Declaratively building behavior by means of scenario clauses*

Fernando Asteasuain[1] and Víctor Braberman

[1] Universidad Nacional de Avellaneda, Dpto. Tecnología y Administración, Ing. en Informática, España 350, BsAs Argentina
[2] Dpto Computación-FCEyN,UBA, Ciudad Universitaria CABA, Argentina

Abstract. In this work we present the article "Declaratively building behavior by means of scenario clauses". This article was accepted in January 2016 in the journal "Requirements Engineering", ISSN: 0947-3602 (http://link.springer.com/journal/766). The publication is available at: http://link.springer.com/article/10.1007/s00766-015-0242-2.

Keywords: Requirements Engineering, Behavioral Modeling, Formal Specifications

1 Description of the article and contributions

Behavior needs to be understood from early stages of software development. In this context incremental and declarative modeling seems an attractive approach for closely capturing and analyzing requirements without early operational commitment. A traditional choice for such a kind of modeling is a logic-based approach. Unfortunately, in many cases, the formal description and validation of properties results in a daunting task, even for trained people. Moreover some authors established some practical limitations with temporal logics expressive power.

In the mentioned article we presented ω-FVS (Omega-Feather Weight Visual Scenarios) a declarative language, not founded on temporal logics, but on simple graphical scenarios, powerful enough to express ω-regular properties. The notation is equipped with declarative semantics based on morphisms and a tableau procedure is given enabling the possibility of automatic analysis.

We conducted three case studies based on different industrial protocol specifications to illustrate several aspects of ω–FVS such its expressive power, its ability to declaratively specify behavior, and the potential of the translation mechanism into Buchi automata. In this sense, we also developed an experiment based on the specification patterns to analyze the size of the automata generated by the tableau procedure. The goal of this experiment was to indirectly analyze whether the tableau yielded automata of reasonable quality -in terms of

* This work was partially funded by UNDAVCYT 2014, PAE-PICT-2007-02278:(PAE 37279), PIP 112-200801-00955 and UBACyT X021.

the costs that would incur a model-checking procedure by using those automata as property monitors.

One of the protocols studied was the .NET NegotiateStream Protocol [1]. In this case, the ω-FVS specification was extracted closely following the existing technical documentation of the protocol found at Microsoft Open Specifications [1]. In this example, besides modeling the behavior of the protocol in a declarative fashion using ω-FVS rules following a given specification document, we used the automata generated from these rules to detect relevant bugs in the specification of the protocol.

We believe these case studies in an industrial domain illustrate the main features of our language: its expressivity, flexibility, and the potential of our tableau procedure, and show the validity and applicability of our approach.

References

1. [ms-nns]: .net negotiatestream protocol specification v2.0. http://msdn.microsoft.com/en-us/library/cc236723.aspx, July 2008.