Berichte aus der Automatisierungstechnik

Doaa Soliman

Verification and Validation of Logic Control Safety Applications

Verifikation und Validierung von Steuerungssoftware für Sicherheitsanwendungen

Shaker Verlag
Aachen 2013

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Zugl.: Saarbrücken, Univ., Diss., 2012

Printed in Germany.

ISBN 978-3-8440-1757-1
ISSN 0945-4659

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen
Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9
Internet: www.shaker.de • e-mail: info@shaker.de

Summary

In this work, an approach was introduced to facilitate the verification process of safety applications built up from the PLCopen (2006) safety library. This approach is practiced with the help of safety applications used in real life, presented by PLCopen (2008). It was discovered that the verification approach is applicable and most helpful. This encouraged the researchers to think about the automation of the presented approach; More precisely, to automate the transformation process of a safety application to a formal model to be verified via the UPPAAAL model checker. However, many difficulties had to be faced in order to realise a transformation tool. Since the presented transformation approach is based on an XML platform, it is necessary to find an IEC 61131-3 programing tool that supports not only the PLCopen safety library, but also the exportation of the PLCopen XML scheme. Unfortunately, a qualified IEC 61131-3 programing tool did not exist at the time the research was undertaken. However, many software producers have future plans to support all required features. To overcome these difficulties, steps were taken to integrate the needed features in the partially qualified IEC 61131-3 tool. It was the MultiProg tool from KW-Software, which supported the exportation of PLCopen XML schema, but not the PLCopen safety library. This therefore, made it necessary to implement a user defined PLCopen safety library to be used in constructing safety applications with the MultiProg tool. This implemented safety library was then directly verified using the [mc]square model checker, which was joint work with the Embedded Software Laboratory (Prof. Kowalewski), RWTH Aachen University, Germany.

Finally, a qualified IEC 61131-3 programming tool was available. Consequently, an exported PLCopen XML from a safety application could be handled. As a first step to realising a transformation tool, meta-models of source and target XML domains are defined. Following this, transformation rules are formalised based on the meta-models. A prototype transformation tool is therefore developed and tested using some real safety applications. The next step is the formalisation of safety applications written in the FBD programing language and required UPPAAL systems. And of course, formalisation of detailed transformation rules is also defined. This led to developing the end version of the safety application to the timed automata SA2TA tool. This was a joint venture with the Software Engineering group (Prof. Thramboulidis) in Patras University, Greece.

As a case study to test the applicability of the transformation tool, the SA2TA is then used as part of a whole methodology to upgrade a legacy system to conform to safety standards. The legacy system needed to be upgraded is an XY drawing table located in the automation laboratory in Saarland University, Germany. The suggested methodology was applied to the XY table, and the designed safety application is transformed to a UPPAAL TA system for verification purposes. Therefore, the verification process is carried out based on safety functionalities defined through the designing stage. It was found that not all safety functionalities were satisfied on the UPPAAL system, which led to some modifications in the designed safety application to meet the required safety functionalities.

Some possible technical extensions to make the proposed methodology even easier to use are for example, the automatic transfer of simulation traces between PLC tools and UPPAAL in both directions. From PLC tools to UPPAAL for automatic validation and from UPPAAL to PLC tools for the visualisation of counter examples.

However, one question still remains. Since there is a gap between safety engineering and software engineering; who is responsible for applying verification processes on the resulting UPPAAL system from safety applications? It is believed that more effort is required in the direction of verification to minimise this gap and facilitate obtaining safety properties in formal languages.