ABSTRACT

The chapter deals with reconfigurable embedded control systems following component-based technologies and/or Architecture Description Languages used today in industry. The author defines Control Components as software units to support control tasks of the system which is assumed to be a network of components with precedence constraints. The author defines an agent-based architecture to handle automatic reconfigurations under well-defined conditions by creating, deleting or updating components to bring the whole system into safe and optimal behaviors. To cover all reconfiguration forms, the agent is modelled by nested state machines such that states correspond to other state machines. Several complex networks can implement the system where each one is executed at a given time when a corresponding reconfiguration scenario is automatically applied by the agent. To check the correctness of each one of them, we apply in several steps a refinement-based approach that automatically specifies feasible Control Components according to NCES. The model checker SESA is automatically applied in each step to verify deadlock properties of new generated components, and it is manually used to verify CTL-based properties according to user requirements. The author implements the reconfiguration agent by three modules that allow interpretations of environment evolutions, decisions of useful reconfiguration scenarios and finally their applications. Two Industrial Benchmark Production Systems FESTO and EnAS available in the author’s research laboratory are applied to explain the paper contribution.
INTRODUCTION

The development of safe and critical embedded control systems is not an easy activity because they have classically to satisfy functional and to meet temporal properties defined in user requirements, in addition to their time to market that should be more and more shorter than ever (Goessler, Graf, Majster-Cederbaum, Martens, & Sifakis, 2007; Lee & Villasenor, 2009; Acharya & Mahapatra, 2008; Song, Kim & Karray, 2008; Tsui, Masmoudi, Karray, Song & Masmoudi, 2008; Pagilla, Dwivedula & Siraskar, 2007). To meet all these constraints, different component-based technologies (Crnkovic & Larsson, 2002) and Architecture Description Languages (abbr. ADL) (Dissaux, Filali Amine, Michel, & Vernadat, 2005) have been proposed last years to reuse already developed components and to follow modular approaches for specification and verification. The Architecture Description Languages define components as independent software units of computations to be connected by well-defined connectors (McKenzie, Petty, & Xu, 2004). ADL examples include Darwin, Aesop, Unicon, Wright, Rapide, Acme, AADL, UML, etc (Dissaux, Filali Amine, Michel, & Vernadat, 2005). Among all industrial technologies, the International Standard IEC61499 (IEC61499-1, 2003; Khalgui & Hanisch, 2008) is a Component-based technology extending the previous well-known Standard IEC61131 which used for developments of Centralized Programmable Logic Controllers (IEC61131, 1993). This technology defines a Function Block as an event triggered component owning data and composed of an interface as well as an implementation. The interface contains data/event inputs and outputs for external interactions with the environment. Events are responsible for activations of the block while data contain valued information. The implementation consists of internal data and algorithms to implement block’s functionalities. Today in academia and also in industry, rich books have been written (Vyatkin, 2007), several research works have been made (Khalgui & Thramboulidis, 2008; Khalgui, Rebeuf, & Simonot-Lion, 2007; Khalgui & Rebeuf, 2008), useful tools have been developed (Rockwell, 2006) and various manufacturing platforms are completely deployed according to this technology: (i) a footwear factory that we simulated in (Hanisch, Khalgui & Carpanzano, 2008) is developed in Italy while following the Function Block technology, (ii) two Benchmark Production Systems FESTO and EnAS are developed according to this standard in our research laboratory at Martin Luther University in Germany. We present in Figure 1 a simple example of an IEC61499 control application named Temperature Regulator to regulate the temperature of an oven in a factory (detailed descriptions are available in (Lewis, 2002)). This regulator is composed of two interface Function Blocks (Input1 and Output1) and a regulation block PID1 which reads data from a sensor before regulations of the temperature (by applying the well-known PID algorithm (Ben Jmaa Derbel, 2007),) and activations of the corresponding actuator.

The Carnegie Mellon university proposes also its own concept of components named Port Based Objects to develop industrial control applications in robotics (Stewart, Volpe & Khosla, 1997). A PBO component is a particular case of a Function Block by containing three different interfaces as well as an implementation. The main first interface contains data inputs and outputs for external interactions with other components. A second parameterization interface is used to update internal data and algorithms. Finally, a third interface is used for interactions with physical processes. We present in Figure 2 Speed Regulator as a simple example of PBO components to regulate the speed of a vehicle. The component cyclic periodically sends desired values to Regulate which regulates measured values from Interface (Stewart, Volpe & Khosla, 1997). We note finally that a rich library is available today to develop applications following this industrial technology (Stewart, Volpe & Khosla, 1997).