Why workflow is NOT just a Pi-process

Wil M. P. van der Aalst

Recently, a paper with the title, "Workflow is just a Pi process," was posted on the BPTrends website. Earlier versions of the paper were distributed through various mailing lists (OASIS, BPEL TC, W3C WS-CHOR, BPMI-INFO), and, in the current version of the paper, Smith & Fingar state, "The draft and its catchy title generated controversy among workflow experts in emails and various response documents. This version reflects feedback from those who responded with valuable criticism and insight." In this brief response, I dare to question this since several inconsistencies pointed out by myself and others still remain. Therefore, I provide another response.

Before criticizing the paper, I would like to emphasize that Pi calculus, as Robin Milner and others have developed it, is a solid foundation for modeling and analyzing processes. However, I'm not convinced that the features present in Pi calculus are vital for Business Process Management (BPM) systems. The only thing which distinguishes Pi calculus from classical process algebras like CCS, CSP, and ACP is the notion of mobility. For some application areas, this feature is very useful. However, for BPM solutions it seems less relevant; anything that can be expressed in terms of Pi calculus can also be expressed in other process algebras (extended with some notion of data) and other process models like, for example, high-level Petri nets.

The best support for my objections is in the paper itself. The paper illustrates the power of Pi calculus by calculating "1+2=3." This is a nice example, showing the essence of Pi calculus, but it does not show things that are not possible in other languages. Another interesting example is the so-called "Deferred choice," modelled in BPML. This example is one of 20 workflow patterns. It is interesting to note that the Deferred choice is supported by any process algebra incorporating "silent actions." Clearly, the pattern is very important and therefore is supported by languages like BPEL and BPML, but also by Petri-net-based workflow systems like COSA. However, the distinguishing features of Pi calculus (e.g., mobility) are not needed at all! The same holds for the 19 other workflow patterns described at www.workflowpatterns.com! Smith & Fingar, but also others advocating the use of BPEL and BPML, have not been able to demonstrate that any of these patterns requires Pi calculus to be supported. This does not mean that Pi calculus cannot support these patterns in some form. Nevertheless, it would be interesting to see a document where this is demonstrated. Smith & Fingar refer to a report of Intalio with the title "Technical Note: Implementing Workflow Patterns in BPML." This document is not about how Pi calculus supports the patterns but about BPML. We have studied several versions of this document and BPML. It shows that BPML, like most other languages, does not support all patterns in a direct manner. See www.workflowpatterns.com for more details. Note that we stimulated Intalio to make this report public. In fact, we offered to put their report on our WWW site. Unfortunately, over the last 1.5 years, Intalio has been reluctant to do so. We can only guess the reasons. In any case, there is no clear evidence that Pi calculus supports patterns in a better way than more traditional languages like Petri nets.

Another problem that I have with respect to the paper is the relation between Pi calculus and BPML and BPEL4WS. It is suggested that BPML and BPEL4WS are based on Pi calculus, but I dare to question this. What concepts in these languages can be attributed to things specific for Pi calculus? Again, concrete information is missing.

Page 3 of the paper refers to YAWL. The authors are interested in an industrial strength implementation of the language. There is a full implementation of the YAWL language. The tool, and also the sources, can be downloaded from www.citi.qut.edu.au/yawl/. This openness should be an example for many organizations in the BPM domain that are reluctant to show what has really been realized! YAWL is not intended as a commercial product, but as an example of a

language supporting the workflow patterns. The (initial) goal was not to build an industrial strength, but to set an example for future BPM products and standards.

We hope that these comments put “Workflow is just a Pi process” into perspective. Workflow requires more than Pi calculus. It would be similar to saying “Workflow is just a Petri net.” Such a “religious” approach to BPM will only create confusion and limit the application of all products and standards in this domain. Let us become less religious about these things and stick to the facts. Let us challenge the “BPM-activists” to demonstrate their support for the workflow patterns (cf. www.workflowpatterns.com) and hope for new and relevant patterns demonstrating the real added value of different languages (including Pi calculus).

Wil M.P. van der Aalst

Prof.dr.ir.Wil van der Aalst is a full professor of Information Systems and head of the section of Information and Technology of the department of Technology Management at the Technische Universiteit Eindhoven. He is also a part-time full professor at the Computing Science faculty at the department of Mathematics and Computer Science at the same university. He can be reached at P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands. His email is: w.m.p.v.d.aalst@tm.tue.nl

Wil van der Aalst is the lead-author of *Workflow Management: Models, Methods, and Systems (Cooperative Information Systems)* (MIT Press, 2002). He is a leading researcher in workflow patterns and has applied workflow patterns to BPEL4WS and other XML BP languages. For more information on this, check the Workflow Patterns site: http://www.tm.tue.nl/it/research/patterns.

