An User-Friendly Approach to Business Process Formal Verification

Flavio Corradini, Andrea Polini, Alberto Polzonetti, Barbara Re
Computer Science Division – School of Science and Technologies
University of Camerino
firstname.lastname@unicam.it

Abstract

Formal methods can bring many advantages to software practitioners and their adoption has been often advocated. In recent years usage of formal techniques certainly increased, nevertheless there is still ample room for further adoption and a wide gap exists between formal methods experts, with knowledge on formal tools, and domain experts, which possess the knowledge on the problems to solve and on users requirements.

In this paper we introduce a user-friendly approach for Business Process assessment based on formal verification techniques. Starting from a semi-formal notation, well understood and largely used by domain experts, we provide a denotational mapping to a formal specification in the form of a process algebra. This transformation makes possible formal and automatic verification of desired quality requirements. The approach has been already applied, with encouraging results, in the e-government domain to verify the quality of Business Processes related to the delivery of e-Government Digital Services. Moreover the approach is supported by a plug-in for the Eclipse platform permitting to have an integrated environment in which to design the Business Process model and to assess its quality.

Introduction

Starting from domain knowledge, the introduction of an approach and a supporting tool permitting to domain experts to formally and automatically verify a designed Business Process (BP), with respect to the defined requirements, is an interesting challenge to be addressed in many application domains. Different sets of languages to express BPs have been investigated and defined. Among the others Business Process Modeling Notation (BPMN) [4]

is certainly the most used language in practical contexts mainly due to its intuitive graphical notation. Nevertheless BPMN does not have a precisely defined semantic. For these reasons, and in order to permit formal verification, we defined a precise denotational mapping of BPMN elements to CSP processes.

The mapping permits the adoption of formal verification techniques and in particular of model checking. Reachability analysis is applied in order to assert whether the goals, that are directly coded from the requirements specifications, are fulfilled or not. It is worth mentioning that in the specific domain, and given the mapping rules we defined, the well know state explotion phenomenon does not seems to emerge.

The approach has been already applied to real case studies in the e-government domain to assess the quality of e-Government Digital Service (GDS) delivery processes [1]. Using the approach BP experts were able to improve their BP making them more acceptable to citizens.

BP Formal Verification

Our work aims at providing to BP and domain experts the power of formal verification techniques still allowing the usage of graphical notation with which they are already acquainted. The approach relies on the following three main steps: (i) BP *specification* and domain requirements *selection* via a user-friendly interface; (ii) Automatic *mapping* of the BP specification and of the set of requirements to CSP processes and to a set of goals, respectively; (iii) Formal *verification* of mapped processes with respect to the specified set of properties (goals). In case the verification phase ends highlighting some problems, i.e. at least one of the requirements selected by the domain experts has been violated, the design process should be restarted taking into account the result of the verification step.

The mapping we have defined covers all the core BPMN elements and almost all those introduced by the OMG notation. Few elements dealing with transactions, such as compensation events and cancel events, or time have been kept outside of our mapping. Main reason for this choice is that they are seldom used in practice [5] and most of the design tools do not support them. Moreover we defined additional constraints on the definition of the BP both to make the verification step easier, and to make BP specifications clearer. Tasks as well as messages can be typed to support specific domain-dependent characterizations. Moreover each introduced task can support at most one communication type (send or receive) but not both. In this way we ask the BP designer to explicitly provide the order of messages exchange, that otherwise would be undefined. Loops can be introduced only using the specific BPMN elements. Implicit loops are not admitted. Somehow this

constraint bans the usage of `goto` statements from the specification. Finally, collapsed sub-process are not supported. For each sub-process BPMN `end` and `start` events have to be explicitly provided since they support the trigger of elements inside the sub-process.

BPs modeled according to the defined constraints can be mapped into CSP models to be successively processed by the model checker. The mapping has been defined according to the following general principles. (i) Each BPMN graphical element included within a pool is formally represented by a CSP process or a parallel execution of the generated CSP processes - we will name such process *Element CSP*. (ii) Each pool is mapped to a parallel composition of *Element CSP* processes with barriers synchronization. In this case no message exchange will be observable - we will name such process *Private CSP*. (iii) The whole process results from the parallel execution of the *Private CSP* processes including their interactions implemented via messages exchange - we will name such processes *Abstract CSP*. Due to lack of space we do not report the mapping rules we have defined. A wider discussion on the mapping can be found in [2].

Domain related quality requirements are generally defined by domain experts. Here our contribution is on the codification of such domain knowledge within a tool defining a set of property templates. These general properties should be satisfied by any process in the given domain. They constitute a checklist that can be used during the design of any BP. In order to actually verify some of the defined properties, we directly intervened on the mapping rules to add statements expressely related to the domain requirements or to verification functionalities. Thus the derived checklist hides a set of assertions that can be assessed thanks to the addition of global variables within the mapping rules. Each mapping rule influences the verification of a property redefining a global variable that is successively combined with other global variables to check the whole assertion. This is what we did in our case exploiting the appropriateness of the approach in e-government domain [1].

Technical Details

The formal verification approach is supported by a plug-in available for the Eclipse Framework. In particular our plug-in is integrated in an Eclipse extension such as the BPMN modeler, and it uses the functionalities of the PAT model checker [3]. The CSP model is derived taking advantage of the Eclipse Modeling Framework (EMF) which is a powerful mechanism made available in the Eclipse platform to define meta-models. EMF, together with other frameworks enabling the graphical rendering of the BPMN constructs, is at the base of BPMN modeler. Therefore through EMF, and the API it makes available, it is possible to interact with the defined BPMN model

to obtain the list of elements which have been included within a BPMN specification. In this way it is possible to implement a simple parser that for each BPMN element generates the corresponding CSP code, using in our case the syntax of the PAT model checker, and according to the mapping rules. Similarly the code generation includes the specification of variables enabling the checking of relevant quality requirements. After that the verification step can take place. To make also this step easier we integrated the PAT model checker within the Eclipse framework. As a result the whole tool-chain is available within a unique IDE.

The current version of the plug-in can be freely downloaded at the BP4PA web page (`http://bp4pa.sourceforge.net/index.html`). It implements verification on the e-government domain requirements that we have already codified to apply the approach to GDS.

Conclusions

In this paper we outline the main elements of the defined user-friendly approach permitting to assess, using formal verification techniques, a BP with respect to a set of defined domain dependent requirements. In such a way domain experts can directly contribute in the design of a BP and at the same time they can take advantage of formal verification techniques.

References

[1] F. Corradini, D. Falcioni, A. Polini, A. Polzonetti, and B. Re. Designing quality business processes for e-government digital services. In *IFIP EGOV 2010*, LNCS, 2010. To appear.

[2] Barbara Re. *Quality of Digital e-Government Services*. PhD thesis, University of Camerino, 2010.

[3] J. Sun, Y. Liu, and J. Song Dong. Model checking CSP revisited: introducing a Process Analysis Toolkit. In Tiziana Margaria and Bernhard Steffen, editors, *ISoLA*, volume 17 of *Communications in Computer and Information Science*, pages 307–322. Springer, 2008.

[4] Stephen A. White and Derek Miers. *BPMN Modeling and Reference Guide Understanding and Using BPMN*. Future Strategies Inc., 2008.

[5] Michael zur Muehlen and Jan Recker. How much language is enough? Theoretical and practical use of the business process modeling notation. In Zohra Bellahsene and Michel Léonard, editors, *CAiSE*, volume 5074 of *Lecture Notes in Computer Science*, pages 465–479. Springer, 2008.