πSOD-M: Building SOC Applications in the Presence of Non-Functional Requirements

(List of changes and answers to the reviewers' observations)

October 16, 2017

Editor's remarks:

> We have received the review reports for your paper "pisodm: Building SOC Applications in the Presence of Non-Functional Requirements".
>
> We require now that you implement in your submission the following recommendations made by the reviewers:

We have taken into consideration all the comments made by the reviewers, and addressed them in the new version of the paper. Our answers to the reviewers (below) explain how this has been done.

The text included in the new version is highlighted for a better identification of the changes to the document.

Reviewers' comments:

Reviewer A:

> First of all, on the form:
> 1. Footnotes do not appear in the paper.

Fixed in the new version.

> 2. On page 21, title 5.2.4 you have a missing reference.

Fixed in the new version.

> 3. On page 23, second paragraph, you have an incomplete sentence: " . . . are implemented according a the knowledge. . . "

Fixed in the new version.

> On the content:
>
> 1. On Section 2, the related works are cited as such. This section lucks of an evaluation of these works, and of the motivation of your proposed method.

We added 2 paragraphs at the end of Section 2: (i) considering the evaluation of the related work; and (ii) also motivating our proposed method.

> 2. This method can be fully automated? Is it appropriate to non-IT professionals?

We highlighted at the beginning of section 4 (page 15), that the method is semi automatic. Transformation in the more abstract stages is completely automatic (i.e., from π-UseCase to π-ServiceProcess, and from π-ServiceProcess to π-ServiceComposition). In contrast, the transformation to specific platforms (i.e. from π-ServiceComposition to π-PEWS), the analyst must intervene and check the generation of the specification code because there are programming details that can be interpreted in different manners depending on the target platform. In this sense, the method is more appropriate for those who have background knowledge in Information Technology. The transformations from CIM to PIM level models are not automatized, due to the informal nature of the CIM level. At the end of section 3.2.2 we mention that the selection of services that match the specification requires human intervention.

> 3. While designing the application to develop using your models, how is the selection of services done?

We highlighted in section 3.2.2, how the service is chosen. In order for a service to be selected, it needs to fulfill contract (constraints). If a service fits the contract, then it will be used in the execution flow. As it is noted in section 3.2.2, this part of the process is not automatic. Choosing the right service depends on the verification of a property that is done manually in the current version of the tool.

> 4. The paper validates the method with an industrial use case, but it lacks of a real evaluation that shows how the proposed method improves the existing once. For instance, evaluating the satisfaction of the users of the method, and comparing their satisfaction with the use of another development method.

We intend to evaluate our solution as suggested by the reviewer in our future work. That said, it is worth mentioning that the effectiveness of the solution proposed was showcased in the paper using a real-world application that underlines the features of our solution. In page 24 Section 5.3 we discussed about the importance of the use case for validating the methodology and about the perspectives of evaluating (i) time to market when a new module will be designed and implemented in the system (ii) maintainability of the new module having a separated and clear vision of business logic and non-functional properties.

> 5. How the integration and the description of the constraints (Non-functional properties) improves the development process and the developed application?

Decoupling the description (specification) of non functional properties from the functional aspects, provides developers with the means to focus on the logic of their application. This isolates them from the nitty-gritty aspects of non-functional aspects. Furthermore, it improves modularity by allowing the same application to be run under different non-functional settings, if needed. We explained this on page 24 paragraph starting with "We learned ..."

> 6. These non-functional requirements can include other types of constraints?

The non-functional requirements concept used in our method caters for the following types of constraints (page 9): (i) business rules (constraints / contract rules) associated to the behaviour of the application (services); (ii) Value Constraint related to the data types associated to attributes (like in the relational model); (iii) exceptional behaviour associated to the behaviour of the application when an exception is produced.

Figure 4 (π-UseCase Meta-Model.) presents the Policy View Concepts used to design the constraints.

> 7. On page 13, you specify that the model can be automatically translated into actual computer programs. Is this actually developed? If not, how is that can be possible?

Yes, that is the case.The π-PEWS model can be automatically translated into a π-PEWS executable program. We implemented this feature using the Acceleo platform.

Reviewer B:

> Changes which must be made before publication: + Footnotes do not appear

Fixed in the new version.

> + Some references are not inserted [?]

Fixed in the new version (Section 5.2.4).

> + Page 4, section 3.1.1: what is "value activities"? operations on values?

We added a footnote explaining the value activity concept (according to [1]).

> + Page 5 , paragraph following items: e3 value value model...

Fixed in the new version.

> + Page 22 paragraph on the bottom of the page: a too long sentence; I do not understand anything in this paragraph.

We have rewritten the paragraph.

> + Page 23: service composition instead of services composition

Fixed in the new version.

> + The method is an extension of SOD-M, but I was wondering why the new name is piSOD-M. If an explanation exists, it would be interesting to indicate (since there is nothing to do with pi-calculus sometimes used in the WS world)

The motivation behind the name is not related to the π-calculus which is indeed used for formalizing service coordinations. π comes from Policy which is the central notion to represent NFR's and the way they are enforced at runtime. As you have seen, NFR is at some point implemented by sets of Event - Reaction rules. In the domain of active databases sets of rules could define a Policy. π stands of the pronunciation of the first letter of the word "policy" which is /p/. The explanation of the name now appears in a footnote of page 2, the first time the name π-SOD-M appears in the text.

> + Section 6 does not mention any future work (except on its title!)

We have added a final paragraph describing directions of future work.

Final remarks

We would like to thank the anonymous referees for their useful and constructive critics to our paper.

References

[1] J. Gordijn and H. Akkermans. Value based requirements engineering: Exploring innovative e-commerce ideas. *Requirements Engineering Journal*, 8:114–134, 2002.