Loose Integration of COTS Tools for the Development of Real Time Distributed Control Systems

Javier Portillo, Oskar Casquero, and Marga Marcos

Escuela de Ingenieros de Bilbao (University of the Basque Country)
Alameda Urkijo s/n, 48013 Bilbao, Spain
jtppobej@bi.ehu.es, cvzcaoio@lg.ehu.es, jtpmamum@bi.ehu.es

Abstract. The development of Real Time Distributed Control Systems (RTDCS) is a very complex and multi-part issue where different specific tools are to be used. As these specialized tools are not designed to work together, it would be desirable to have a flexible tool framework where all the information were managed and stored following a predefined Model Driven Architecture. XML technologies and Web Applications (implemented as a component-based multi-tier application design defined by J2EE) have been selected to put into practice such a framework. It is proposed a model-based approach to develop software systems that require the collaboration of specific tools. This collaboration is achieved thanks to a Tool Collaboration Engine based on XML and Web Applications. A prototype of the framework was built for RTDCS, yet these concepts can easily be applied to any area of knowledge. The paper presents some conclusions on the integration of COTS.

1 Introduction

The development of Real Time Distributed Control Systems (RTDCS) is a very *complex and multi-part* issue implied in the generation of *heterogeneous applications* with *changing needs*. It is a *complex* topic because several phases are involved: requirements gathering, specification, design, simulation, analysis and code-generation. It is a *multi-part* matter because several knowledge areas are implied. The suitability of different concepts such as control algorithms, network communications and real-time constraints must be considered as a whole. Very *heterogeneous applications* are produced because Real Time or embedded applications vary in size and scope; from microwave ovens to factory automation, aerospace industry or railway control. Finally, the development of RTDCS must progress with these *changing needs* and one cannot ignore the support of new control algorithms, HW devices, network protocols, programming languages, operating systems, temporal constraints, etc.

Therefore, different specialists (control engineers, real time experts, software engineers) should work together and understand each other's needs. Each expert is assisted by domain specific tools (mainly COTS, Commercial Off The Shelf tools), but there is a gap in connecting specific tools from different domains and, therefore connecting different (but inter-related) knowledge areas.

X. Franch and D. Port (Eds.): ICCBSS 2005, LNCS 3412, pp. 191 – 200, 2005.

As new features and tools are constantly added, software vendors adopt a tool integration approach. The current market offers well-known software packages for system analysis and simulation that allow some exchange of information between the tools that cover the design phase. For instance, the two packages Xmath and Statemate [4] can be linked together at the code level and an interface allows the joint simulation within the Statemate environment. The Matlab / Simulink / StateFlow environment [11] offers graphical tools that allow cooperative simulation of models edited in Simulink and Stateflow. Some of these COTS tools allow code generation but they only consider a single design domain and further code is necessary to support network communication, signal conditioning, input/output data checking, fault detection, isolation and accommodation, etc.

Previous work of the authors [13] proposed a Matlab-based Framework for the integration of all the phases involved in the design of Real-Time Distributed Control Systems (RTDCS). Nevertheless, this framework was based in Matlab programming environment. A more general open framework is required to integrate COTS tools from different fields of expertise so they can work together in the generation of the RTDCS application.

Portillo [14] presented a Model Driven Framework aimed at the integration of COTS tool used in the development of Real Time Distributed Control Systems. This approach is based on Domain Specific Models; they must be understood as formal descriptions (of the system to develop) from the point of view of different knowledge areas. The key element of this approach is the *Model Collaboration Engine* (MCE) that stores, manages and coordinates the information of different models. A *Tool Collaboration Engine* (TCE) links COTS tools to those models handled by the MCE. Integration is achieved by means of data sharing because particular tools read/write information automatically from/to the Domain Specific Models. The major advantage here is specialists still use their own tools but the framework feeds them with results obtained from other domain specific tools (see Fig. 1). The TCE offers an open and standard interface to COTS thanks to XML and Web Services. The initial findings in tool and model integration of some open source projects (like Eclipse [2]) are very promising efforts and support the results presented here.

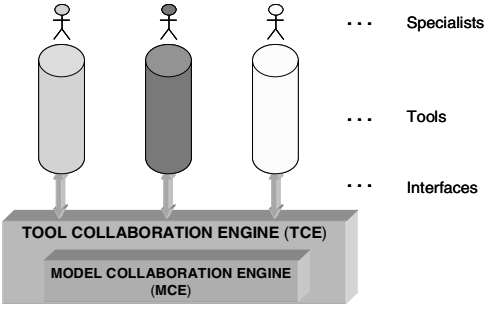

Fig. 1. TCE (Tool Collaboration Engine) embodies the MCE (Model Collaboration Engine)