World Applied Sciences Journal 28 (11): 1809-1814, 2013
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2013.28.11.1879

Rich Petri net with VDM-SL for Business Process Modeling

Majid Aboutalebi and Nona Ghasemi

Department of Computer Engineering, Islamic Azad University, Sari Branch, Sari, Iran

Abstract: Today, business process modeling languages as one of the main components of information systems design are discussed. Many of these languages include limitations and possibilities that they cause only certain types of business process can be modeled. Just some of the business processes modeling languages, can support all of perspectives and few of them have real-time descriptions. This study provides a rich modeling language called PVDM, which integrate Petri net with VDM-SL.

Key words: Business Process Modeling · Formal Methods · Petri nets · PVDM · VDM-SL

INTRODUCTION

In the past decade the use of formal methods to develop reliable systems is greatly increased and accordingly many formal methods have been developed [1], that each of these methods has specific applications such as development concurrent, distributed, real-time systems, etc. The development of software systems based on business process is also considered [2]. The design languages of business process are usually only able to express process or control-flow, while considering the data, resource and task perspectives for business processes modeling is also very important. In this study a business process modeling language (called PVDM) is presented. PVDM integrated Petri net as a modeling tool and formal specification of VDM-SL, such that it can not only display control-flow aspect of the business process but also describe other perspectives.

Business Process Models should have a formal foundation. The two main reasons are: 1) formal models are devoid of ambiguity and 2) formal models increase the potential for analysis [3].

Formal methods are widely recognized as a means to write precise, consistent and unambiguous specifications [4]. VDM have been developed in the late 1970s [5]. It has a recognized international standard and also a comprehensive set of tools supporting it [6]. The corresponding standardized definition language is called VDM-SL [5, 7]. Since it is one of the oldest and well developed formal methods, it has the most history of use in industry [5, 6]. VDM is a model-based method for description and development of sequential software systems [8, 9]. However, VDM does not support the effective definition of concurrent systems.

Petri nets are a graphical and mathematical model for describing the control structures and dynamic behavior of concurrent and distributed systems [10, 1]. The graphical nature and powerful design tool of Petri nets provide a visual communication [11].

There are more than 300 business processes modeling language such that, Petri nets are the core of many of these languages; but Petri nets alone have advantages and disadvantages for business processes modeling. Therefore, new developments of this model and its integration with other models usually are used. Some of the most important advantages and disadvantages are:

Advantages of Petri nets [12]:

- Formal semantics despite the graphical nature.
- State-based instead of event-based.
- Abundance of analysis techniques.

Disadvantages of Petri nets [13, 14, 15]: conflict, livelock, deadlock and confusion.

The benefits of integrating Petri nets with VDM-SL include:

- Improvement of Petri net problems through formal descriptions of VDM-SL.

Corresponding Author: Majid Aboutalebi, Department of Computer Engineering, Islamic Azad University, Sari Branch, Sari, Iran. Tel: +989112917092.

- A unified formal language for modeling different types of business process (sequential and concurrent process).
- Explain and describe the various aspects of business process (data, process, resource and task perspective).
- New theoretical and methodological research problems resulted from the integration.

Related Work: Methods integration is a promising approach for complex software development [16]. Nowadays a lot of works focus on the integration of several approaches in a unique method [17] in order to provide techniques and tools for software development.

Due to the use of visual methods is easier for analysts and users, nowadays many researchers sought to find the combination of graphical methods. One of the best researches is UML 2.0 which is combined with OCL formal language [18].

In [19] an effective integration of the two techniques, Object-*Z* with Timed Automata, with communication mechanisms for designing complex systems is presented.

In [1] a formal method integrating Petri nets with *Z* (called *PZ* nets) for specifying concurrent and distributed systems is presented. The essence, benefits and problems of the integration are discussed.

Gabriel Ciobanu and Maciej Koutny presented a process algebra (called TiMo) in which timeouts of interactions and adaptable migrations in a distributed environment with explicit locations can be specified. Also they provided a structural translation of TiMo into behaviorally equivalent high level timed Petri nets [20].

SOFL is a formal language and method for practical systems development [21]. As a language, it integrates data flow diagrams, VDM-SL and Petri nets to provide a graphical and textural formal notation for specification construction. As a method it is a combination of structured method and object-oriented method.

RAISE [22] which integrates VDM and Petri Nets had also used such approach of integration [16].

PVDM: In this study a new business process modeling language (called PVDM) is presented. PVDM integrates Petri nets and VDM-SL such that Petri net is used to specify the overall structure, control-flow and causal relation of a business process [1]; and VDM-SL is used to define tasks and constrains of the business process.

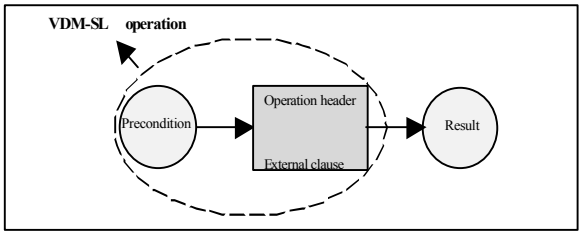

Fig. 1: Petri net elements participating in the VDM-SL operation

For this purpose, for each transition a VDM-SL operation with same name is determined, which formally describe operation of each transition. In addition, places contain required conditions for execution of each transition.

For establishing correspondence between formal descriptions and behavior of a system in designing PVDM, each operation of VDM-SL exactly equivalent with pattern of Fig. 1 of Petri net.

PVDM is a rich Petri net such that has been developed with VDM-SL notations. Based on a contract, notations are followed:

- In each transition, operation header, mode of access to attributes and postcondition, are defined.
- Preconditions are described in the input places.
- The results of execution of each transition are defined in the output places.

As can be seen in Fig. 1, input places of each transition (preconditions) indeed are predicate, which if they be true, its corresponding operation is executed. In addition, output places represent results of execution of operation with a predicate, which indeed provide preconditions for next transitions.

Modeling Steps by PVDM. In system modeling, the identification and representation of an environmental model is important [23]. Modeling by PVDM following steps are suggested:

- The main events are modeled by Petri net.
- A formal operation is defined for each transition.
- Necessary conditions for executing of each transition are described.
- Types, variables, constants, invariant and initialization are determined.

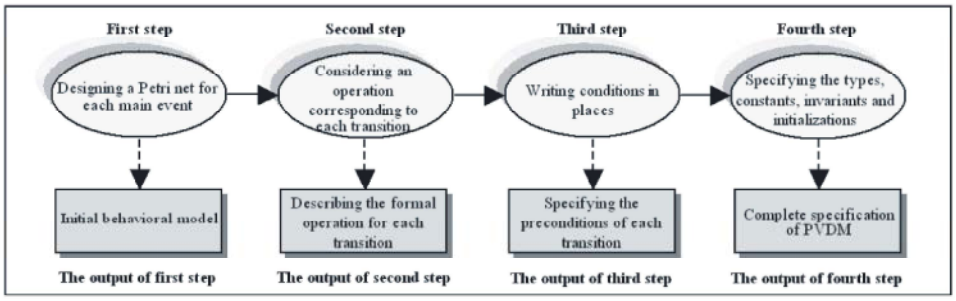

Fig. 2: Shows the above steps

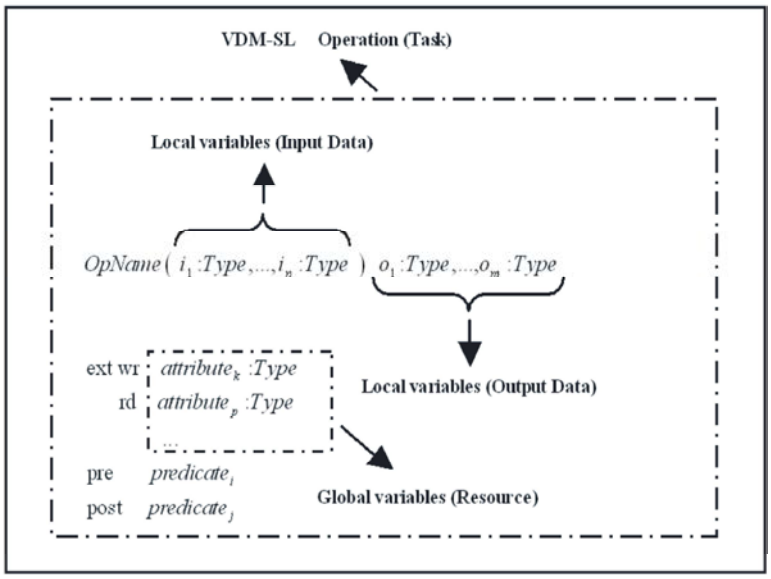

Fig. 3: Presenting information, resource and task perspective in the VDM-SL operation

The Perspectives of Business Process Modeling. The four main perspectives for business process modeling have been introduced as follows:

- Process perspective
- Information perspective
- Resource perspective
- Task perspective

The process perspective describes the control-flow, i.e., the ordering of tasks. The information perspective describes the data that are used. The resource perspective describes the structure of the organization and identifies resources, roles and groups. The task perspective describes the content of individual steps in the processes [24].

PVDM supports each of four perspectives that mentioned above.

Process Perspective in PVDM. Process perspective in PVDM is displayed via graphical structure of Petri net. So control-flow or ordering of tasks in the business process modeling can be easily identified.

Resource Perspective in PVDM. Necessary resources are introduced in the state definition and are called for any of operations. After the operation, disposal resources will be released.

VDM-SL has the capability that comments can be added to complete specifications of system through symbol - - [6]. In PVDM, resources and data are introduced with this symbol, without a compromise to be entered into VDM-SL structure. In each operation are used keyword "*use*" for calling required data and resources in postcondition and keyword "*avail*" for ensuring the availability of resources in precondition. Finally, by keyword "*free*", used resources are released. Thus, in addition to supporting of resources perspective in PVDM, readability of specifications will be increased.

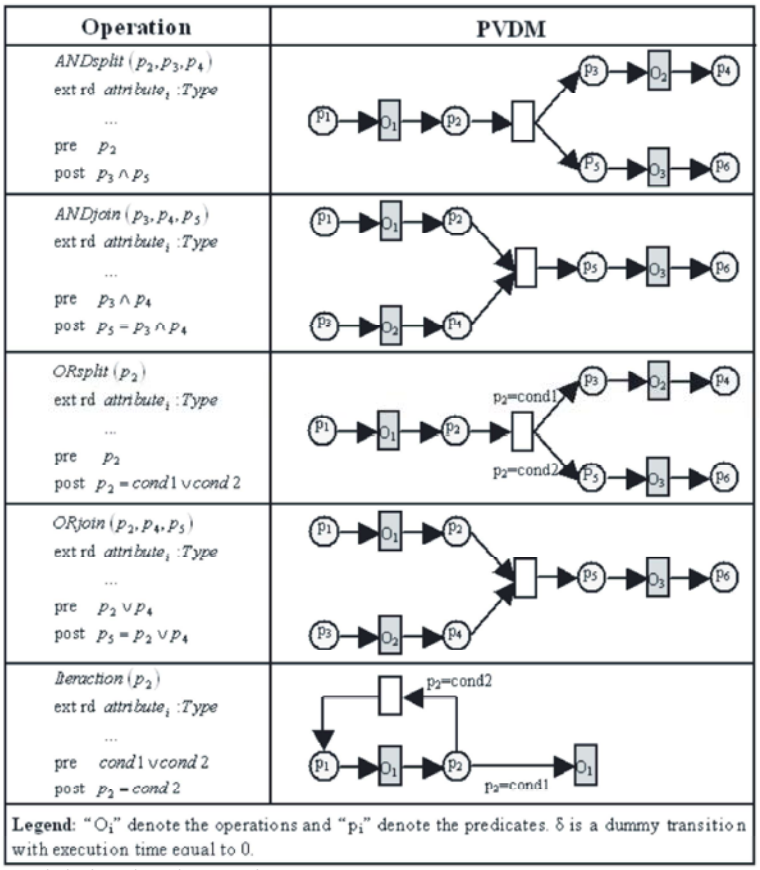

Fig. 4: Specific patterns and their related operations

Information Perspective in PVDM: PVDM variables are divided into two categories:

- Global Variable: these variables are defined in the *state* and for all operations are known. In each operation, global variable with its type is called and its access mode by keyword "*ext*" is determined.
- Local Variable: these variables are also divided into two categories:
- Input Variable: these variables and their types, both of them are defined in the operation header within parentheses.
- Output Variable: these variables and their types, both of them are defined in the operation header outside parentheses.

Local variables only for operation are known that are defined for them. Data or global variables can be considered as resource.

Task Perspective in PVDM. One important features of PVDM is supporting of task perspective. This possibility is provided by definition of the VDM-SL operation for each transition. In other words, content of each element in the control-flow is described as a precise and unambiguous formal description.

Information, resource and task perspective in the standard template for VDM-SL operation are shown in Fig. 3.

Specific Patterns: PVDM can support special patterns, such as AND-split, OR-split, etc. For each transition, a VDM-SL operation is considered, Such that takes places as argument. Preconditions include the predicate of the input places and postcondition is a logical composition of precondition. Patterns [25] and their related operations are shown in Fig. 4.

CONCLUSION

BPMN and Petri nets are used for modeling different types of systems (sequential and concurrent) and can show overall structure of models. VDM is only used for specifying sequential systems, but PVDM creates the possibility that advantages of VDM also be used in

Table 1: Comparison BPMN, PN, YAWL and PVDM

Aspect of comparision	YAWL	PN	BPMN	PVDM
Model-based	+	+	+	+
Number of graphical symbols	14	3	39	3
Supporting of the information perspective	+	+	+	+
Supporting of the process perspective	+	+	+	+
Supporting of the resource perspective	-	-	-	+
Supporting of the task perspective	-	-	-	+
Modeling capability of sequential processes	+	+	+	+
Modeling capability of concurrent processes	+	+	+	+
Modeling capability of real-time processes	-	-	-	+
simplicity	+	-/+	+	+/-
Complete formal description	-	-	-	+

concurrent systems. VDM-SL descriptions are very useful in real-time systems [8, 9]. Hence real-time processes can be modeled by PVDM.

BPMN for process modeling uses more than thirty symbols. This makes difficult for understanding, learning and its analysis. Also YAWL can cover 14 different patterns [26, 27] while PVDM uses only three graphical symbols of Petri net.

BPMN and Petri net just support control-flow perspective and YAWL just supports control-flow and information perspectives while PVDM not only support control–flow, information and resource perspectives but also formally describe tasks.

Comparison BPMN, PN, YAWL and PVDM is shown in Table 1.

REFERENCES

1. Xudong He, 2001. PZ nets: a formal method integrating Petri nets with *Z*, Information and Software Technology, 43: 1-18.
2. Joseph Barjis, 2008. The Importance of Business Process Modeling in Software Systems Design, Science of Computer Programming, 71(1): 73-87.
3. Van Der Aalst, W.M.P., A.H.M. Ter Hofstede and M. Weske, 2003. Business Process Management: A Survey, International Conference on Business Process Management (BPM 2003), volume 2678 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, pp: 1-12.
4. James, D. Kiper and James E. Tomayko, 1998. Techniques for Safety Critical Software Development, IEEE.
5. Dines Bjorner, 2000. Pinnacles of software engineering: 25 years of formal methods, Annals of Software Engineering, 10: 11-66.
6. Quentin Charatan and Aaron Kans, 2004. Formal Software Development From VDM To Java, First published, New York, Palgrave Macmillan, ISBN 0-333-99281-4.
7. Juan Bicarregui, Brian Matthews, Brian Ritchie and Sten Agerholm, 1998. Investigating the Integration of two Formal Methods, Formal Aspects of Computing, 10: 532-549.
8. Kryvyi, S.L. and L. Ye, 2003. Matveyeva, Formal Methods of Analysis of System Properties, Cybernetics and Systems Analysis, 39(2).
9. Ian Summerville, 2006. Software Engineering, Eighth Edition (ISBN 13: 978-0-321-31379-9, ISBN 10: 0-321-31379-8), China Machine Press.
10. Tadao Murata, 1989. Petri Nets: Properties, Analysis and Applications, IEEE, 77(4): 541-580.
11. Khodakaram, Salimifard and Mike Wright, 2001. Petri net-based modeling of workflow systems: an Overview, European Journal of Operational Research 134: 664-676.
12. Van Der Aalst, W.M.P., 1996. Three Good Reasons for Using a Petri-net-based Workflow Management System, In S. Navathe and T. Wakayama, editors, Proceedings of the International Working Conference on Information and Process Integration in Enterprises (IPIC'96), pp: 179-201, Camebridge, Massachusetts.
13. Gianfranco Balbo, Jorg Desel, Kurt Jensen, Wolfgang Reisig, Grzegorz Rozenberg and Manuel Silva, 2000. Petri Nets 2000, 21ST International Conference on Application and Theory of Petri Nets, Aarhuse, Denmark, June 26-30, 2000.
14. Walter Nauber, 2010. Design and Analysis with Petri Nets, Institute for Theoretical Computer Science.
15. Ho, A., S. Smith and S. Hand, 2005. On deadlock, livelock and forward progress, Technical Report UCAM-CL-TR-633, University of Cambridge, Computer Laboratory, May 2005, Information on http://www.cl.com.ac.uk/TechReports/.
16. Christian Attiogbe, 2000. Formal Methods Integration: Some Locks and Outlines, RR 00.8 (December 1998, Revised: December 1999, January 2000, July 2000).
17. Eerke A. Boiten, John Derrick and Graeme Smith, 2004. Integrated Formal Methods, 4th International conference, IFM 2004, ISBN 3-540-21377-5, Springer-Verlag.
18. UML 2.0 OCL Specification, Information on *http://www.omg.org/espec/OCL/2.0/PDF/*.

19. Dong, J.S., R. Duke and P. Hao, 2005. Integrating Object-Z with Timed Automata, Proceedings of the 10th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'05) 0-7695-2284-X/05, IEEE, 2005.
20. Gabriel Ciobanu and Maciej Koutny, 2011. Timed Mobility in process algebra and Petri nets, The Journal of Logic and Algebraic Programming, 80: 377-391.
21. Shaoying Liu, A. Jeff Offutt, Chris Ho-Stuart, Yong Sun and Mitsuru Ohba, 1998. SOFL: A Formal Engineering Methodology for Industrial Applications, IEEE Transactions on Software Engineering, 24(1): 337-344, Special Issue on Formal Methods, January 1998.
22. The RAISE Language Group, The RAISE Development Method, BCS Practitioner Series, Prentice Hall, 1995.
23. Yourdon, E., 1989. Modern Structured Analysis, Prentice-Hall, Englewood cliffs, NJ.
24. Van Der Aalst, W.M.P., 2002. Making Work Flow: On the Application of Petri nets to Business Process Management, In J. Esparza and C. Lakos, editors, Application and Theory of Petri Nets 2002, volume 2360 of Lecture Notes in Computer Science, pp: 1-22. Springer-Verlag, Berlin.
25. Ming Dong and F. Frank Chen, 2005. Petri net-based workflow modeling and analysis of the integrated manufacturing business processes, springer, Int J Adv Manuf Technol., 26: 1163-1172.
26. Van Der Aalst, W.M.P. and A.H.M. Ter Hofstede, 2005. YAWL: Yet Another Workflow Language, Information Systems, 30(4): 245-275.
27. Arthur H. M. Ter Hofsted, Wil M.P. Van Der Aalst, Michael Adams and Nick Russell, 2010. Modern Business Process Automation, YAWL and its support Environment, ISBN 978-3-642-03120-5, Springer-Verlag Berlin Heidelberg.