A Proposal for a Common Representation Language for MDE Settings

Fábio Paulo Basso and Toacy Cavalcante Oliveira (Advisor)

Federal University of Rio de Janeiro, COPPE-PESC, RJ, Brazil
{fabiopbasso,toacy}@cos.ufrj.br, expected defense date: 03/2016

Abstract. Empirical evidences suggest the need for a common representation language to be used in the core of a Knowledge Base (KB) for Model Driven Engineering (MDE) Settings (components, libraries, metamodels and model transformations). The absence of a common representation hampers the reuse and collaboration. The state-of-art introduces MDE in target software projects with Domain Specific Languages (DSL) for integration of MDE Settings, embedded with concepts for Model Transformation Chain (MTC), Model-Driven Integration (MDI), shared model-based tools though Web Services (WS), process patterns and others. This research abstract present a proposal for a common representation language named RAS++, intersecting existing concepts from DSLs for integration of MDE Settings and adding concepts from the Reusable Asset Specification. Thus, it is presented past and ongoing work in the RAS++ in a PHD proposal.

Keywords: Knowledge Base; Model Driven Engineering; Reusable Asset Specification; Common Representation Language

1 Introduction

Regardless of a certain maturity in tool support, Model-Driven Engineering (MDE) still need improvements [1]. Mussbacher and others [2] pointed out that concepts already well established within MDE community include tools for modeling, model-based analysis and model transformation management, while some issues include other concepts that lead MDE engineers to find and integrate appropriate solutions for specific needs in MDE-based processes. As a long-term goal for the MDE community, some authors suggested that solutions for MDE should be shared on the web in a Knowledge Base (KB) [3], thus allowing the quickly discovery and comprehension of appropriate MDE Settings (model transformations, tools, metamodels, files, etc.).

A central component of a KB is a "common representation language" [2]. In a common KB, MDE Settings must be represented independently from the Domain Specific Languages (DSLs) used for orchestration and execution of Model Transformation (MTs) such as: a) Model Transformation Chains (MTCs) [4, 5]; b) Component models for model transformations (CBD of MTs) [6]; c) Model-Driven Integration(MDI) for model tool chains [7]; d) Orchestration of MDE settings on the cloud through Web Services (WS) [8]; e) Integrated Software Development Process (SDP) with MDE (MDE-SDP) [9] and; f) Process patterns for MDE, considering MDE Settings as process components shared on repositories [10].

We have found in the literature some requirements for this common representation language. Concepts introduced in the aforementioned DSLs are important [2]. It is also important to consider that this language will be used on the core of a KB, which means that it should be flexible to represent MDE Settings that provide "theory", such as those stored in the Repository for Model Driven Development - ReMoDD [11] (examples, data-sets and modeling pearls). Due to specificity on MDE [5], it also must be different from the Essence, a language for method engineering used in the KB proposed by SEMAT [12], which is limited to be used in this context. The common language should facilitate the discover and comparison of MDE Settings [3]. In this sense, another level of information is needed between the catalog of MDE Settings [13] and the representation of its technicalities [10], such as descriptive information that enable to non-experts the access to this shared data [14].

We propose a new language named RAS++ with concepts relevant for a common representation language for MDE Settings. RAS++ extends the Reusable Asset Specification (RAS) [15], an important language to structure elements for reuse through instruction for integration and classification of these settings in repositories. RAS++ also maps common concepts from the aforementioned DSLs. This work discusses main features of the RAS++, pointing out to results and preliminary work.

2 Motivation and Issue

Concepts that circle the MDE adoption are of interest of some companies whose the core business is the MDE, as illustrates Figure 1. In an effort to introduce model-based solutions in start-up contexts (Figure 1, box 3), several resources for MDE (core assets) have been developed to support for specific needs in target software projects [4]. MDE Settings (e.g., model transformations and tools) are applied in different contexts of target companies, thus reusable [10]. The reuse of these resources need a representation language and tool implementation that make flexible the construction of MDE Settings [6], as the FOMDA DSL [16] illustrated Figure 1, box 2.

FOMDA DSL have been used to adapt and execute model transformation components in several software projects, similarly as many other DSLs discussed previously. So far, the existing tool support accomplished the needs. However, the current scenario is more challenging because target companies are also interested in MDE Settings developed around the world, as illustrated Figure 1 (1). In this scenario, MDE settings will be available in KBs and must be searched, analyzed and integrated with the core assets illustrated in Figure 1 (2). Besides, MDE Settings are represented with different DSLs, imposing some difficulties for this

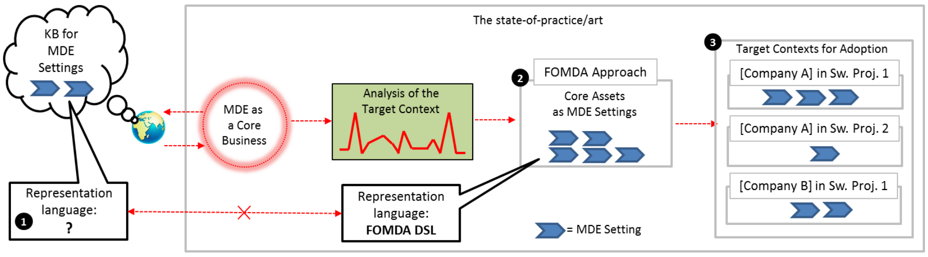

Fig. 1. Illustration of the Current Scenario for MDE as a Core Business

integration. Thus, we need a common representation language to be used in a KB shown in Figure 1 (1).

In order to implement this new scenario for reuse, we found that at least three steps shown in Figure 2 must be executed. The first step "Specification" is to specify MDE Settings in a common format (model in conformity with the CRL), allowing the storage of data associated with MDE Settings in a KB and retrieval in the same format. Then, software engineers/developers download these MDE Settings (not yet the physical content, but the information represented in a model) by searching the KB and comparing its features in the step "Acquisition". In the step "Transformation", model-to-model transformations together with a "deploy engine" will adapt MDE Settings for a specific format (model in conformity with a DSL X). This would allows to implement the scenario motivated in Figure 1, which present the following issue:

No agreement for common representation for MDE Settings. This is key point in this research, because the lack of this language makes difficult the reuse of MDE Settings in a global reuse scenario [3, 2]. The literature of the area is rich with contributions for specific representation languages for MTC [5, 4], Component model for MTs [16, 6], MDI [7], WS [8], integrated SDP [9] and process patterns [10]. However, it is missed a more abstract and general representation, which could promote the interchange of the information shareable between these DSLs (e.g., semantic for model-based operations, model transformation components and metamodels). The lack of a common representation language makes the concepts associated with MDE Settings replied in each DSL, which is bad to the advent of a common KB for MDE that must keep a uniformity for the stored information. This would allow, for example, to simplify the use of FOMDA DSL (focused in orchestration and adaptation of MT components) [16] together with other DSLs such as TIL (focused in MDE tool integration) [7]. In other words, a common representation language could promote the complementarity of these specific representation languages.

In this regard, our research question is *what is common between these DSLs?* To the best of our knowledge, there is no answer in the literature of the area. Due to our previous experience with the FOMDA DSL [16], we assumed in [13] that some of abstractions from MTCs should be used. The problem is that, besides DSLs for MTC, several other also allows to integrate MDE Settings in target

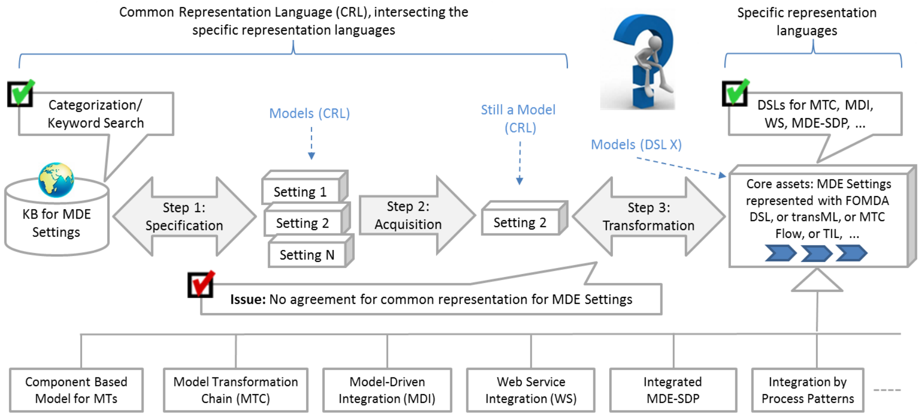

Fig. 2. A Scenario of Use of Information Associated with MDE Settings Through a Common Representation Language

contexts. Thus, a study that find the common abstractions would be of great contribution to the research and practice on the MDE.

3 Related Work

This work targets a subject of research that is quite new in literature of the area: a common representation language for MDE. Accordingly, there is no specific work that we can use to compare our contributions. Thus, we compared with the information shared in existing repositories used to store this type of data, with assets specifications and with specific DSLs for technicalities on the MDE.

Knowledge Bases. The ReMoDD [11], for example, makes use of only searches about some proposals published in some conferences such as MODELS, ECMFA, etc. Most information that should be explicit is available in documents, papers and tutorials, which requires long time to find and analyze adequate options. On the other hand, we are representing this information summarized in a structured and interchangeable format (RAS++). SEMAT [12] is an initiative to provide a knowledge base in Software Engineering, using the Essence as a core representation language, which is limited to represent data from MDE Settings but useful to represent methods. This suggest that research on MDE to accomplish this new scenario can mirror on the SEMAT, but it must keep focus on the MDE specificity (the case for the proposed language).

Assets. We can mention some extensions for RAS to represent information associated with software components [17–20]. These works propose to specify data related with components for application integration and service oriented architecture [20, 17], feedback from users [18], component software license [19] and standard taxonomies. These extensions can also be applied to summarize information of some technical solutions for MDE. However, they are limited to represent technical information associated with MDE settings as the ones represented with the RAS++.

Technicalities. Our current contribution analyzes abstractions needed in a common representation language, while our previous contributions in FOMDA DSL [16] analyzed requirements to support the execution and adaptation of MDE settings. Similarly to the FOMDA DSL, others are essential to introduce MDE in target contexts such as transML [5], MTC Flow [4], Bento DSL [6], TIL [7], , SPEM extensions [9] and process patterns [10]. However, they are limited to be used in the core of a KB due to its specificity and lack of rich structured descriptive data. Thus, RAS++ is more abstract and general.

Related work do not scales well in this new scenario for reuse. This way, RAS++ present considerable contributions to enable its implementation, complementing existing DSLs with concepts for a common representation.

4 Overview of the Proposed Solution

In [13] we proposed the RAS++, a new metamodel with concepts considered common to represent MDE Settings. In the following we discuss on the main features in RAS++.

- **This metamodel is based on the Reusable Asset Specification (RAS) [15]. An asset is:** anything that provides reuse and value through reference (links), cataloged with standardized taxonomies, described by a set of properties and owning zero or more specifications about artifacts [21]. Assets are specifications usually represented in XML, that abstract data associated with any sort of artifact that can be reused. Through concepts related with semantics for reuse, assets are used to describe software components [19], application and domain models [22] and, more recently, tools that help in the execution of software engineering tasks in the context of MDE [8]. Reusable assets (RAS) [15] can be stored in existing reuse repositories and retrieved in the same format, providing structures to support the reuse of MDE Settings and tasks to instruct the integration of these settings in target contexts. Thus, concepts introduced in asset specification languages such as RAS and Asset Management Specification (AMS) [21] are important for a common representation language.
- **RAS++ intersects concepts from specific representation languages**, some presented in [5–10] for MTCs, MDI, WS, etc. These DSLs are designed to manage model transformations in a higher-level of abstraction than the model transformations rules implemented with ATL, QVT, ETL, etc. DSLs such as transML [5], MTC Flow [4], TIL [7] and FOMDA DSL [16] have been reported as important to introduce MDE in target contexts. Therefore, some concepts introduced in these proposals such as tasks/components, artifacts/parameters, metamodels and connectors/bindings are also of interest for a common representation language.
- **RAS++ increments existing works with structure for rich descriptive data connected with technicalities.** After the study in [13] we noticed that other level of information is needed in between the catalog and instruction promoted by assets and the representation of technicalities of MDE Settings used by aforementioned DSLs. In this regard, Mohagheghi et al. concluded in 2009 that establishing a bridge for this gap between data used by technicians (e.g., MTCs)

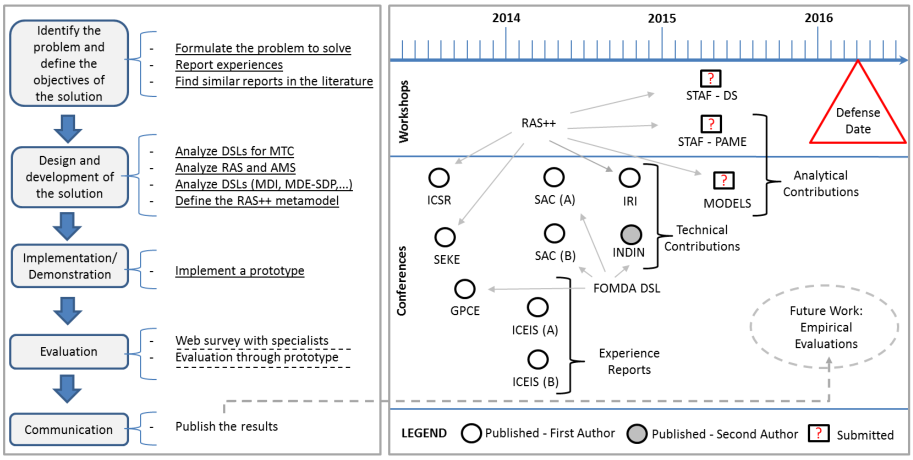

Fig. 3. Research methodology and progress

and by non-experts (e.g., description) is a key for success of MDE [14]. In order to implement the scenario illustrated in Figure 2, bridge this gap is a need. Today, the existing DSLs do not allow to connect rich descriptive information with technical data. For example, the use of catalog information associated with MDE settings as allowed in the RAS [15] and in a MTC approach (e.g., Bento DSL [6]) is not enough to ensure that a setting A is properly compared with a setting B. Recent surveys claimed that this is a problem because industry needs to compare MDE settings to decide the one that best fits for specific needs [3]. Thus, connect rich descriptive data with technicalities of MDE Settings is also a requirement for a common representation language.

In current version of the RAS++ we bridged this gap presented by Mohagheghi et al. [14], introducing concepts that connect rich and structured descriptive data with technicalities from MDE Settings. This way, it is proposed a study focused on concepts needed in a common representation language.

Expected contributions: A new representation language implemented with two prototypes, one to design assets and other to transform assets into target DSLs. The result will allow to implement the scenario illustrated in Figure 2 with well accepted and common concepts for MDE Settings and with increments to the state-of-art in asset specifications.

5 Preliminary Work

Figure 3 shows the research methodology and progress. Underlined text indicate a concluded activity, dashed text indicate ongoing activities and the other activities are to be started. To a follow-up of our progress, a grade of our contributions is presented in right-part of Figure 3.

– **Identify the problem and define the objectives of the solution**

  - **Formulate the problem to solve.** We started with a previous experience reported in past publications. In PHD disciplines for software reuse it was discussed on many problems that hampered the introduction of MDE in target context such as the lack of a KB and a common representation. These issues resulted on the analysis of the RAS as a complement for MTCs. We developed a preliminary version of RAS++ and a tool prototype, allowing to publish new works in conferences (ICSR and SEKE) [23, 24].
  - **Report experiences on accomplishments and challenges for the scenario illustrated in Figure 1.** In parallel to the RAS++, it was important to report benefits, limitations and drawbacks from our experiences in conferences (GPCE and ICEIS) [16, 25, 26].
  - **Find similar reports in the literature.** Since 2013, the literature of the area on issues for the MDE adoption have been investigated. We found recent in surveys, experiences and reports common positions on issues that makes hard the reuse of MDE Settings considering a global scenario. Thus, we have found only in recent publications the basis that support our claims for the discussed issues, which suggest that this is a new and relevant research topic.
- **Design and development of the solution.**
  - **Analyze DSLs for MTC.** We looked for the state-of-art for MTCs to compare it with our experiences in [16]. In addition, it was presented new contributions to the FOMDA DSL in conferences (SAC and INDIN) [27–29]. We found common concepts and representations. Moreover, the state-of-art is limited to accomplish the three steps (Specification, Acquisition and Transformation) illustrated in Figure 2.
  - **Analyze RAS and AMS.** We found in RAS the possibility to implement the Step 1 (Specification). In [13] we presented the result of our extension for RAS, proposing new meta-classes in support for the technicalities found as common for MTCs. A new contribution submitted for the MODELS conference targets a “modeling pearl” track [30], focusing in the modeling descriptive information in assets represented with the RAS and AMS.
  - **Analyze DSLs (MDI, MDE-SDP, ...).** We found common concepts proposed in these DSLs. A future work will present the result of our analysis.
- **Implementation/Demonstration.** In order to validate the RAS++ metamodel, we developed two prototypes: **a) An EMF-based designer tool** used to specify the assets, important for the Step 1 (Specification); **b) Eclipse-based “RAS++ Deploy” plug-in** that aims at transforming the RAS++ models into some of specific DSLs. These prototypes are continually improved to support new metaclasses introduced in the RAS++ metamodel and are discussed in [23, 24, 13].

6 Evaluation Plan

The evaluation will be carried out through literature review, web survey with specialist and practical work.

6.1 Goal

Our goal with this evaluation is to assess the feasibility and effectiveness of the RAS++ as a common representation language. In other words, we wanted to verify the ability of RAS++ (and related support mechanisms) to express semantics for MDE Settings, which should be possible to be integrated in other specifications. This goal is detailed bellow according to the structure proposed by Wohlin et al. [31]:

Table 1. Research goal

Analyze	The specification of MDE Settings by using our RAS++ DSL implemented in an EMF-based support prototype.
For the purpose of	Characterize.
Regarding	Feasibility and effectiveness of common representation language.
From the viewpoint of	Researcher and Domain Expert.
In the context of	MDE-based software projects.

6.2 Hypothesis

RAS++ provides concepts needed for a common representation language for MDE Settings. This hypothesis will be evaluated through the verification of the meta-classes introduced in the RAS++ and evaluating the transformation from RAS++ assets to the existing DSLs for model transformation management:

- **Web survey with specialists.** This study aims at understand some metaclasses proposed on the literature (e.g., metamodel, artifacts, workflow elements, filters, etc), evaluating what do people consider relevant to include in a common representation language for MDE Settings. Accordingly, we planned a web survey with questionnaire to find qualitative and quantitative data that highlight to the relevance of some concepts introduced in existing DSLs. Thus, this study will clarify which concepts should or should not be part of RAS++;
- **Evaluation through prototype.** This is a practical evaluation, based on implementations for the step 3 (Transformation). This practical evaluation will allows to transform MDE assets represented with the RAS++ in target specifications e.g., MTCs [6], MDI [7], SPEM [9].

7 Conclusion

This work presented a proposal for a common representation language for MDE Settings named RAS++. In order to enable the implementation of a new scenario which needs the introduction of MDE Settings through Knowledge Bases (KBs), we surveyed the literature to find similarities and differences between proposals that aim at introducing MDE Settings in target contexts. Therefore, we highlighted to our main contributions that add in RAS++ the support for syntax and semantics, associated with MDE Settings, in structures for reuse promoted by asset specifications.

A last consideration is that, although a KB for MDE is a long-term goal for the MDE community, it will only be viable if it agrees in a common representation language. Thus, through the work done so far, we conclude that the concepts introduced in RAS++ are relevant to be included in this language.

References

1. Liebel, G., Marko, N., Tichy, M., Leitner, A., Hansson, J.: Assessing the state-of-practice of model-based engineering in the embedded systems domain. In: Model-Driven Engineering Languages and Systems. MODELS'14 (2014) 166–182
2. Mussbacher, G., Amyot, D., Breu, R., Bruel, J.M., Cheng, B.H., Collet, P., Combemale, B., France, R.B., Heldal, R., Hill, J., Kienzle, J., Schöttle, M., Steimann, F., Stikkolorum, D., Whittle, J.: The relevance of model-driven engineering thirty years from now. In: Model-Driven Engineering Languages and Systems. (2014) 183–200
3. Whittle, J., Hutchinson, J., Rouncefield, M., Burden, H., Heldal, R.: Industrial adoption of model-driven engineering: Are the tools really the problem? In: Proceedings of the 16th International Conference on Model Driven Engineering Languages and Systems. MODELS'13 (2013) 1–17
4. Yie, A., Casallas, R., Deridder, D., Wagelaar, D.: Realizing model transformation chain interoperability. Software & Systems Modeling **11**(1) (2012) 55–75
5. Guerra, E., de Lara, J., Kolovos, D.S., Paige, R.F., dos Santos, O.M.: transml: A family of languages to model model transformations. In: 13th International Conference on Model Driven Engineering Languages and Systems. MODELS 2010 (2010) 106–120
6. Cuadrado, J.S., Guerra, E., Lara, J.D.: A component model for model transformations. IEEE Transactions on Software Engineering **40**(11) (Nov 2014) 1042–1060
7. Biehl, M., El-Khoury, J., Loiret, F., Törngren, M.: On the modeling and generation of service-oriented tool chains. Software & Systems Modeling **13**(2) (2014) 461–480
8. Zhang, W., Moller-Pedersen, B.: Modeling of tool integration resources with oslc support. In: Model-Driven Engineering and Software Development (MODELSWARD), 2014 2nd International Conference on. (Jan 2014) 99–110
9. Polgár, B., Ráth, I., Szatmari, Z., Horvath, A., Majzik, I.: Model-based integration, execution and certification of development tool-chains. In: 2th Workshop on Model-Driven Tool & Process Integration (MDTPI). (2009) 35–46
10. Hebig, R., Giese, H., Stallmann, F., , Seibel, A.: On the complex nature of mde evolution. In: Proceedings of Model Driven Engineering Languages and Systems. MODELS 2013 (2013) 436–453
11. : ReMoDD av. at <www.cs.colostate.edu/remodd/v1/>. at sept. 2014. (2014)
12. Johnson, P., Ekstedt, M., Jacobson, I.: Where's the theory for software engineering? Software, IEEE **29**(5) (2012) 96–96
13. Basso, F.P., Werner, C.M.L., Oliveira, T.C.: Towards facilities to introduce solutions for mde in development environments with reusable assets. In: International Conference on Information Reuse and Integration. IRI'14 (2014) 195–202
14. Mohagheghi, P., Fernandez, M.A., Martell, J.A., Fritzsche, M., Gilani, W.: Mde adoption in industry: Challenges and success criteria. In Chaudron, M.R., ed.: Models in Software Engineering. Volume 5421 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2009) 54–59

15. : RAS reusable asset specification version 2.2 november 2005. av. at <http://www.omg.org/spec/ras/>. at march 2014. (2014)
16. Basso, F.P., Pillat, R.M., Oliveira, T.C., Becker, L.B.: Supporting large scale model transformation reuse. In: 12th International Conference on Generative Programming: Concepts & Experiences. GPCE'13 (2013) 169–178
17. Park, S., Park, S., Sugumaran, V.: Extending reusable asset specification to improve software reuse. In: Proceedings of the 2007 ACM symposium on Applied computing. SAC '07 (2007) 1473–1478
18. Hadji, H.B., Su-Kyoung, K., Ho-Jin, C.: A representation model for reusable assets to support user context. In: IEEE International Symposium on Service-Oriented System Engineering, SOSE '08. (2008) 91–96
19. Hong-min, R., Zhi-ying, Y., Jing-zhou, Z.: Design and implementation of ras-based open source software repository. In: Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Volume 2. (2009) 219–223
20. Elgedawy, I.: Reusable soa assets identification using e-business patterns. In: Services - II, 2009. SERVICES-2 '09. World Conference on. (2009) 33–40
21. : Asset management specification. av. at <http://open-services.net/wiki/asset-management/oslc-asset-management-2.0-specification/> (2014)
22. dos Santos, R.P., Werner, C.: Analyzing the concept of components in the brechó-vcm approach through a sociotechnical and software reuse management perspective. In: Software Components, Architectures and Reuse (SBCARS), 2010 Fourth Brazilian Symposium on. (2010) 21–30
23. Basso, F.P., Werner, C.M.L., Pillat, R.M., Oliveira, T.C.: A common representation for reuse assistants. In: 13th International Conference on Software Reuse. ICSR'13 (2013) 283–288
24. Basso, F.P., Werner, C.M.L., Pillat, R.M., Oliveira, T.C.: How do you execute reuse tasks among tools? a ras based approach to interoperate reuse assistants. In: 25th International Conference on Software Engineering and Knowledge Engineering. (2013) 721–726
25. Basso, F.P., Pillat, R.M., Rooz-Frantz, F., Frantz, R.Z.: Study on combining model-driven engineering and scrum to produce web information systems. In: 16th International Conference on Enterprise Information Systems. ICEIS'14 (2014) 137–144
26. Basso, F.P., Pillat, R.M., Frantz, R.Z., Rooz-Frantz, F.: Assisted tasks to generate pre-prototypes for web information systems. In: 16th International Conference on Enterprise Information Systems. ICEIS'14 (2014) 14–25
27. Basso, F.P., Pillat, R.M., Oliveira, T.C., Fabro, M.D.D.: Generative adaptation of model transformation assets: Experiences, lessons and drawbacks. In: 29th Symposium On Applied Computing. SAC'14 (2014) 1027–1034
28. Basso, F.P., Oliveira, T.C., Farias, K.: Extending junit 4 with java annotations and reflection to test variant model transformation assets. In: 29th Symposium On Applied Computing. SAC'14 (2014) 1601–1608
29. Paulon, A., Frohlich, A., Becker, L., Basso, F.: Wireless sensor network uml profile to support model-driven development. In: 12th IEEE International Conference on Industrial Informatics, At Porto Alegre, RS, Brazil. INDIN 2014 (2014) 227–232
30. Basso, F.P., Oliveira, T.C., Werner, C.M.L.: Modeling mde settings with asset specification languages (on review) <http://prisma.cos.ufrj.br/wct/articles/models15.pdf>. MODELS'15 (2015)
31. WOHLIN, C., RUNESON, P., HÖST, M.: Experimentation in Software Engineering: An Introduction. Springer; 1 edition (1999)