

Application of Automata-Based Programming for Construction of Business Processes Management Systems

Evgeny Andreevich Mandrikov

Vladimir Anatolievich Kulev

Abstract—In this article the problem of business processes modeling languages consolidation in a uniform management system is considered. It will be shown that it is reasonable to use automata-based programs as a base for proposed system.

I. INTRODUCTION

The main objective of business process management systems is the automation of the processes taking place in business [1]. When building software products, working with business processes, there are three basic roles: the end-user, business analyst and developer. Business analyst examines and describes the business process and formulates the requirements for software, the developer implements them in the final product.

Traditional business process management systems are trying to be based on the model created by business analyst, and build executable program on the basis of this. Such a way of automating business processes, has gradually forced to move away to second place due to the fact that making even small changes in the business process logic means time-consuming and expensive reprogramming. As a result, applications are not able to be updated at the desired speed, dictated by changing business conditions and needs of the enterprise [2]. It should also be noted that software products developed in such manner are deeply attached to the specific business processes and end-users. This does not allow software to be rapidly adopted in other enterprises.

There is a trend of replacing traditional way of automating business processes with usage of various domain-specific languages [3], based on the description of specific types of business processes, such as: task management (Issue Tracking, Bug Tracking) [4], document management (Document Management System) [5], etc. Using this approach, business analyst and the developer communicate in one language with a graphical representation of the process. Business analyst is responsible for the graphical representation and should not

deal with technical details of the process. But without these details the business process is not fully defined and thus can not be done, so the developer is responsible for their program implementation.

Most of existing software products allow usage of only one or few languages to describe business processes. This makes it difficult to integrate various business processes that occur on the same enterprise, and force developers to do more work. This article describes the approach to solving this problem by translating business processes descriptions into automata programs [6].

II. BUSINESS PROCESSES DESCRIPTION LANGUAGES

BPEL¹ – a widely used language for describing business processes, designed to work in the web-services environment [7]. It is based on WSDL², and while WSDL allow the use of JavaBeans, the natural choice is web-services.

The deployment of business process in BPEL leads to the publication of a web-service, which is the primary interface for interaction with the process. Variables within the BPEL are XML fragments or XSD³ basic types. BPEL has structures to describe the control flow logic and call other WSDL services. Ultimately, BPEL – a language for describing web-services management business processes.

Another popular language is jPDL, created by the Jboss jBPM project [8]. One of its main purposes is task management. For this the language has special design to allow creating tasks in process. Changes in tasks (such as beginning or end of execution) are also events which business process is driven by. Also jPDL supports asynchronous execution of tasks and business process branching into several parallel threads.

By the example of BPEL and jBPM it can be seen that the existence of domain-specific languages is feasible because of broad spectrum of challenges faced in enterprise automation. But it is also easy to see that these languages have many common features, in particular the representation model of business process in the form of a graph.

E. A. Mandrikov is MSc student at the Computer Technologies Department, Saint-Petersburg State University of Information Technologies, Mechanics and Optics, Saint-Petersburg, Russia (e-mail: mandrikov@rain.ifmo.ru).

V. A. Kulev is MSc student at the Computer Technologies Department, Saint-Petersburg State University of Information Technologies, Mechanics and Optics, Saint-Petersburg, Russia (e-mail: kulev@rain.ifmo.ru).

The research is supervised by A. A. Shalyto, PhD, professor at the Computer Technologies Department, Saint-Petersburg State University of Information Technologies, Mechanics and Optics, Saint-Petersburg, Russia (e-mail: shalyto@mail.ifmo.ru).

¹Business Process Execution Language

²Web Services Description Language

³XML Schema Definition

III. TRANSLATION OF BUSINESS PROCESSES DESCRIPTIONS INTO AUTOMATA PROGRAMS

The proposed approach is to translate business processes descriptions into a system of interacting state machines. They may interact:

- by nesting – a machine nested into one or more states of the other machine;
- by calls – a machine is called on a certain event generated by some transition;
- by messages exchange – a machine receives a message from another one;
- by states – a machine checks the state of another machine.

Figure 1 is an example of the simplest process description in jPDL. This example will show how business process can be translated into automata program.

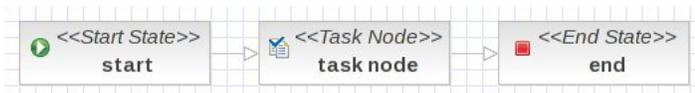


Figure 1. An example of business process description in jPDL

In this example, task node is translated into nested state machine, managing the internal state of the task. This automata program is shown on Figure 2.

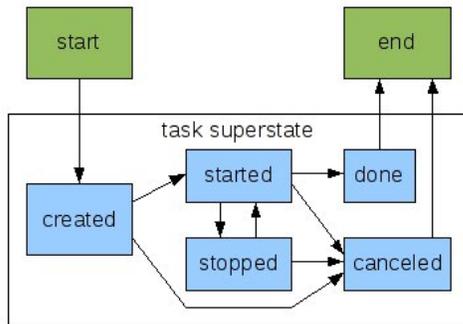


Figure 2. An example of resulting automata program

Similarly, all other high-level business processes languages can be translated into automata programs. Resulting programs can be executed in a single environment for all processes (Process Virtual Machine), or translated into executable code.

IV. CONCLUSION

The benefits of the proposed approach are:

- possibility of a business analyst using conventional business processes modeling software;
- ease of integration of different business processes description languages in a single system;
- extensibility of the system due to possibility of adding new languages;
- possibility of business processes verification (in fact, verification of resulting state machines [9]).

REFERENCES

- [1] “Business process management.” [Online]. Available: http://en.wikipedia.org/wiki/Business_process_management
- [2] V. Kleban and F. Novikov, “Application of finite state machines in the document flow,” *Scientific and technical bulletin of SPbSU ITMO*, no. 53, Automata programming, pp. 286–294, 2008.
- [3] “Domain-specific programming language.” [Online]. Available: http://en.wikipedia.org/wiki/Domain-specific_programming_language
- [4] “Issue tracking system.” [Online]. Available: http://en.wikipedia.org/wiki/Issue_tracking_system
- [5] “Document management system.” [Online]. Available: http://en.wikipedia.org/wiki/Document_management_system
- [6] “Automata programming homepage,” Programming Technologies dpt., SPbSU ITMO. [Online]. Available: <http://is.ifmo.ru/>
- [7] “Business process execution language.” [Online]. Available: <http://en.wikipedia.org/wiki/BPEL>
- [8] “Jboss jBPM.” [Online]. Available: <http://www.jboss.com/products/jbpm/>
- [9] E. Kurbatsky and A. A. Shalyto, “Verification of programs built on automata-based approach,” *Scientific software in education and scientific research. SPbSPU.*, pp. 293–296, 2008.