EARS-CTRL: The Demonstration

Levi Lúcio[1], Salman Rahman[1], Saad Bin Abid[1], and Alistair Mavin[2]

[1] fortiss GmbH
Guerickestraße 25, 80805 München
{lucio,abid}@fortiss.org, salman.rahman@tum.de
[2] Rolls-Royce, PO Box 31, Derby, UK
alistair.mavin@rolls-royce.com

Abstract. The demonstration part of the paper comprised of the following, 1) We have discussed a working case of sliding door controller, 2) a semi-automated process for requirements building and synthesizing of the the controller and 3) automatic generation of the test cases for the synthesized controller....

1 Introduction

EARS (Easy Approach to Requirements Syntax) is an effective technique utilized by many organizations (e.g., Rolce Royce and others) to build effective requirements [2]. In this demonstration part of the paper, we are going to go through steps for building the EARs-based requirements and performing test case generation for the sliding door example of a PLC based controller. Our tool is developed in Jetbrains MPS, a projectional meta editor for DSL development[3]. The work presented here is an extension to our previous work on building complex controller requirements and automatic generation of the specified requirements [1].

A brief installing information and links to the main github projects, URL links of the projects.

Discuss the main purposes of our work,that are,

- Progressively build a set of requirements for the controller
- Perform analysis to ensure correct building of the example case (whether the controller is synthesizable or not!!!) and correctness by construction...
- Automatic realisation of the synthesized controller (one click approach)
- Automatic test case generation for performing conformance analysis

Notes: Point can be arised that if you are generating code from the model why do we need tests? we can sell our idea by stating that we don't necessarly want to generate the code but write the controller requirements in EARS and somebody else can write the code. Some engineers don't rely on automatic code generation and want to develop/build controllers explicitly. In such a situation, test case generation would help to perform conformance between the specified controller and its respective implementation.

2 The Running Example: Automatic Sliding Door

Our running example for this demo is a sliding door system of PLC based controller. The controller for this system is shown in fig 1. The shown controller implements the following behavior,

- the sliding door opens if somebody enters by sensing the object using the infrared sensor (X0),
- the sliding door opens until the opening limits are reached, also detected by the sensor (X2),
- upon reaching the opening limits the count down timer starts and
- closes the door when the timer expires until the closing limit is reached detected by (X1).

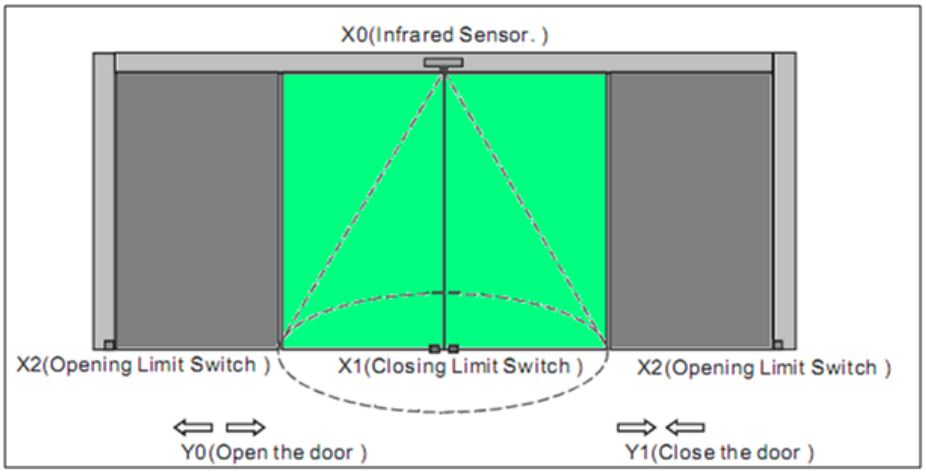

Fig. 1. A running example of Sliding Doors

3 Tool Demonstration

Discuss the main steps of process that needs to be followed for synthezing and performing the conformance analysis,that are,

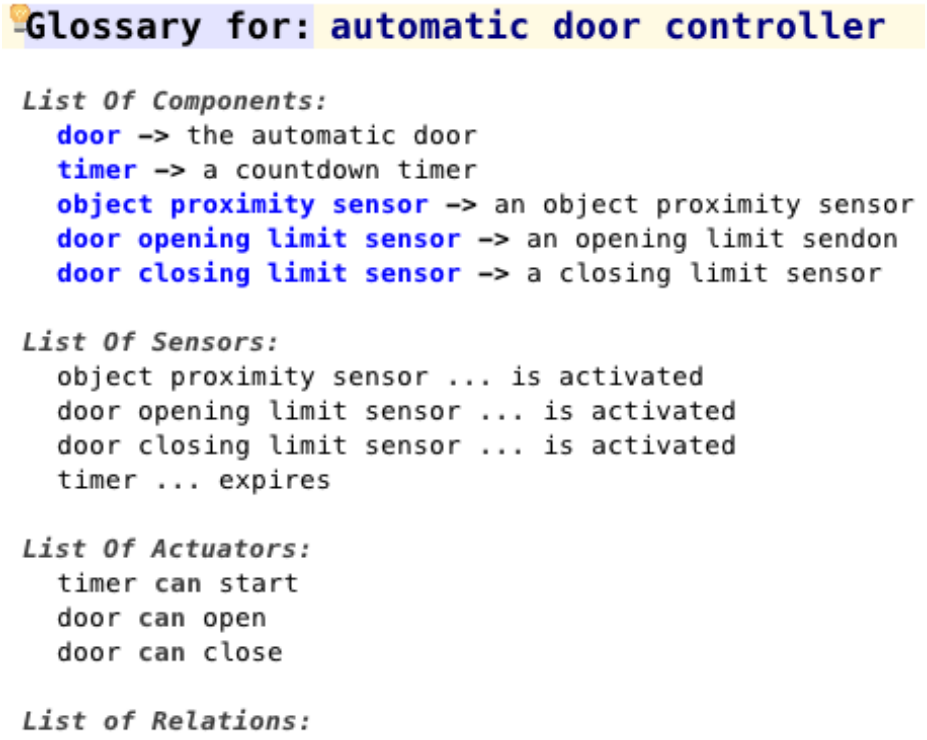

Fig. 2. Glossary building for sliding door controller

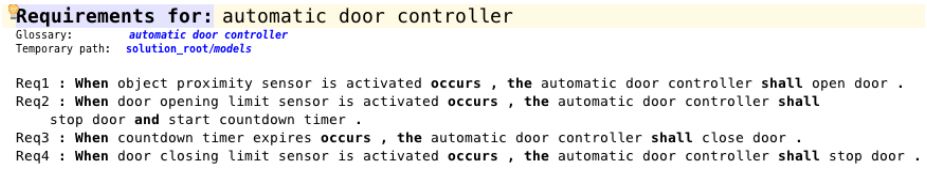

Fig. 3. EARS requirements for sliding door

3.1 Glossary building and terms definition

3.2 EARs-based requirements building for the controller

3.3 Synthesizing the EARs-based requirements

3.4 Test-case generation

4 Discussion

Discussion will go here...

Notes: Point can be arised that if you are generating code from the model why do we need tests? we can sell our idea by stating that we don't necessarly want to generate the code but write the controller requirements as EARS and somebody else can write the code. Some engineers don't rely on automatic code generation and want to develop/build controllers explicitly. In such a situation, test case generation would help to perform conformance between the specified controller and its respective implementation. More points: 4. Code generation and synthesizer for EARS-based requirements 5. Test case generation for conformance checking of the generated controller 6. Interfacing with Matlab Simulink 5. Viewing the Results 7. Lessons learned

8. Start discussing the steps as flow model and follow exactly the same steps!!!! 8.1 Expression of Controller Requirements as EARS and related Models (e.g., Glossary) 8.2 MPS constraints to ensure completeness of the EARS-based requirements (hint: discuss some examples if something breaks) 8.3 Test case generation process (step-by-step) that includes interfacing with simulink, test case generation sequences, showing the results of the results as inputs and output as the SimulinkResult model

Acknowledgements

This work presented in this paper was developed in the context of the "IETS3" research project, funded by the German Federal Ministry of Education and Research under code 01IS15037A/B.

References

1. Levi Lúcio, Salman Rahman, Chih-Hong Cheng, and Alistair Mavin. Just formal enough? automated analysis of ears requirements. In *9th NASA Formal Methods Symposium NFM 2017, California, USA, May 16 - 18, 2017*, 2017. To appear.
2. A. Mavin, P. Wilkinson, A. Harwood, and M. Novak. Easy approach to requirements syntax (ears). In *2009 17th IEEE International Requirements Engineering Conference*, pages 317–322, Aug 2009.
3. Vaclav Pech, Alex Shatalin, and Markus Voelter. Jetbrains MPS as a tool for extending java. In Martin Plümicke and Walter Binder, editors, *Proceedings of the 2013 International Conference on Principles and Practices of Programming on the Java Platform: Virtual Machines, Languages, and Tools, Stuttgart, Germany, September 11-13, 2013*, pages 165–168. ACM, 2013.