Choose your preferred life-cycle and SofIA will do the rest ⋆

María-José Escalona[1], Laura García-Borgoñon[2], Julían García-García[1], Guillermo López-Nicolás[2], and Nora Parcus de Koch[1]

[1] Universidad de Sevilla, Spain
mjescalona,juliangg,norakoch@us.es
http://es3.us.es/
[2] Instituto Tecnológico de Aragón, Spain
laurag,glopez@itainnova.es
http:www.itainnova.es

Abstract. The importance of requirements engineering for software quality is well-understood in industry. It is also clear that requirements engineers need tools that do not prescribe only one type of development process. This paper presents an overview of SofIA, a CASE tool that provides maximum flexibility when modeling functionality, data or prototypes because it can use any given model to generate other models. SofIA is built on the experience acquired from having used the previous NDT-Suite in industrial projects for more than 20 years. It achieves its objectives by supporting bidirectional transformations and guaranteeing traceability between all models. Initial evaluations performed in the academic environment have shown that students require less training and feel more comfortable when following their own modeling process.

Keywords: Requirements modeling · Mockups · Early testing · Model verification · Bidirectional traceability

1 Introduction

Quality software development is a complex issue that continues to pose many challenges. Development methodologies have evolved and have had to adapt to different life cycles, such as waterfall and iterative processes, and to different management models, including agile modes. SofIA (Software Methodology for Industrial Applications) is a proposal for models and artifacts that is accompanied by a CASE tool of the same name. In this work, we present an overview of the SofIA tool. SofIA is inspired by an earlier proposal called NDT (Navigational Development Techniques), and its corresponding tool, NDT-Suite [2]. NDT is a model-driven methodological environment that has been widely used in the industry over the last twenty years. Thanks to the lessons learned, SofIA incorporates new research results that improve the NDT proposal and

⋆ Supported by Nico Project (PID2019-105455GB-C31). Ministerio de Ciencia, Innovación y Universidades.

increase its applicability in industrial projects. The most notable features are: (1) SofIA proposes not just one specific life cycle but a set of models that are defined over metamodels. The artifacts of these metamodels are related by bidirectional traceability. (2) This bidirectional traceability allows the developers to start with the approach they find most comfortable. For example, some teams prefer to develop prototypes as the first element, while others prefer first to elaborate the functional requirements or to start with a conceptual model. SofIA supports all three of these approaches because they are all linked by transformations that allow a model to be generated from the others. (3) SofIA offers transformation-based model synchronization to guarantee consistency. This ensures that if either of two artifacts changes, the analyst can trace their relationship and check that the change does not produce any inconsistency between them. As mentioned, this work presents the tool that supports the SofIA methodology. Section 2 offers an overview of the tool and its architecture and explains how the three indicated improvements were implemented. Section 3 describes the first validations that were carried out. The article ends with some conclusions and ideas for future work in Section 4.

2 SofIA in a Nutshell

SofIA was designed using the four-level architecture [3], that has traditionally been used to establish relationship between models and metamodels (see Figure 1): **M3 (metametamodel level)**. Following the OMG MDA architecture, MOF was established as the meta-metamodeling language. **M2 (metamodel level)**. Here, metamodels were defined describing M1 level models and the traceability between them. SofIA incorporates mainly aspects related to requirements (functional, UX/UI, interaction flow) and testing. **M1 (model level)**. This level includes models that represent the reality of M0 level are included at this level. The following models were included in SofIA: class diagrams, use cases, scenarios, tests, prototypes, interaction flows. **M0 (real world level)**. Real world data and information was incorporated. Two engines, denominated Driver and Quality, were developed. As can be seen, SofIA's main contribution was at the M2, M1 and M0 levels. At the M2 level, five metamodels were defined: Conceptual, Functional Requirements, Prototype, Testing and Interaction Flow. These metamodels provided us with the concepts and relationships that allowed us to represent the elements used in the SofIA proposal. A Traceability metamodel [1] was also included, establishing conceptual traceability connections between the different elements of the metamodels. At the M1 level, models conforming to the previous metamodels used in the proposal were selected. Thus, for the Conceptual metamodel a UML class diagram was incorporated, whilefor the Functional Requirements metamodel a UML use case model and one or more scenario models were included. For the prototypes metamodel a mockups model was introduced and, the IFM metamodel incorporated an interaction flow model. Finally, at the M0 level, we included transformation and traceability engines to give SofIA three fundamental features: the engines perform transformations,

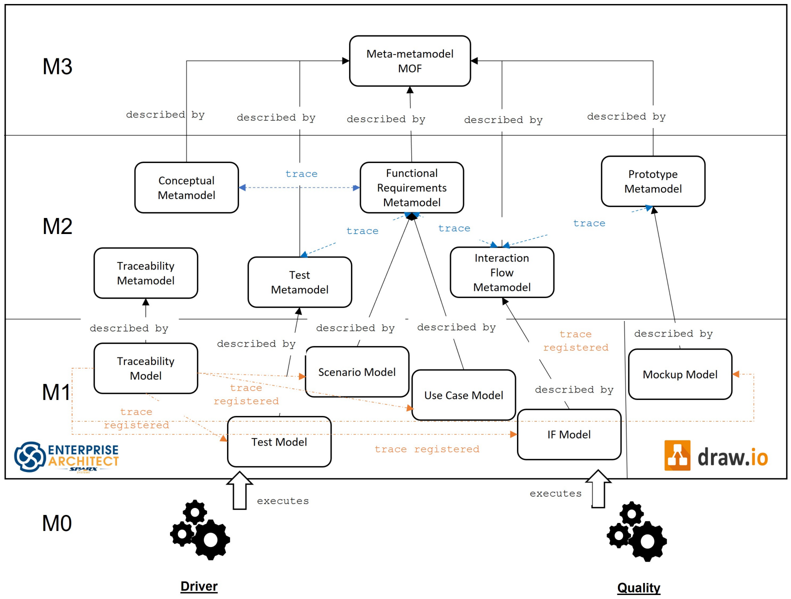

Fig. 1. SofIA's architecture

establish and maintain bidirectional traceability between models as described in level M2, and ensure that models are well-formed, that is to say, they conform to the corresponding metamodels, including their constraints. In accordance with the definition of MDE solution proposed in [4], SofIA was built up by using the facilities provided by the Enterprise Architect (EA) and Draw.io tools to incorporate the necessary metamodels, models and transformations. EA offersed the capability to extend base UML modeling elements through profiles to include our own metamodels and modeling tools (i.e, diagrams or toolboxes). It also provided an Add-In facility, which enabled us to extend default functionalities using our own code. Add-Ins are the best way to implement transformations and maintain bidirectional traceability. However, EA is not user friendly for the quick design of sketches as mockups, hence the inclusion in SofIA of Draw.io, an intuitive web tool, that makes it possible to develop their own toolboxes, add tags to convert elements into metaclasses, and import/export models into XML format to maintain consistency between prototypes and all other models.

3 Evaluation

SofIA is currently being evaluated in academic and industrial contexts, but some preliminary results from the academic environment can already be reported. In this regard, we have analyzed the number of meetings and training sessions requested by 29 computer engineering students using SofIA (16 FDP; 55,17% ratio) and NDT-Suite (13 FDP; 44.83% ratio). Here, it is relevant to mention the evaluation context: (1) these FDPs were of similar complexity in terms of factors like use cases, data entities and mockups; (2) the students had just one a single SofIA training video; and (3) the students had 8 NDT-Suite audiovisual resources (manuals, YouTube videos, research papers, etc.) plus an example of a software project designed with the NDT-Suite. Considering this scenario, we observed that the total number of training sessions required by the students was higher when NDT-Suite was used. More specifically, the students required 46 and 35 training sessions, respectively when using NDT-Suite and SofIA respectively. On average, these preliminary results show that the flexibility and automation offered by SofIA made it possible to reduce the training sessions by 23.91% compared to NDT-Suite.

4 Conclusions

This paper briefly presents SofIA, a CASE tool for designing and developing software applications, which offers flexibility regarding starting points for designs and automated support for bidirectional traceability. The focus is on requirements engineering and the building of mock-ups, use cases and data structure models. The SofIA architecture is based on the four levels M3 to MO of the OMG MDE definition and was created by extending the Enterprise Architect and Draw.io tools. So far, the tool has been validated by students working on their Final Degree Projects (FDP). In the future task we intend to implement more models in the platform to support other kind of requirements, such as security or accessibility. We also want to explore the possibility of adding heuristics, patterns, and even some machine learning protocols to help analysts define their requirements in a high quality. One future task will be to implementat additional alternatives regarding requirements models. We already started using SofIA in some transference projects, but these are still in their initial phases, so it is still too early to present anyresults. We plan to use and evaluate SofIA in controlled software experiments in industrial projects in the near future.

References

1. Escalona, M. J., Koch, N., García-Borgoñón, L. Lean requirements traceability automation enabled by model-driven engineering. PeerJ Computer Science, 8.2022.
2. García García, J. A., Escalona, M. J., Domínguez, F. J., Salido. A. NDT-Suite: a methodological tool solution in the model-driven engineering paradigm. Journal of Software Engineering and Applications, 7 (4), 206-217.2014.
3. González, C., Henderson, B.: Metamodelling for Software Engineering. John Wiley & Sons. 2008.
4. Molina, JG, Rubio, F.O.G., Pelechano, V., Vallecillo, A., Vara, J.M., Vicente-Chicote, C. Desarrollo de software dirigido por modelos: conceptos, métodos y herramientas. Alfaomega. 2014.