Gmail
by Google

SOFSEM 2014 notification for paper 86

SOFSEM 2014 <sofsem2014@easychair.org>   Fri, Oct 11, 2013 at 11:51 PM
To: Martin Musicante <mam@dimap.ufrn.br>

Dear Martin Musicante,

I am happy to inform you that your paper
Supporting Non-Functional Requirements for Services Software Process: An MDD Approach
submitted to SOFSEM 2014 was accepted.

Please, follow the recommendations of the reviewers and submit the final version
of your paper for the proceedings in time {i.e. October 25, 2013 at latest).

In order to include your paper in the LNCS proceedings we shall need the LNCS Consent to Publish form ;
this (recently updated version of the) form has to be signed by the authors of each paper, at least by the
corresponding author, and collected electronically together with the final version of the paper. We shall
send it to you in due time.

It is also important to communicate with the organizing committee in case you
need visa to enter Slovakia.

Looking forward to seeing you at SOFSEM.

Branislav Rovan

---------------------- REVIEW 1 --------------------
PAPER: 86
TITLE: Supporting Non-Functional Requirements for Services Software Process: An MDD Approach
AUTHORS: María Valeria De Castro, Martin Musicante, Umberto Costa, Plácido A. Souza Neto and
Genoveva Vargas Solar

OVERALL EVALUATION: 1 (weak accept)

----------- REVIEW -----------
In the paper, the authors introduce an extension of the Service-Oriented Development Method (SOD-M).
SOD-M provides guidelines to specify functional requirements of business logic by combining pre-existing
services. Their extension πSOD-M aims to add support for non-functional requirements (NFR) specification,
so it should be an improvement of the construction process by providing an abstract view of the application
and ensure the conformance to its specification.

SOD-M provides the viewpoints CIM, PIM and PSM and transformation rules to convert models top-down. In
πSOD-M, the authors add the concept of Policy to SOD-M, which is referenced but not explained. πSOD-M
extends SOD-M with (meta-)models at the PIM and PSM, and transformation rules between them. So the
πSOD-M models are able to represent and transform specifications of both, functional and non-functional
requirements. They prove their concept with an example.

Reading Section 4, I stumbled over the definition of the NFRs. For example, NFR 1 should be formulated as
follows: To access Twitter, one needs to authenticate himself. Describing the way of authentication by a
protocol and to define that three failure login attempts block an account is more of a functional requirement
specification. I see an inadequate formulation of NFR 4 and 5 too, because the described functional behavior
is a response to an infringed constraint.

Section 4 describes the proof of concept, which is appropriate to underpin the abstract introduction of πSOD-
M. The relationship between the model elements at different abstraction levels is understandable, but the
explanation of the transformation processes could use some improvement. The authors state that the π-
UseCase model is transformed into a more detailed model semi-automatically. It is not clear to me which
steps are conducted automatically and which ones manually. Furthermore I do not know if the other

transformations (model-to-model or model-to-text) are made manually or which of them are realized through the named or other tools.

Besides this, all described constraints deal with rights and performance issues, which are only a small part of NFRs. Because I think this approach is an interesting way to break NFRs down, I tried to access the comprehensive information as stated in the last paragraph of Section 4. The ref... Portuguese; I would highly recommend not referring to non-English references... Furthermore the related work and conclusion sections should be separate...

I separated both sections we are still in 12 pages

---------------------- REVIEW 2 --------------------
PAPER: 86
TITLE: Supporting Non-Functional Requirements for Services Software Process: An MDD Approach
AUTHORS: María Valeria De Castro, Martin Musicante, Umberto Costa, Plácido A. Souza Neto and Genoveva Vargas Solar

OVERALL EVALUATION: -1 (weak reject)

----------- REVIEW ----------
Summary

In the paper, the authors extend their previous work, the Service-Oriented Development Method (SOD-M), to include the representation of non-functional requirements (NFRs).
SOD-M proposes dedicated meta-models on each of the three MDD levels (CIM, PIM, PSM) to develop and generate service-oriented applications using model transformations (model-to-model, model-to-text).
These meta-models have been extended to include Policies for representing NFRs in the following way:
 (i)   In the new Use Case meta-model, non-functional constraints can be attached to functional tasks
 (ii) In the new Service Process meta-model, the defined constraints are represented as assertions (pre- and post-conditions) to the actions representing the tasks
 (iii) In the new Service Composition Model, dedicated policies group the assertions, which are executed before and after each service call
 (iv) In the new PEWS model (textual), policies are represented as part of the PEWS composition language
An example is provided that shows the transformation from constraints on the PIM to text in the PSM.

Contributions

The main contributions of this paper are the extensions of the meta-models and the respective transformations to allow the specification of non-functional requirements in the SOD-M method.

Strengths

The topic is very interesting and it seems that tool support for the methodology has been (partially?) provided.
The example in the paper illustrates the additions necessary to allow the specification of NFRs very well.

Weaknesses

Overall, the paper misses to convince the reader regarding the relevance of the contribution and leaves some open questions, as discussed below:
- The relevance to the general community is not clearly motivated. Is the PEWS language widely used? W... are the target users of this approach?
- The concepts of the base methodology and of the extension are not made clear, e.g., used modeling and transformation languages, definition of constraint types etc.
- In the related work section, many other approaches have been mentioned, but no clear distinction is made to the presented approach. What is the benefit of using this instead of other approaches?

Major Improvements

- A clear motivation should be given, in the introduction as well as in the related work as to why this work is relevant or how it differs from existing work.
- The actual basis for the work, i.e., the SOD-M method, is only introduced very shortly (1/2 page) and also the description of the extension is kept very short (1 - 1 1/2 page). Many general concepts could be discussed in these sections, e.g., what basis is used for modeling (UML?), what transformation language is used (QVT?), how are the constraint types (value, business, exception behavior) defined, how is the grouping within the policies done (e.g., performance + availability), etc.

Minor Improvements

- Tool support is mentioned, but not clearly defined. So there is a tool supporting the transformation from use case models to service process models, what about the other transformations?
- Figures
  * Legends in the figures would make the figures be able to stand on their own, e.g., in the Use Case Model: Bubbles are use cases/tasks, 'rectangles' are constraints
  * Figure 4 is hard to read if the paper is printed out
- References
  * In the reference section, the capital letters in the titles are not preserved, e.g., 'mda-based' -> 'MDA-based'
  * Reference 20 is a bit vague, do you reference an OMG papers, the website?
  * Reference 28, to which the reader is directed twice in the paper, is in Portugese, so it would be good to have additional information on this topic to target also a broader audience
- Minor spelling errors
  * When counting something within a sentence, I would use lower case letters, e.g., 'So, the challenge... is related to (i) the.., (ii) the...' (the instead of The)
  * Check for uniform spelling, e.g., 'service-oriented' vs 'service oriented' and spaces, e.g., SOMF [7], but BPM[18]
- The language and spelling in the paper may be improved by an additional proof-read (e.g., 'modes' instead of 'models')

---------------------- REVIEW 3 --------------------
PAPER: 86
TITLE: Supporting Non-Functional Requirements for Services Software Process: An MDD Approach
AUTHORS: María Valeria De Castro, Martin Musicante, Umberto Costa, Plácido A. Souza Neto and Genoveva Vargas Solar

OVERALL EVALUATION: -1 (weak reject)

---------- REVIEW ----------
The authors propose an approach that extends SOD-M with the objective to also model non-functional requirements and report a toy-example to illustrate the application of the approach.

In general, the paper is readable but should undergo some further spell checking (see examples of typos). The topic of modeling and assuring software quality (usually expressible by non-functional requirements) is an important topic, especially, for new paradigms such as model-based development.

The proposed method may extend the scope of requirements expressible with SOD-M, however, it seems still a long way to support non-functional requirements in general (as suggested by the title of the paper). The example application focuses mainly on security requirements, which are depending on one's viewpoint rather functional than non-functional requirements. Moreover, "maintain a log" is a cross-cutting concern but not really a non-functional requirement. How to deal with typical non-functional requirements such as performance, availability, reliability, or maintainability requirements is not mentioned in the paper.

Your discussion of related work is very focused on the approaches to develop web services but does not take in consideration work on quality modeling and non-functional requirements. There are a number of quality models in literature that may help you with your effort better support non-functional requirements in development of web services (see e.g., Kläs, M., Heidrich, J., Münch, J., Trendowicz, A., "CQML Scheme: A Classification Scheme for Comprehensive Quality Model Landscapes" for an overview and classification of quality models, and, e.g., A. Goeb, K. Lochmann "A software quality model for SOA" for a specific quality model that might be relevant for your work). There is also work on requirements methods that focus on the definition and operationalization of non-functional requirements (e.g., Doerr, Joerg, et al. "Non-functional requirements in industry-three case studies adopting an experience-based NFR method."). So please take in consideration to also make u
se of existing work on quality modeling and non-functional requirements or at least discuss in the related work section why you cannot make use of them.

In comparison to the presented proof of concept, the description of SOD-M and your extension are very short and hard to follow. You should explain the concepts you introduce in these sections in more detail and give examples to illustrate them. Moreover, it will help the reader if you include not only the elements of piSOD-M but also their mapping to the original SOD-M in Fig. 1.

Examples of typos: p2: notion of model+"s" ; … applications and provide-"s"; service+"-"oriented ; p6: are model+"ed" ; model+"-"to+"-"mo

---------------------- REVIEW 4 --------------------
PAPER: 86
TITLE: Supporting Non-Functional Requirements for Services Software Process: An MDD Approach
AUTHORS: María Valeria De Castro, Martin Musicante, Umberto Costa, Plácido A. Souza Neto and Genoveva Vargas Solar

OVERALL EVALUATION: 2 (accept)

---------- REVIEW ----------
The paper presents an extension of Service-Oriented Development Method (SOD-M) to include meta-model/model for representing non-functional requirements, transformation rules, and rules to transform models to concrete implementations.

I would suggest that the authors provide a more detailed description of SOD-M in chapter 2. It should be also compared with other existing solution and its specific features should be highlighted.