of "activity" implicit; only the events resulting from an activity execution are observable. In [BPMDS], we use the term "navigation" to describe the way a process should be executed.

We suggest that, instead of following a predefined execution scenario, **a process navigates in the process "state space", dynamically adjusting its path based on the current state, current situation and navigation rules.**
At a given moment of time, a process state can be defined by its business status or, more formally, by a set of characteristic variables and their values at this moment of time (e.g. for the patient care process, we can define the abstract states "admitted", "inDiagnostics", "inTreatment", "discharged". Each of these states can be defined by a set of values characterizing patient's condition: body temperature, blood pressure, etc.)
Based on this formalism, initial navigation rules for process guidance based on Formal Concept Analysis and Galois lattices [36][37] are defined. We specify the resulting process as a set of activities that can be dynamically assembled at run time into one of the (non-forbidden) process scenarios.

The research questions for this research axis are the following:

- What is an appropriate formalism for (case management) process modelling that supports agility? Which model of computation suits best for CMP? (FSM, LTS, others?) What is the appropriate abstract syntax?
- What type of semantics to define (denotational, axiomatic, operational)? How?
- What is the appropriate concrete syntax and (visual) modelling notation? (BPMN-like? Other?)
- Which perspectives to distinguish (designer's, user's, manager's, analyst's, engineer's)?

2. Processes Simulation and Analysis

Once the process formalism is chosen and the process can be described or documented, the next step is to simulate, analyse and improve the process (as traditional workflow-based approaches do). **The second challenge related to this PhD work is to explore the opportunities provided by automated model checking, theorem proving and formal concept analysis for process model validation and for guided process execution**.

The research questions for this axis are the following:

- How to simulate a declarative process specification? How to interpret the results? How to communicate the results back to the domain specialists? How to establish a feedback loop?
- What are "navigation rules"? How to define/formalize/communicate them? How to choose the "right" action? What the "right" action actually means? Metrics of success or "rightness"?

- Path finding: Given that the "destination" for a process it is its goal, how to calculate the "most favourable" path to this destination? How to navigate when the main destination is not achievable?
- What kinds of analysis/validation can be required? Which techniques to apply?

Scientific background

Business Process Modeling

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. These and other dominant business process and workflow supporting formalisms are almost systematically activity oriented (or *imperative*): 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 case management process modeling (CMPM) in particular.

Formal Methods

In computer science, formal methods are a particular kind of mathematically based techniques for the specification, development and verification of software and hardware systems[14]. The examples of formal methods include Z notation [21][22][23], B-method [24], Alloy specification language [18], etc.
Formal methods can be used for **step-wise system design**: they provide a formal specification of the system to be developed at different levels of details and allow for accurate refinement (transition from one level to another). The resulting formal specification can be used to **verify that the requirements** for the system being developed have been completely and accurately specified.
There are two main approaches to formal verification: model checking [25] and a theorem proving based on logical inference [26]. These approaches are proven very efficient in design and specification of safety-critical system. However, due to their complexity, approaches based on a formal semantics, model checking and verification using theorem proving are rarely used outside the technical domain.

The claim of workflow-based, process-driven or other imperative design methodology proponents is that the imperative specifications can be validated, analyzed and controlled, assuring stable performance and predictable results. Similar results, however, can be assured by providing formal semantics for "unpredictable" declarative models and, eventually, applying the formal methods for process model verification.

Formal Concept Analysis

FCA is a mathematical theory relying on the use of formal contexts and Galois lattices. The use of Galois lattices to describe a relation between two sets has led to various classification methods: a Galois lattice gathers elements, which have common properties in clusters, called formal concepts. A partial order exists among these concepts, which form a lattice with an upper and a lower bound. Since their creation, Galois lattices have been used in numerous contexts to extract hidden knowledge from data.

Formal concept analysis provides a universal tool for clustering the objects as well it can be used as underlying semantics for a recommenders system, providing a selected subset of elements that correspond to a certain criteria. Its joint use with formal methods and context modelling is very promising in the context of agile business management.

Application domain: Case Management Process

Davenport [12], [12] defines case management process as a process that is not predefined or repeatable, but instead, depends on its evolving circumstances and decisions regarding a particular situation, a case. He discusses the need in specific approaches to handle such processes. The Case Management Process Modeling (CMPM) Request For Proposal released by OMG on September 2009 [4] expresses the particular demand of practitioners in the case management solutions. OMG defines case as "A situation, set of circumstances or initiative that requires a set of actions to achieve an acceptable outcome or objective.…". A 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.

Requested Work

The candidate will have to carry out the work in the following three directions:

A. State of the art
B. Development of a methodology for agile process modelling simulation and analysis
C. Definition of Formal semantics, Concrete syntax, techniques for Simulation and Analysis.
D. Implementation of this methodology: developing a prototype
E. Experimenting with this prototype: developing and working on the case studies

Requirements

The candidate must hold a M.Sc. in Computer Science (or equivalent). He/She must have very good programming skills (C++ or Java), solid skills in formal methods (Alloy is a preference) and model checking / theorem proving techniques. The candidate should have good writing skills in English. He/She must be highly motivated, independent, with a real ability to organize and follow a schedule.

To apply: send a detailed CV (resume) (in English), a motivation letter (in English), copy of official transcript of student record (B.Sc and M.Sc) and letters of reference to Irina Rychkova (irina.rychkova@univ-paris1.fr)

Références

1. 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)
2. G. Regev, and A. Wegmann, Regulation Based Linking of Strategic Goals and Business Processes, Proceedings of the 3rd BPMDS Workshop, London, (2002)

3. 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
4. Object Management Group, Case Management Process Modeling (CMPM) Request For Proposal: Bmi/2009-09-23
5. Raschke, R. L., David, J.S.: Business process agility. (2005).
6. Bider, I., Johannesson, P., Perjons, E.: Do workflow-based systems satisfy the demands of the agile enterprise of the future? In: Rosa, M.L., Soffer, P. (eds.), Business Process Management Workshops (BPM 2012), Lecture Notes in Business Information Processing, vol. 132, pp.59-64,., Springer (2013)
7. Lamport, L.: Time, clocks, and the ordering of events in a distributed system. Communications of the ACM, vol. 21, no. 7 (1978), pp. 558-565.
8. S. Nurcan, M.-H. Edme. Intention Driven Modelling for Flexible Workflow Applications. Special issue of the Software Process: Improvement and Practice Journal on "Business Process Management, Development and Support", (2005)
9. 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).
10. 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)
11. 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)
12. Davenport, T., Nohria, N.: Case management and the integration of labor. Sloan Management Review 35 (1994) 11–11
13. Davenport, T.: Thinking for a living: How to get better performances and results from knowledge workers. Harvard Business Press (2005)
14. vom Brocke, J.HKVJH & Rosemann, M. (2010), Handbook on Business Process Management: Strategic Alignment, Governance, People and Culture (International Handbooks on Information Systems) (Vol. 1). Berlin: Springer
15. R. W. Butler (2001-08-06). "What is Formal Methods?". Retrieved 2006-11-16.
16. Daniel Jackson and Jeannette Wing, "Lightweight Formal Methods", IEEE Computer, April 1996
17. Vinu George and Rayford Vaughn, "Application of Lightweight Formal Methods in Requirement Engineering", Crosstalk: The Journal of Defense Software Engineering, January 2003
18. [7] Jackson, D.: Software Abstractions- Logic, Language and Analysis. MIT Press (2011)
19. Richard Denney, Succeeding with Use Cases: Working Smart to Deliver Quality, Addison-Wesley Professional Publishing, 2005, ISBN 0-321-31643-6.
20. Sten Agerholm and Peter G. Larsen, "A Lightweight Approach to Formal Methods", In Proceedings of the International Workshop on Current Trends in Applied Formal Methods, Boppard, Germany, Springer-Verlag, October 1998
21. Abrial, Jean-Raymond; Schuman, Stephen A; Meyer, Bertrand (1980), "A Specification Language", in Macnaghten, AM; McKeag, RM, *On the Construction of Programs*, Cambridge University Press, ISBN 0-521-23090-X (describes early version of the language).
22. "ISO/IEC 13568:2002". *Information Technology — Z Formal Specification Notation — Syntax, Type System and Semantics*. ISO. 2002-07-01. 196 pp
23. "ISO/IEC 13568:2002/Cor.1:2007" (PDF). *Information Technology — Z Formal Specification Notation — Syntax, Type System and Semantics — Technical corrigendum 1*. ISO. 2007-07-15. 12 pp.
24. *The B-Method: An Introduction*, Steve Schneider, Palgrave Macmillan, Cornerstones of Computing series, 2001. ISBN 0-333-79284-X.
25. Dijkstra, E., Dijkstra, E., Dijkstra, E.: Notes on structured programming. Technological University, Department of Mathematics (1970)
26. Gordon, M.J.C., Melham, T.F., eds.: Introduction to HOL: a theorem proving environment for higher order logic. Cambridge University Press, New York, NY, USA (1993)
27. Holzmann, G.J.: The SPIN Model Checker - primer and reference manual. Addison-Wesley (2004)
28. Jackson, D.: Alloy Analyzer tool. http://alloy.mit.edu/alloy/ (2011)
29. Weidenbach, C., Brahm, U., Hillenbrand, T., Keen, E., Theobald, C.,Topi´c, D.: Spass version 2.0. Automated DeductionCADE-18 (2002)45–79
30. Paulson, L.: Isabelle: A generic theorem prover. Volume 828. Springer(1994)
31. Snook, C., Butler, M.: Uml-b: Formal modeling and design aided by uml. ACM Trans. Softw. Eng. Methodol. 15(1) (January 2006) 92–122

32. Snook, C., Savicks, V., Butler, M.: Verification of uml models by translation to uml-b. In: Proceedings of the 9th international conference on Formal Methods for Components and Objects. FMCO'10, Berlin, Heidelberg, Springer-Verlag (2011) 251–266
33. Plotkin, Gordon D.: A structural approach to operational semantics (1981).
34. Milner, Robin (1999). Communicating and Mobile Systems: The π-calculus. Cambridge, UK: Cambridge University Press.
35. Barendregt, Hendrik Pieter (1984), The Lambda Calculus: Its Syntax and Semantics, Studies in Logic and the Foundations of Mathematics, 103 (Revised ed.), North Holland, Amsterdam.
36. Barbut, M., Monjardet, B. : Ordre et classification. Algebre et combinatoire, Tome 2, Hachette (1970)
37. Birkhoff, G.: Lattice Theory, First Edition, Amer. Math. Soc. Pub. 25, Providence, R. I. (1940).