Software and Systems Modeling manuscript No.
(will be inserted by the editor)

Workflow Patterns Put Into Context

W.M.P. van der Aalst[1,2], **A.H.M. ter Hofstede**[2,1]

[1] Eindhoven University of Technology, The Netherlands
e-mail: `w.m.p.v.d.aalst@tue.nl`

[2] Queensland University of Technology, Brisbane, Australia
e-mail: `a.terhofstede@qut.edu.au`

Received: date / Revised version: date

Abstract In his paper "Approaches to Modeling Business Processes. A Critical Analysis of BPMN, Workflow Patterns and YAWL" Egon Börger criticizes the work of the Workflow Patterns Initiative in a rather provocative manner. Although the workflow patterns and YAWL are well-established and frequently used, Börger seems to misunderstand the goals and contributions of the Workflow Patterns Initiative. Therefore, we put the workflow patterns and YAWL in their historic context. Moreover, we address some of the criticism of Börger by pointing out the real purpose of the workflow patterns and their relationship to formal languages (Petri nets) and real-life WFM/BPM systems.

Key words workflow patterns – YAWL – Petri nets – business process management

1 Introduction

In [9], Egon Börger states that the patterns by the Workflow Patterns Initiative are "not well founded", "badly described", and "no suitable BPM benchmark". Moreover, he criticizes the semantical foundation of YAWL. Börger makes these claims in a rather provocative and emotional manner. Therefore, the editors encouraged us to write this rebuttal.

In the remainder, we first sketch the Workflow Patterns Initiative in its historic context. In [9], a substantial number of sentences are quoted from ten of our publications, which appeared between 2000 and 2011. Many of these quotations are out of context, e.g., quotations related to specification, analysis, and implementation are muddled up. Therefore, Section 2 describes the developments related to the Workflow Patterns Initiative over the last decade thereby clarifying the purpose of the patterns and the role of YAWL. In Section 3, we argue that patterns need to balance between generality and precision and should *not* be viewed as a formal specification of behavioral properties. Patterns should allow for multiple modeling techniques and not impose very specific semantics, e.g., the well-known design patterns [11] also do not impose the use of Java or any other programming language. In Section 4, we point out the complexity of the various interacting workflow perspectives and argue that there is no “Silver Bullet Formalism” that will make everything as simple as suggested in [9]. Finally, in Section 5, we acknowledge the need for empirical validation and discuss our efforts to find out what is happening in *real* workflow processes and WFM/BPM tools.

2 History of the Workflow Patterns

In the mid-nineties, Wil van der Aalst got involved in the *Sagitta* project when he was working as a part-time workflow consultant for Bakkenist Management Consultants. The goal of this project was to realize a workflow system for supporting the key processes within the Dutch Customs [3]. He was surprised by the way the Workflow Management (WFM) system was selected. Stakeholders had problems describing its desired functionality. For example, they did not distinguish between – what is called today – the *Deferred Choice* (Pattern 16) and the *Exclusive Choice* (Pattern 4). Although the Dutch Customs clearly had many processes that required support for the *Deferred Choice*, they were about to select a WFM system that did not support this at all [3]. This experience and also several similar experiences of the first author in the second half of the nineties, clearly showed the need for describing workflow patterns.

However, the real start of the Workflow Patterns Initiative was in 1999 when the authors met at CoopIS 1999 in Edinburgh. This resulted in the first paper on the workflow patterns at the same conference one year later [2]. Initially, the work received little attention, but this changed when the `www.workflowpatterns.com` website was launched and started to attract the attention of people involved in the selection and use of WFM systems. Several reports on the patterns appeared in Dutch IT journals and this encouraged vendors to study the workflow patterns more closely. In parallel, we refined and extended the patterns and started to systematically evaluate commercial WFM systems [6,20,21]. We also evaluated emerging standards such as BPEL4WS, BPML, XLANG, WSFL and WSCI.

Initially, the focus was on control-flow only. This changed in 2004 when Nick Russell joined the Workflow Patterns Initiative. Over time, we added patterns for the other core perspectives such as the workflow resource patterns [18], the workflow data patterns [19], and the exception handling patterns [17]. Later also patterns related to service interaction and flexibility were added. Whereas the control-flow patterns could be used to evaluate a language or notation, for the other patterns it made more sense to consider them in the context of a WFM/BPM system. In 2006 the original set of 20 control-flow patterns was revisited with the goal to provide a more precise description of each of the patterns and also to identify any potential gaps in the original set of patterns. As a result the number roughly doubled. In [9], Börger criticizes the growing number of patterns: “an exponential growth can be observed starting with 20 workflow patterns in 2003 going through 43 WPs in 2006 and reaching a praised 126 patterns (obtained by adding patterns for the so-called data and resource perspectives) in 2010. No reason is given that this growth has a *fundamentum in re* and a limit.” (note: references in this quote were removed). The reason for the growing number of patterns was the expanding scope of the Workflow Patterns Initiative and our experiences with a wide range of systems and standards. Moreover, as was clearly stated in our publications, we do not claim that WFM systems/languages should support all patterns. Users should first analyze which patterns they need and then use this short-list of patterns to select a system.

In parallel with the refinement and extensions of the patterns, we also started the development of the YAWL language and system [5,12]. YAWL is inspired by the workflow patterns and extends Petri nets to facilitate the use of more advanced patterns [4]. Börger is very critical about the development of YAWL as a reference implementation of the workflow patterns. He talks about “an obvious group interest” (Section 4.1 in [9]) suggesting that the patterns were used to promote YAWL. However, Börger misunderstands the true intentions of YAWL. The development of YAWL started in 2002 and was triggered by frequent discussions with practitioners that had problems understanding the subtle but important differences between the various patterns. Some vendors doubted that comprehensive patterns support could be achieved (without adding a lot of complexity). They would often ask “How do you do this?” and wanted to see concrete examples. This instigated the development of YAWL. The acronym (Yet Another Workflow Language) clearly shows that YAWL was not intended as a commercial effort trying to gain competitive advantage by exploiting the patterns. However, YAWL has had a very positive effect on the Workflow Patterns Initiative. When implementing YAWL new questions emerged, especially with respect to the interaction of the various patterns. The YAWL environment also helped to provide a proof-of-concept for various innovative research ideas related to workflow flexibility, work distribution, worklist visualization, configurable process models, verification, simulation, and process mining [12]. YAWL is one of the most visible open source WFM systems (see [22] for a patterns-based analysis of the other main open source WFM systems). YAWL has been downloaded approx. 120,000 times and the key paper [5] has been cited more than 800 times according to Google Scholar. Overall, we think that YAWL helped to progress WFM/BPM research and it is unclear why Börger goes to great lengths to denounce it (in [9] no other WFM systems are singled out for criticism).

In our view, the workflow patterns have had a very positive effect on WFM/BPM research and WFM/BPM products. Today, the patterns are widely used to describe workflow functionality in a language/system-independent manner. In addition, the patterns are also highly visible. The `www.workflowpatterns.com` website has been one of the most visited websites in the field of BPM averaging more than 300 visitors per working day over the last decade. Moreover, [6] is the most cited workflow paper ever (it has more than 2,250 citations according to Google Scholar).

3 Balancing Between Generality and Precision

In [9] both the informal descriptions of the workflow patterns and the example formalizations using YAWL and CPN Tools are criticized. Surprisingly, despite his criticism, Egon Börger has published various papers on formalizing the control-flow patterns in terms of Abstract State Machine (ASMs) [8]. On the one hand, he claims that our informal descriptions are vague and ambiguous. On the other hand, Börger does not like the explanations/formalizations in terms of (colored) Petri nets. In this section, we address these comments.

The notion of *patterns* as a means of categorizing recurring problems and solutions in a particular domain is generally attributed to Christopher Alexander [7]. Later the concept of patterns was adopted in software development, resulting in well-known patterns collections such as the design patterns by the "Gang of Four" [11], the business analysis patterns [10], and the enterprise application integration patterns [13]. See `http://hillside.net/patterns/patterns-catalog` for pointers to other collections. All of these collections have in common that they do not impose a particular language and that the description of the pattern is deliberately kept informal. When describing a pattern one needs to resort to concrete examples, e.g., the design patterns book by the "Gang of Four" [7] includes many examples using C++ and Smalltalk. *However, these examples should not be confused with the actual patterns.* We often use Petri nets to explain patterns. However, this does not imply that these Petri nets should be viewed as formal specifications of the pattern. Moreover, as our evaluations show, systems not based on Petri nets can also support any of the patterns if they provide the desired functionality. It is quite OK to formulate pattern examples in terms of ASMs, just like they have been formulated in various process algebras (e.g. in [14]). However, this does not touch upon the essence of the workflow patterns which are representation independent.

In Section 4.2 of [9], Börger argues that (colored) Petri nets are unsuitable "for the practice of BPM". As indicated in [4] and [5], we agree that Petri nets are unsuitable as a business process modeling language for end users. (This is the reason for developing more suitable notations like YAWL.) However, ASMs are even less suitable as an end-user language for modeling processes as they are not graphical and very different from mainstream languages such as BPMN, UML, EPCs, etc. We do not propose Petri nets as an end-user language. Instead we use it for:

- *Clarification:* Petri nets are graphical, well-known, and widely used. Therefore, it makes sense to illustrate the basic patterns in terms of small Petri net examples.
- *Semantics:* YAWL and various workflow functionalities (e.g., work distribution mechanisms) can be formalized in terms of colored Petri nets. Moreover, using CPN Tools such models are directly executable and can be used for analysis (simulation and state-space exploration).
- *Analysis:* Petri nets allow for different types of analysis. Not just model checking but also for various linear-algebraic techniques (invariants and the marking equation), Markov analysis, simulation, etc.

Clearly, many properties are undecidable unless rigorous abstractions are used. However, there is a large body of knowledge on analyzing workflows using Petri nets. A system such as YAWL provides support for checking soundness [12], but other techniques in areas such as process mining [1] and simulation [15] have been developed to support specific types of analysis relevant for the field of BPM. Such dedicated techniques do not exist for ASMs. In his paper, Börger refers to work on "Software Product Lines" for solutions. However, he is not very clear about this as it is just a side remark in the four page discussion on the "unsuitability" of Petri nets.

4 Make Things as Simple as Possible, But Not Simpler

In [12] we discuss the semantics of YAWL and in the PhD thesis of Nick Russell [16] a formalization of YAWL's control-flow, data, and resource perspectives is given in terms of colored Petri nets. The formal model of YAWL can be executed using CPN Tools. Börger complains that the CPN Tools model is not shown in [12]. The model is indeed large: 55 pages, 480 places, and 138 transitions. However, this model covers all three aforementioned perspectives and also the interactions between these perspectives. It is absurd to think that another formalization technique like ASMs will suddenly make things simple. The CPN Tools model describes many essential and complex mechanisms.

Egon Börger seems to advocate a "Silver Bullet Formalism" that will make things simple and manageable. There is definitely room for improvement. However, making such claims without having specified or implemented a WFM system that is able to support a reasonable number of patterns covering the control-flow, data, and resource perspectives is not very convincing.

5 Need for Empirical Validation

In Section 3.1 of [9], Börger writes "In fact there is *no statistical underpinning* showing how frequently which patterns appear in real-life business processes. Experimental data like that produced for BPMN constructs in [61] should have been put in place to validate the pattern selection.” We acknowledge the importance of empirical validation. However, the workflow patterns have been validated. First of all, the patterns are based on a detailed analysis of various WFM/BPM systems. Moreover, practical experiences obtained through a multitude of evaluations resulted in reformulations and further refinements of the patterns. Systems such as Staffware, WebSphere MQ Workflow, FLOWer, COSA, iPlanet, SAP Workflow, Filenet, jBPM, OpenWFE, Enhydra Shark, WebSphere Integration Developer and Oracle BPEL, and languages/standards such as BPMN, XPDL, BPEL, UML and EPCs have been analyzed using the patterns. Moreover, we have been interacting with workflow vendors, consultants, end-users and analysts regarding the patterns. Second, we looked at the frequencies of patterns in real-life projects. See for example the work done by Kristiaan de Vries and Oscar Ommert [20,21]. Under the supervision of the first author they evaluated five workflow projects conducted by ATOS/Origin to get quantitative data about the frequency of patterns. Each of these projects involved multiple processes for which the number of activities ranged from dozens to hundreds. The systems Eastman Enterprise Workflow, Staffware, and Domino Workflow were used in these projects. Empirical findings showed that in many of ATOS/Origin’s projects workflow designers were forced to adapt the process or had to resort to spaghetti-like diagrams or coding because particular patterns were not supported by the WFM system. The project also showed a positive correlation between the patterns being used and the patterns that were well supported, e.g., the processes developed using Staffware had much more parallelism than the processes developed using Eastman Enterprise Workflow. This example shows that patterns tend to be used frequently if they are supported well. This indicates that process design is influenced by patterns support.

A more elaborate empirical validation of the workflow patterns is welcome. Moreover, we would also like to refer to our work on process mining [1]. In this work we analyze processes based on event logs, i.e., we discover, monitor and improve real processes (i.e., not assumed processes) by extracting knowledge from event logs readily available in today’s information systems. Experience obtained by applying process mining in over 100 organizations shows that processes are often much more involved than what people would like to think. Typically, 80% of the process instances can be described by a rather simple process model. However, to model the remaining 20% more advanced patterns are needed.

Acknowledgements The authors would like to thank all that contributed to the Workflow Patterns Initiative and the development of YAWL. In particular Nick Russell, Nataliya Mulyar, Bartek Kiepuszewski, Alistair Barros, Petia Wohed, and Michael Adams.

References

1. W.M.P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes.* Springer-Verlag, Berlin, 2011.
2. W.M.P. van der Aalst, A.P. Barros, A.H.M. ter Hofstede, and B. Kiepuszewski. Advanced Workflow Patterns. In O. Etzion and P. Scheuermann, editors, *7th International Conference on Cooperative Information Systems (CoopIS 2000)*, volume 1901 of *Lecture Notes in Computer Science*, pages 18–29. Springer-Verlag, Berlin, 2000.
3. W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods and Systems (in Dutch).* Academic Service, Schoonhoven, 1997.
4. W.M.P. van der Aalst and A.H.M. ter Hofstede. Workflow Patterns: On the Expressive Power of (Petri-net-based) Workflow Languages. In K. Jensen, editor, *Proceedings of the Fourth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2002)*, volume 560 of *DAIMI*, pages 1–20, Aarhus, Denmark, August 2002. University of Aarhus.
5. W.M.P. van der Aalst and A.H.M. ter Hofstede. YAWL: Yet Another Workflow Language. *Information Systems*, 30(4):245–275, 2005.
6. W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns. *Distributed and Parallel Databases*, 14(1):5–51, 2003.
7. C. Alexander. *A Pattern Language: Towns, Building and Construction.* Oxford University Press, 1977.
8. E. Börger. Modeling Workflow Patterns from First Principles. In C. Parent, K.D. Schewe, V. Storey, and B. Thalheim, editors, *Conceptual Modeling (ER 2007)*, volume 4801 of *Lecture Notes in Computer Science*, pages 1–20. Springer-Verlag, Berlin, 2007.
9. E. Börger. Approaches to Modeling Business Processes. A Critical Analysis of BPMN, Workflow Patterns and YAWL. *Software and Systems Modeling*, ??(??):??–??, 2012.
10. M. Fowler. *Analysis Patterns: Reusable Object Models.* Addison-Wesley, Reading, Massachusetts, 1997.
11. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software.* Professional Computing Series. Addison Wesley, Reading, MA, USA, 1995.
12. A.H.M. ter Hofstede, W.M.P. van der Aalst, M. Adams, and N. Russell. *Modern Business Process Automation: YAWL and its Support Environment.* Springer-Verlag, Berlin, 2010.
13. G. Hohpe and B. Woolf. *Enterprise Integration Patterns.* Addison-Wesley Professional, Reading, MA, 2003.
14. F. Puhlmann and M. Weske. A Look Around the Corner: The Pi-Calculus. In K. Jensen and W.M.P. van der Aalst, editors, *Transactions on Petri Nets and Other Models of Concurrency II*, volume 5460 of *Lecture Notes in Computer Science*, pages 64–78. Springer-Verlag, Berlin, 2009.
15. A. Rozinat, M. Wynn, W.M.P. van der Aalst, A.H.M. ter Hofstede, and C. Fidge. Workflow Simulation for Operational Decision Support. *Data and Knowledge Engineering*, 68(9):834–850, 2009.
16. N. Russell. *Foundations of Process-Aware Information Systems.* PhD Thesis, Queensland University of Technology, 2007.
17. N. Russell, W.M.P. van der Aalst, and A.H.M. ter Hofstede. Workflow Exception Patterns. In E. Dubois and K. Pohl, editors, *Proceedings of the*

18th International Conference on Advanced Information Systems Engineering (CAiSE'06), volume 4001 of *Lecture Notes in Computer Science*, pages 288–302. Springer-Verlag, Berlin, 2006.

18. N. Russell, W.M.P.van der Aalst, A.H.M. ter Hofstede, and D. Edmond. Workflow Resource Patterns: Identification, Representation and Tool Support. In O. Pastor and J. Falcao e Cunha, editors, *Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05)*, volume 3520 of *Lecture Notes in Computer Science*, pages 216–232. Springer-Verlag, Berlin, 2005.
19. N. Russell, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst. Workflow Data Patterns: Identification, Representation and Tool Support. In L. Delcambre, C. Kop, H.C. Mayr, J. Mylopoulos, and O. Pastor, editors, *24nd International Conference on Conceptual Modeling (ER 2005)*, volume 3716 of *Lecture Notes in Computer Science*, pages 353–368. Springer-Verlag, Berlin, 2005.
20. K. de Vries and O. Ommert. Advanced Workflow Patterns in Practice (1): Experiences Based on Pension Processing (in Dutch). *Business Process Magazine*, 7(6):15–18, 2001.
21. K. de Vries and O. Ommert. Advanced Workflow Patterns in Practice (2): Experiences Based on Judicial Processes (in Dutch). *Business Process Magazine*, 8(1):20–23, 2002.
22. P. Wohed, N.C. Russell, A.H.M. ter Hofstede, B. Andersson, and W.M.P. van der Aalst. Patterns-based Evaluation of Open Source BPM Systems: The Cases of jBPM, OpenWFE, and Enhydra Shark. *Information and Software Technology*, 51(8):1187–1216, 2009.