Chapter 10
Reconfigurable Embedded Control Systems

Mohamed Khalgui
Xidian University, China

Olfa Mosbahi
Martin Luther University, Germany

ABSTRACT
The chapter deals with distributed multi-agent reconfigurable embedded control systems following the component-based International Industrial Standard IEC61499 in which a Function Block (abbreviated by FB) is an event-triggered software component owning data and a control application is a distributed network of Function Blocks that have classically to satisfy functional and to meet temporal properties described in user requirements. The authors define a new reconfiguration semantic where a crucial criterion to consider is the automatic improvement of the system’s performance at run-time, in addition to its protection when hardware faults occur. To handle all possible cases in industry, the authors classify thereafter the reconfiguration scenarios into three forms before the authors define an architecture of reconfigurable multi-agent systems where a Reconfiguration Agent is affected to each device of the execution environment to apply local reconfigurations, and a Coordination Agent is proposed for any coordination between devices in order to guarantee safe and adequate distributed reconfigurations. A Communication Protocol is proposed in our research work to handle coordinations between agents by using well-defined Coordination Matrices. The authors specify both the reconfiguration agents to be modelled by nested state machines, and the Coordination Agent according to the formalism Net Condition/Event Systems (Abbreviated by NCES) which is an extension of Petri nets. To verify the whole architecture, the author check by applying the model checker SESA in each device functional and temporal properties described in the temporal logic “Computation Tree Logic”, but the authors have also to check any coordination between devices by verifying that whenever a reconfiguration is applied in a device, the Coordination Agent and other concerned devices should react as described in user requirements. The chapter’s contributions are applied to two Benchmark Production Systems available in our research laboratory.

DOI: 10.4018/978-1-60960-086-0.ch010
INTRODUCTION

Nowadays in industry, the development of safety distributed embedded systems is not a trivial activity because a failure can be critical for the safety of human beings (e.g. air and railway traffic control, nuclear plant control, aircraft and car control). They have classically to satisfy according to user requirements, functional and temporal properties (Baruah, & Goossens, 2004), but their time to market should be shorter than ever. To address all these important requirements, the component-based approach is studied in several academic research works and also in industrial projects to develop modular embedded systems in order to control the design complexity and to support the reusability of already developed components (Goessler & Sifakis, 2002). Several component-based technologies have been proposed in industry to design the application (as a composition of components) (Crnkovic & Larsson, 2002). Among all these technologies, the International Standard IEC61499 is proposed by the International Electrotechnical Commission to design distributed control applications as well as corresponding execution environments (IEC61499, 2004). A Function Block in this Standard is an event-triggered software component composed of an Interface and an Implementation where the interface contains data/event inputs/ outputs for external interactions with the environment. Events are responsible for the activation of the block, whereas data contain valued information. The Implementation of the block contains algorithms to execute when corresponding events occur. The selection of an algorithm to execute is performed by a state machine called Execution Control Chart (ECC) which is also responsible for sending output events at the end of the algorithm execution. An IEC61499 application is therefore a network of blocks that should meet functional as well as temporal properties defined in user requirements. Today in academia and industry, rich books have been written (Lewis, 2001), many research works have been made, useful tools have been developed and finally real industrial platforms have been deployed while following this International Standard. In our research laboratory at Martin Luther University in Germany, two Benchmark Production Systems (FESTO and EnAS) are completely developed according to this component-based technology.

The new generation of industrial control systems is addressing today new criteria as flexibility and agility. To reduce their cost, these systems should be changed and adapted to their environment without disturbances. Several interesting academic and industrial research works have been made last years to develop reconfigurable control systems (Gehin & Staroswiecki, 2008). We distinguish in these works two reconfiguration policies: static and dynamic reconfigurations where static reconfigurations are applied off-line to apply changes before the system cold start (Angelov Sierszecki & Marian, 2005), whereas dynamic reconfigurations are applied dynamically at run-time. Two cases exist in the last policy: manual reconfigurations applied by users (Rooker, Sunder, Strasser, Zoitl, Hummer & Ebenhofer, 2007) and automatic reconfigurations applied by Intelligent Agents (Al-Safi & Vyatkin, 2007). We are interested in this chapter in automatic reconfigurations of industrial control systems following the International Standard IEC61499. We define at a first time a new reconfiguration semantic where a crucial criterion to consider is the automatic improvement of the system performance at run-time, in addition to its protection when hardware faults occur. To explain in detail this semantic, we give examples of automatic reconfigurations to be applied to the Benchmark Production Systems FESTO and EnAS. To cover all possible reconfiguration cases in industry, we define a classification of scenarios into three forms: the first deals with the system architecture, the second with the internal structure of blocks or with their composition, finally the third deals with the reconfiguration of data. To handle all possible reconfiguration
Related Content

A Service-Oriented User Interface for the Next Generation Web
[www.igi-global.com/article/service-oriented-user-interface-next/55123?camid=4v1a](www.igi-global.com/article/service-oriented-user-interface-next/55123?camid=4v1a)

[www.igi-global.com/chapter/t-way-testing-strategies/108628?camid=4v1a](www.igi-global.com/chapter/t-way-testing-strategies/108628?camid=4v1a)

An Evaluation of a Pure Embedded Domain-Specific Language for Strategic Term Rewriting
[www.igi-global.com/chapter/evaluation-pure-embedded-domain-specific/71817?camid=4v1a](www.igi-global.com/chapter/evaluation-pure-embedded-domain-specific/71817?camid=4v1a)

Towards Method Component Contextualization
[www.igi-global.com/article/towards-method-component-contextualization/58645?camid=4v1a](www.igi-global.com/article/towards-method-component-contextualization/58645?camid=4v1a)