Aspect-Oriented Business Process Modeling Approaches: An assessment of AOP4ST

Fernando Pinciroli, Jose Luis Barros Justo, and Raymundo Forradellas

Abstract—Aspect-oriented business process modeling (AOBPM) is an emerging discipline which has recently attracted the attention of researchers and professionals. In 2014, Amin Jalali reviewed the state of the art of AOBPM and proposed a framework to assess the available approaches. We have developed AOP4ST, an aspect-oriented process for the software development life cycle, which includes the business process modeling as its first phase, and which follows the Jalali's recommendations. In this paper, we present the assessment of the AOP4ST's business modeling capability by comparing it with all of the original proposals assessed by Jalali's framework. Finally, we present ongoing work for the improvement of our approach on its AOBPM capabilities and its ability to detect and separate concerns at early stages of the development.

Index Terms— business process modeling; aspect-oriented paradigm; AOBPM; requirement assessment; AOP4ST

1 Introduction

THE business model is one of the main models in the early stages of software development life cycle (SDLC) and the first one of the Enterprise Architecture [1], in which it plays a key role. Furthermore, any instrument enhancing the scope and benefits of the business model would be a very valuable contribution. The incorporation of the concepts of the aspect-oriented paradigm at this stage of the SDLC seems to be a good bet [2][3].

We have seen how different software development paradigms were initially appearing in the programming phase and then continue to evolve, upstream along the SDLC [4][5]. The aspect-oriented paradigm is no stranger to this process, so we can find that aspect-oriented programming languages are most widespread than the practices of requirement specification, and these, in turn, are more extended than the proposals at the business process modeling stage [6]. For this reason, we have not a lot of aspect-oriented proposals at the business modeling phase, although some are appearing, little by little, such as those reported by Jalali's work [7].

We have proposed AOP4ST, that stands for Aspect-Oriented Process for a Smooth Transition, that is a framework process to make the most with the benefits of the aspect-oriented paradigm by using tools and techniques currently spread in the industry, in order to allow a smooth transition until both, aspect-oriented tools and techniques, achieve a higher level of maturity and acceptance in real-world settings. AOP4ST proposes the application of the aspect-oriented paradigm for the SDLC. In this paper, we will focus on its application to the business modeling phase [8].

In 2014, Amin Jalali [7] proposed a set of requirements to be met for aspect-oriented methods at this stage of the SDLC and also provided a method for assessing these methods regarding the fulfilment of the requirements. The final product of Jalali's work was a comparative table covering all the existing proposals so far (up to the year 2013). Finally, the author describes open issues that the techniques still have to resolve. We have not found another comparison scheme for AOBPM so far.

In this paper, we apply the Jalali's criteria to assess AOP4ST's business model and compare it with other existent approaches.

In section 2 we briefly present the AOP4ST's schema. In Section 3 will make a brief description of the expected benefits from the application of the aspect-oriented paradigm in the business model. Section 4 will explain how AOP4ST covers the demanded requirements, while Section 5 presents the comparative table for the different approaches and how AOP4ST is positioned among them. Finally, we present our conclusions and highlight some open issues and future work in Section VI.

2 About AOP4ST

AOP4ST is a framework process, it is not a method or a methodology. It covers the whole SDLC.

AOP4ST's name highlights two main concepts: a) the AOP, "Aspect-oriented Process", indicates that it is truly aspect-oriented, ensuring that the widely known benefits of this paradigm are reached; b) the 4ST, "for a Smooth Transition", points to the possibility of applying this process immediately, because it employs widespread techniques, notations, standards, tools, etc. and allows to move to an aspect-oriented reality, taking advantage of the current state of the paradigm, until their own tools, techniques, etc. were imposed on the market.

The problem to be solved by AOP4ST is how to bring the benefits of aspect-oriented paradigm to the whole software development life cycle (SDLC), and allow the use of techniques, tools and standards currently widespread in the industry. In addition, the use of well-known techniques and tools allows to incorporate the aspect-oriented paradigm

- F. Pinciroli is with the Faculty of Informatics and Design, Champagnat University, Mendoza, Argentina. E-mail: pincirolifernando@uch.edu.ar.
- J.L. Barros Justo is with the School of Informatics (E.S.E.I.), University of Vigo, Ourense, Spain. E-mail: jbarros@uvigo.es.
- R. Forradellas is with the Faculty of Engineering, National University of Cuyo, Mendoza, Argentina. E-mail: kike@uncu.edu.ar.

gradually, until the different existing proposals have sufficient diffusion and maturity to warrant their employment in real and complex projects.

And when we talk about the whole SDLC, we are including the business model, not considered for the authors offering aspect-oriented approaches for the early stages of the SDLC, widely known as "early aspects".

Our proposal arises from various factors, which we have to face in the industry and in the adoption of new technologies for software development.

First, as we have previously said, the different software development paradigms initially appear in the programming phase and then continue their definitions upstream, along the SDLC. The aspect-oriented paradigm follows the same pattern, so we can find more proposals for the programming phase than for the early phases of the SDLC.

Second, many previous proposals linked to software development sound promising and offer benefits difficult to refuse, but their massive use in industry depends on many factors. A well-known case is that of the object-oriented databases, that beyond the benefits they offered and the enormous popularity of the object-oriented languages and development tools today, they did not yet achieve the leading role in industry that could be expected [9].

Finally, software development projects have to deal with many risks and, the main function of project leaders is to minimize the damage that these risks can cause. The use of immature technologies, tools without support in the market, techniques that have not been tested enough, etc., would be very risky decisions to take by who has the responsibility to carry out a successful software development project.

On the other hand, the availability of well-known tools and techniques and adherence to standards and best practices, will help professionals to make good estimates and to take better decisions.

AOP4ST is based on the hypothesis: it is possible to design an aspect-oriented software development process that encompass techniques, tools, notations and standards of widespread use in the industries. This development process is suitable for the early stages of the SDLC, making the most with the benefits of the aspect-oriented paradigm in real-world settings. Under certain circumstances, it is possible to make use of existing techniques, tools and standard notations, right now, while specific theoretical and practical instruments are developed and introduced in the market, achieving enough dissemination and support to justify its use in real software development projects, without the risks that this use imply today.

AOP4ST's basic structure for early aspects is composed of three models: business model, user requirements model and software requirements model. The last one if divided in three views: functional, static and state views. Concerns are progressive discovered along these models and views.

The business model, that it is the assessed in this paper, starts the process, being the first layer of the enterprise architecture and pulling on the rest of the models in order to meet the business goals. After detecting concerns (manual or automated aspect mining techniques can be used [10]), they are encapsulated and described with a BPMN 2.0 based notation. A set of composition rules allows business models to be rebuilt and potential conflicts minimized [8]. This is the model that we evaluate in this paper.

It is possible to find more information about AOP4ST in [11] and [12], and about AOP4ST's business model in [13], all published in Spanish.

3 Benefits from Applying the Aspect-Oriented Approach to the Business Model

Modularization is a long-sought goal along different paradigms. The object-oriented approach was an important step in this direction, but there are still many unresolved issues. Some of them are those that aspect-oriented paradigm try to solve by the so called of concerns: *"But nothing is gained —on the contrary!— by tackling these various aspects simultaneously. It is what I sometimes have called "the separation of concerns", which, even if not perfectly possible, is yet the only available technique for effective ordering of one's thoughts, that I know of. (…) It is being one- and multiple-track minded simultaneously"* [14].

At all stages of the SDLC it is possible to find elements of the problem domain, called "crosscutting concerns", entangled with other support elements, which, in turn are scattered throughout the system [15]. The aspect-oriented approach provides the solution to scattering (the same concept scattered throughout the system) and tangling (different concepts mixed together, decreasing cohesion and increasing coupling), leading to the benefits of maintainability, reusability, greater protection against errors, reduced complexity, understandability, etc. Examples of crosscutting concerns present in the business models are: transaction management, auditing, authorization, monitoring, security, and others.

If concerns are detected, separated and encapsulated away from the elements of the problem domain, we may obtain:

1. Cleaner business models, since they are not tangled with unrelated issues to the problem domain.
2. Business models totally focused on the needs of the very specific business processes.
3. Crosscutting concerns modeling and encapsulating specific processes not belonging to the problem domain.
4. The possibility to focus each analyst on each party's specific goals: the functional perspective is modeled separately of the quality perspective for the business models.

4 The Business Model in AOP4ST

AOP4ST offers the business model as the first level of abstraction and as the starting point of concern identification. They are not supposed to be found in their entirety, it is expected to identify them along the rest of the models. The most important thing here is to fulfill the objective of the business model and that let the concerns arise naturally.

The aspect-oriented approach allows in this model to clearly separate the activities from the problem domain with respect to those that are not. Thus, the first thing to do is to model the primary processes. During this modeling process, there may be recurring activities that are of a different nature: activities that belongs to the problem domain and supporting activities. The first ones are equivalent to the functional requirements and the second to nonfunctional requirements. These latter, in turn, will have a greater probability of being crosscutting, and not only to the processes of the domain of the problem that is being modeled, but also are for other different problem domains.

In other words, if we are modeling a banking system, we may see certain activities of the domain of the problem being repeated, such as checking the balance of a bank account, while we will also find other activities that are also repeated and do not belong to the domain of the problem, such as access control with user and password, logging, etc.

The repeated activities that belong to the domain of the problem can be modeled as reusable processes, using the notation that BPMN offers for this case, whereas the tasks that do not belong to the domain of the problem can be modeled as crosscutting concerns (Figure 1). Or we can model them both as crosscutting concerns. Whatever the case, the crosscutting concerns will be extracted from the process, encapsulated (Figure 2), and their insertion points will have to be indicated with an adequate notation (Figure 3), which in AOP4ST corresponds to an improvement of AO4BPMN, proposed by Charfi et al. [16].

The encapsulated piece of process is called “advice”, the insertion point is called a “join point” and the explicit indication in the join point is called “pointcut”. It is supposed that it will be possible to correctly compose the processes with the insertion of the crosscutting concerns. For this, we have defined a set of rules of composition [8] (see Section 7).

In AOP4ST, all the elements of modeling of processes (activities, events and gateways) can be join points. And there may be three kinds of crosscutting concerns: “before”, “after” and “around”, which will insert advices after, before and both, before and after, the join points, respectively.

This way of indicating the join points is called “explicit form” in AOP4ST. This approach offers a second way of indicating them, which is called “indirect form” and it is specified by means of a special notation that is placed in the encapsulated advices (Figure 4).

Later, the concerns will continue being discovered and completed throughout the rest of the AOP4ST’s models.

5 Requirements Covered by AOP4ST

Jalali proposes an evaluation method for business process modeling approaches based on a set of requirements to measure what an aspect-oriented business modeling technique must satisfy [7]. The author establishes the requirements that have to be met mandatorily and other optional requirements, along with a formula for calculating the degree of compliance with those requirements:

$$T = \frac{(\sum_{i=1}^{n} Ri * 4)}{R}$$

“T” is the final value of the measurement.
“R” is the requirement sub-item being assessed.
“i” is the ith requirement sub-item.

In this section, we present the requirement compliance by AOP4ST along with its corresponding foundation.

1. Basic requirements: they are mandatory and AOP4ST has a full compliance due to:
   1.1. Support definition of business processes using functional, control-flow, data and resource perspectives: the essence of AOP4ST lies in the use of standard notations and techniques. In particular, for the business model we use BPMN 2.0 [17], a standard that completely covers this requirement.
   1.2. Remove scattering problem in definition of concerns in process models: in AOP4ST we have included the Charfi et al.’s proposal [16], called AO4BPMN, to remove scattering. These authors offer two modeling alternatives: light and heavy versions. We opted for the light version because its notation adheres strictly to BPMN 2.0.
   1.3. Remove tangling problem in definition of concerns in process models: this requirement is met by incorporating AO4BPMN in AOP4ST, as in the previous requirement.

Crosscutting concerns often appears scattered throughout all business processes and tangled with the rest of their activities (Figure 1).

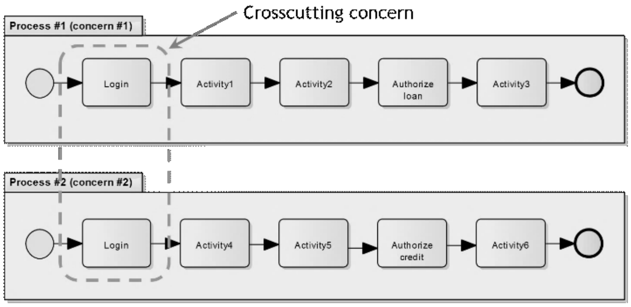

Fig. 1. Tangled and scattered crosscutting concern.

AOP4ST contemplates the encapsulation of crosscutting concerns (Figure 2) and, the addition of an annotation element, based on BPMN 2.0, to indicate where the concern has to be composed into the base processes (Figure 3).

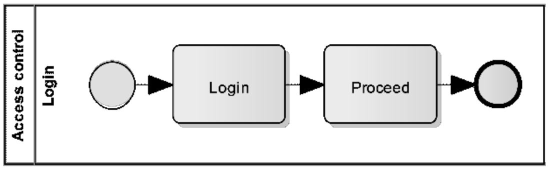

Fig. 2. Encapsulation of a crosscutting concern.

The "Proceed" element on Figure 2 represents the elements of the base processes where the encapsulated functionality, called "advice", have to be composed. These target elements are called "join points".

A "pointcut" denotes the joint points. AOP4ST uses the annotation element of BPMN 2.0 to do that, plus a short text with a controlled syntax to specify how to compose the advices. Figure 3 shows a "pointcut" for the concern "Access control" in two different processes.

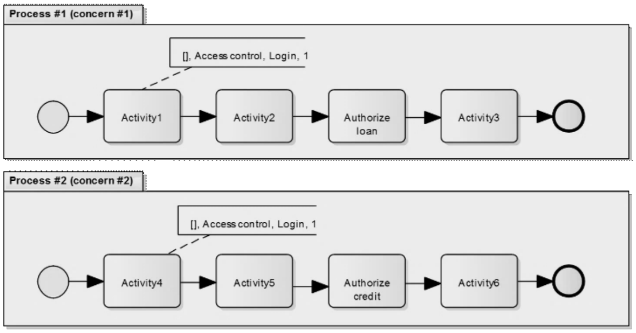

Fig. 3. Pointcut represented with annotation elements and indicating the join points.

The compliance of the optional requirements allows us to assess the offered aspect-oriented business modeling process. AOP4ST meets these requirements as follows:

2. Signature exposition.
   - 2.1. Expose points of control-flow perspective of processes for which a concern can be defined: processes are modeled in AOP4ST as an independent element, and within this element are modeled the diagrams to depict the processes. Thus, since a complete process is modeled as an individual element, it is possible to apply a concern to a process.
   - 2.2. Expose points for functional perspective for which a concern can be defined: as in Charfi et al.'s proposal, AOP4ST allows concern to be applied to any of the three elements of flow modeling in BPMN 2.0, that is: activities, events and gates.
   - 2.3. Expose points for data perspective for which a concern can be defined: in AOP4ST concerns can be applied to data elements of BPMN 2.0, which will imply the application of advices before, during or after reading or writing data, as appropriate.
   - 2.4. Expose points for resource perspective for which a concern can be defined: currently, AOP4ST does not provide the application of advices to resources.

3. Rule composition.

AOP4ST supports two ways to indicate a join point: the first way is to apply an annotate element of BPMN 2.0 on the same join point element; the second way is indirect, and it is defined by a pointcut with the information indicating the condition to be met by a joint point.

Figure 3 presents the direct join point selection, since an annotation element explicitly indicates a join point. An indirect advice application may be made in AOP4ST by means of the syntax presented on Figure 4, where it can be interpreted that an advice must be applied unconditionally (indicated with the empty brackets [ ]), on all the elements of the type "activity" (indicated with "act") whose names begin with "author".

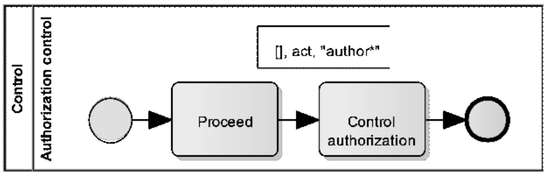

Fig. 4. Indirect advice application.

By considering the pointcut on Figure 4, two activities of Figure 5 will be selected as join points since their names match with the pointcut specification.

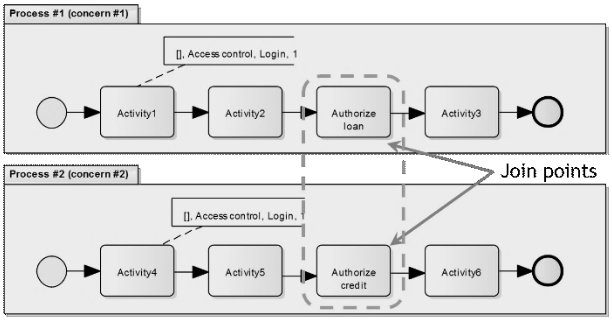

Fig. 5. Activities reached by indirect join point selection.

   - 3.1. Support definition of rules based on control-flow perspective information: advices can be applied directly and indirectly on the business process elements.
   - 3.2. Support definition of rules based on functional perspective information: advices can be applied directly and indirectly on all the activities, event and gate elements of the business process diagrams.
   - 3.3. Support definition of rules based on data perspective information: advices can be applied directly and indirectly on data elements.
   - 3.4. Support definition of rules based on resource perspective information: not supported in

AOP4ST yet.

3.5. Support composition of rules based on combinations of different process perspectives: direct and indirect selection ways for join points selection support this requirement, since they are independent of the modeling element which they are applied.

3.6. Support composition of rules without any dominant perspective: except for the composition of rules based on resources, that are not covered, AOP4ST has no dominant perspective, since it allows to apply an advice on a join point of any type without having to identify it in advance.

4. Advice relations.

4.1. Enable definition of before advice.

4.2. Enable definition of after advices.

4.3. Enable definition of around advices.

4.4. Enable definition of parallel advices for a join point: requirements 4.1 to 4.4 are already supported from Charfi et al.'s proposal.

4.5. Enable definition of nested advices: AOP4ST allows an element within an advice to be a join point (except "Proceed").

4.6. Enable definition of precedence among advices for a join point: in the form of the direct selection of a join point, AOP4ST considers the indication of the order of precedence of concerns for each join point in particular.

5. Transformation patterns.

5.1. Enable transformation of process level data among different related modules: not supported in AOP4ST yet.

5.2. Enable synchronization of "Proceed" placeholders in advices with advised join point: supported from Charfi et al.'s proposal.

5.3. Enable transformation of data among different related modules: not supported in AOP4ST yet.

5.4. Enable transformation of resources which has performed activities among different related modules: not supported in AOP4ST yet.

6. Phases support.

6.1. Support the design of aspect oriented business process modeling.

6.2. Support enactment of aspect oriented business process models.

6.3. Support adjustment of running advices.

6.4. Support adjustment of new advices for running cases.

6.5. Support adjustment of advices for new cases: requirements from 6.2 to 6.5 are not supported in AOP4ST yet.

6 Assessment Results

After applying the calculations to place AOP4ST in Jalali's comparative table, we reached the results presented in the tables below. In the last table (Table V), we have included a total column as the sum of all previous qualifications (FQ: final qualification) which reflects the overall score of the various proposals.

TABLE 1
BASIC REQUIREMENTS AND SIGNATURE EXPOSURE

Technique	1. Basic requirements			2. Signature exposure				
	1.1	1.2	1.3	2.1	2.2	2.3	2.4	SE
AO4BPEL [18]		1	1					0
Wang et al. [19]		1						0
Shankardass [20]		1						0
Collell [21]	1	1						0
AOBPMN [22]	1	1						0
Patiniotakis [23]	1	1	1		1			1
AO4BPMN [16]	1	1	1		1			1
Jabeen [24]	1	1	1		1			1
Cappelli [25]	1	1	1		1			1
Jalali [26]	1	1	1		1			1
AOP4ST	1	1	1	1	1	1		3

TABLE 2
RULE COMPOSITION

Technique	3. Rule composition						
	3.1	3.2	3.3	3.4	3.5	3.6	RC
AO4BPEL							0
Wang et al.							0
Shankardass							0
Collell							0
AOBPMN							0
Patiniotakis et al.		1					0.67
AO4BPMN		1					0.67
Jabeen et al.		1					0.67
Cappelli et al.		1	1		1	1	2.67
Jalali et al.		1	1		1		2
AOP4ST	1	1	1		1	1	3.33

TABLE 3
ADVICE RELATIONS

Technique	4. Advice relations						
	4.1	4.2	4.3	4.4	4.5	4.6	AR
AO4BPEL							0
Wang et al.							0
Shankardass							0
Collell							0
AOBPMN							0
Patiniotakis et al.	1	1	1				2
AO4BPMN	1	1	1	1			2.67
Jabeen et al.	1	1	1	1			2.67
Cappelli et al.	1	1		1			2

Technique	4. Advice relations						
	4.1	4.2	4.3	4.4	4.5	4.6	AR
Jalali et al.	1	1	1	1			2.67
AOP4ST	1	1	1	1	1	1	4

TABLE 4
TRANSFORMATION PATTERNS

Technique	5. Transformation patterns				
	5.1	5.2	5.3	5.4	TP
AO4BPEL					0
Wang et al.					0
Shankardass					0
Collell					0
AOBPMN					0
Patiniotakis et al.		1			1
AO4BPMN		1			1
Jabeen et al.		1			1
Cappelli et al.		1			1
Jalali et al.		1	1		2
AOP4ST		1			1

TABLE 5
PHASES SUPPORT AND FINAL QUALIFICATIONS

Technique	6. Phases support						
	6.1	6.2	6.3	6.4	6.5	PS	FQ
AO4BPEL						0	**0**
Wang et al.						0	**0**
Shankardass						0	**0**
Collell						0	**0**
AOBPMN						0	**0**
Patiniotakis et al.	1					0.8	**5.47**
AO4BPMN	1					0.8	**6.14**
Jabeen et al.	1					0.8	**6.14**
Cappelli et al.	1					0.8	**7.47**
Jalali et al.	1	1		1	1	3.2	**10.87**
AOP4ST	1					0.8	**12.46**

Finally, the radial graph shows the spatial coverage of the different proposal (Figure 6).

Figure 7 clearly shows how the requirement coverage of the two more complete techniques are almost complementary, since their overlapping is small, but the total coverage is close to one hundred percent (except for transformation patterns).

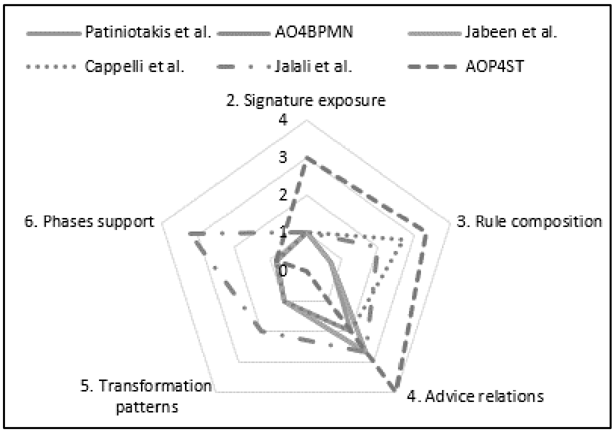

Fig. 6. Radial graph comparing the techniques.

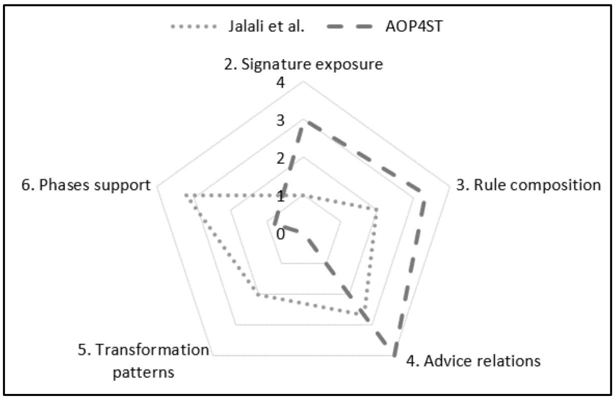

Fig. 7. Radial graph for Jalali et al. and AOP4ST approaches.

7 Conclusions and Future Work

Separation of concerns begins in the business process modeling phase and continues along the rest of the phases of the SDLC. Particularly, it allows experts in the problem domain to model specific business processes, while the rest of the functionality (non-problem domain) can be modeled separately, producing more understandable, maintainable and extensible models. The most important thing is to have the possibility to reach the objective of the business model, collaborating not only with a good business process description but also with the ability to improve processes, manage indicators, subordinate processes to business goals, etc.

Jalali's framework was designed to measure the ability of different approaches to allow the development of good business process models and achieve an optimal separation of concerns.

Perhaps it would be worth analyzing the potential threats to validity of Jalali's assessment method. We have subjected AOP4ST to the same criteria as the rest of the approaches, so we consider that the comparison allows us to obtain conclusions at least in relative terms.

The tables above and the radial chart show that most of the proposals are very similar, except for the Jalali's one and AOP4ST. These two proposals have a greater requirement coverage, and at the same time they are also complementary, as they cover the requirements with a different emphasis. This gives the opportunity to strongly improve AOP4ST from the ideas of Jalali's approach.

We have also developed a plug-in for the Enterprise Architect modeling tool that enables the composition of advices into the base processes, strictly using BPMN 2.0 and applying a set of composition rules that we have developed and presented in [8].

Pending issues in AOP4ST are:

1. Include resource perspective, according to that considered in requirements 2.4, 3.4 and 5.4: we are making progress in incorporating the lane element of BPMN 2.0 to identify the resources involved on each part of the processes.
2. Improve transformation patterns: to enable transformation of process level data and process level data among different related modules; no proposal supports the former, but Jalali presents an alternative to solve the latter that we are analyzing.
3. Incorporate the rest of the phases support: we have not advanced in this regard yet but the fact of working with the standard BPMN 2.0 gives us the possibility to move forward with BPEL (Business Process Execution Language [27]) fully on this issue.

ACKNOWLEDGMENT

The authors want to thank Solus S.A., the company who sponsored their research project at Champagnat University and allowed them to progressively apply AOP4ST in real world projects. They also wish to thank Aconcagua Software Factory S.A., because they could apply and validate AOP4ST in some of their projects too. And finally, they want to thank Fernando Leiva, Director of IT at ATM (Tax Administration of the province of Mendoza, Argentina) who gave them the opportunity of applying AOP4ST to model the whole business processes that will be entirely supported by a digital record platform, seeking to reduce the need for paper in public administration.

REFERENCES

[1] D. Greefhorst and E. Proper, Architecture Principles. The Cornerstones of Enterprise Architecture, vol. 6, no. 3. 2011.

[2] A. Pourshahid et al., "An aspect-oriented framework for business process improvement," in E-Technologies: Innovation in an Open World, vol. 26 of the series Lecture Notes in Business Information Processing, pp 290-305, 2009.

[3] A. Jalali, "Foundation of Aspect Oriented Business Process Management," Master's Thesis DSV, Stockholm University, 2011.

[4] L. F. Capretz, "A brief history of the object-oriented approach," SIGSOFT Software Engineering Notes, vol. 28, no. 2, p. 6, 2003.

[5] "History of Programming Languages," in History of Programming Languages Conference, vol. I, II and III.

[6] C. V. Lopes, "Aspect-Oriented Programming: An Historical Perspective (What's in a Name?)," pp. 1–5, 2002.

[7] A. Jalali, "Assessing Aspect Oriented Approaches in Business Process Management," in Perspectives in Business Informatics Research, vol. 194 of the series Lecture Notes in Business Information Processing, pp. 231–245, 2014.

[8] F. Pinciroli, "Aspect-oriented business process composition rules in AOP4ST," in 35th International Conference of the Chilean Computer Science Society (SCCC 2016) held in conjunction with the 42th Latin American Computing Conference (CLEI 2016), 2016.

[9] N. Leavitt, "Whatever Happened to Object-Oriented Databases?," Computer, Long Beach, California, vol. 33, no. 8, pp. 16–19, 2000.

[10] F. Pinciroli, "Considerações acerca da mineração de aspectos," Perspectivas em Ciências Tecnológicas, vol. 5, no. 5, pp. 83–101, 2016.

[11] F. Pinciroli, "AOP4ST – Aspect-Oriented Process for a Smooth Transition," in Proceedings of the XVII Workshop de Investigadores en Ciencias de la Computación, WICC 2015, Salta, 2015.

[12] F. Pinciroli, "Aspect-Oriented Process for a Smooth Transition," in Ph.D. Symposium of the IEEE 11 Congreso Colombiano de Computacion Proceedings, Popayan, 2016.

[13] F. Pinciroli and L. Zeligueta, "El modelo de negocio en AOP4ST," in Proceedings of the XVIII Workshop de Investigadores en Ciencias de la Computación, WICC 2016, pp. 423-427, Concordia, 2016.

[14] E. W. Dijkstra, "On the role of scientific thought," in Selected writings on Computing: A Personal Perspective, New York: Springer-Verlag, pp. 60–66, 1982.

[15] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. V. Lopes, J.-M. Loingtier, and J. Irwin, "Aspect-Oriented Programming," ACM Comput. Surv., vol. 28, no. June, pp. 220–242, 1997.

[16] A. Charfi, H. Müller, and M. Mezini, "Aspect-oriented business process modeling with AO4BPMN," in Modelling Foundations and Applications, vol. 6138 of the series Lecture Notes in Computer Science, pp. 48–61, 2010.

[17] Object Management Group, "BPMN 2.0." 2011.

[18] A. Charfi, "Aspect-Oriented Workflow Languages AO4BPEL and Applications," Dissertation at the Darmstadt Univ., 2007.

[19] J. Wang, J. Zhu, H. Liang, and K. Xu, "Concern oriented business process modeling," Proceedings of the IEEE 10th International Conference on e-Business Engineering, pp. 355–358, 2007.

[20] A. Shankardas, "The Dynamic Adaptation of an Aspect-Oriented Business Process in a Service-Oriented Architecture Platform." Master's Thesis, Athabasca Univ., Canada, 2009.

[21] D. C. Collell, "Aspect-Oriented Modeling of Business Processes," Master's thesis, der Technischen Universitat Darmstadt, Darmstadt, 2012.

[22] A. Jalali, P. Wohed, and C. Ouyang, "Aspect Oriented Business Process Modelling with Precedence," in Business Process Modeling and Notation, vol. 125 of the series Lecture Notes in Business Information Processing, pp. 23–37, 2012.

[23] I. Patiniotakis, N. Papageorgiou, Y. Verginadis, D. Apostolou, and G. Mentzas, "An aspect oriented approach for implementing situational driven adaptation of BPMN2.0 workflows," vol. 132 of the series Lecture Notes in Business Information Processing., pp. 414–425, 2013.

[24] A. Jabeen, S. Tariq, Q. U. A. Farooq, and Z. I. Malik, "A lightweight aspect modelling approach for BPMN," Proceedings of the 14th IEEE International Multitopic Conference, pp. 255–260, 2011.

[25] C. Cappelli, F. M. Santoro, J. C. S. do P. Leite, T. Batista, A. L. Medeiros, and C. S. C. Romeiro, "Reflections on the modularity of business process models: The case for introducing the aspect-oriented paradigm," Business Process Management Journal, vol. 16, no. 4, pp. 662–687, 2010.

[26] A. Jalali, P. Wohed, C. Ouyang, and P. Johannesson, "Dynamic Weaving in Aspect Oriented Business Process Management," On the Move to Meaningful Internet Systems, vol 8185 of the series Lecture Notes in Computer Science, pp. 2–20, 2013.

[27] OASIS, "OASIS Web Service Business Process Execution Language," 2007.

Fernando Pinciroli has a degree in Systems and Computing, finalizing his PhD thesis in the National University of San Juan. He is Dean of the Faculty of Informatics and Design, in the Champagnat University (Mendoza, Argentina) and Director of the Research Institute of the same Faculty. He taught postgraduate courses and numerous courses and lectures in several countries. Has published articles in numerous international events and he is co-author of three books. Participated in the definition of the UML through the Object Technology User's Group. He is a consultant in Software Engineering since 1993 and is currently president of Solus S.A. in Mendoza, Argentina, and is President of Sparx Systems Argentina, sister company of Sparx Systems Ltd. Pty. in Australia.

José L. Barros-Justo hold a degree in Computer Science (1986) and a PhD degree in Software Engineering since 2006. He worked at Hewlett-Packard, K-Centro Training and i.d.e.a. informatics. Since 1994, he has been working as a full-time professor (tenured track) at the University of Vigo (Spain). Dr. Barros-Justo has served as committee member in International Conferences such as ICSR, CAiSE and Quatic. His research interests include software reuse, evidence-based software engineering, aspect-orientation, requirements patterns, controlled vocabularies and software development methodologies.

Raymundo Forradellas is Engineer in Electronics, by the National Technologic University of Buenos Aires, Engineer in Telecommunication, homologated by the Ministry of Education and Sciences of Spain, and hold a PhD degree in the Program of Artificial Intelligence by the Polytechnic University of Madrid, Spain. University professor since 1972, in all categories, with activities in degree and postgraduate, and Category I Professor/Researcher. He is Director of diverse master programs in Argentina, Coordinator of international programs between France and Argentina, Director of several Research Projects, and has multiple publications in Scientific Journals and in International Conferences.