Verification and Validation of Safety Applications based on PLCopen Safety Function Blocks using Timed Automata in Uppaal

Doaa Soliman and Georg Frey

Chair of Automation, Saarland University
University Campus Building A5.1, 66123 Saarbrücken, Germany
(e-mail: doaa.soliman@aut.uni-saarland.de, georg.frey@aut.uni-saarland.de)

Abstract: Functional Safety is a major concern in the design of automation systems today. Many of those systems are realized using PLCs programmed according to IEC 61131-3. PLCopen as IEC 61131 user organization specified a set of software Function Blocks to be used in Safety Applications according to IEC 61508 in 2006. The specification of Technical Committee 5 contains twenty Safety Function Blocks (SFBs) as a library together with some specifications of their use. A second part issued in 2008 demonstrates the use of the defined SFBs in real applications. In the presented work, formal models for the SFBs are derived from the semi-formal specification in the PLCopen documents. Those blocks are verified using model checking and the accordance of their temporal behavior with the PLCopen specification is further validated by simulation. The resulting library of formal models allows to build a formal model of a given safety application – built from SFBs – and to verify its properties. This is demonstrated using an example from the second part of the PLCopen specification.

Keywords: Safety Application, Timed Automata, PLC, Safety Function Block, IEC 61508, IEC61131-3 Verification and Validation, Model-Checking.

1. INTRODUCTION

Nowadays, PLCs are increasingly being used to implement safety functions for safety critical systems. One of the preferred languages in this area is Function Block Diagram (FBD) according to IEC 61131-3. There are many research projects in the field of verification and implementation of function blocks libraries according to this standard e.g. Völker and Krämer (2001) or Song et al. (2004). However, safety issues are not addressed in IEC 61131-3, and PLC programming software packages have only library function blocks dealing in general with communication, mathematical operations, logic, and so on. As a step for building safety applications in IEC 61131-3 FBD according to IEC 61508, PLCopen (2006) specifies a set of so-called Safety Function Blocks (SFBs). Several manufacturers of IEC 61131 programming tools have already implemented libraries according to this specification.

The main aim of the presented work is to ease the verification of safety applications built up using the PLCopen SFBs in one of the commercially available tools. To this end a modular (function block oriented) approach is taken. For all specified SFBs a corresponding formal model is built using Timed Automata (TA). To verify a safety application, the formal model of the application is derived by combining the previously specified TA in the same structure as the original SFBs are combined in the PLC software.

Before applications can be verified however, it has to be assured that the formal models are actually describing the behaviour of the SFBs correctly.

The description of SFBs by PLCopen contains three parts for each SFB respectively:

1. A graphical description of the internal states and behaviour using a state diagram.
2. A list of properties described in natural language.
3. Timing diagrams describing the temporal behaviour for some specific scenarios.

In the presented approach the graphical description (1st part) is translated into a TA in the language of the Uppaal tool. To validate the temporal behaviour, simulations of the model are performed and the results are compared to the timing diagrams (3rd part). The list of textual properties (2nd part) is formalized using temporal logic. By using model checking – see e.g. Berard et al. (2001) – it is then verified that these properties hold on the TA.

This paper is organized as follows: In the next section an overview of the proposed approach to verify safety applications is given. Section 3 describes the formalization of SFBs in detail using an example from the PLCopen Library. The use of the formalized SFBs for verification is demonstrated in section 4, utilizing an example from PLCopen (2008). The paper closes with a short summary and an outlook on further work.

2. APPROACH TO VERIFY SAFETY APPLICATIONS

The presented approach is based on the development process given in Fig.1. To build a safety application an engineer is assumed to build his software using a library of SFBs that