A SCHEDULABILITY ANALYSIS OF AN IEC-61499 CONTROL APPLICATION

Mohamed Khalgui, Xavier Rebeuf, Françoise Simonot-Lion

LORIA - INPL
Campus Scientifique, BP 239
54506 Vandoeuvre-lès-Nancy - France
{khalgui, rebeuf, simonot}@loria.fr
tel: +33 3 83 58 17 28, fax: +33 3 83 58 17 01

Abstract: This paper deals with the temporal correctness of control applications designed using the component-based standard IEC 61499. In this standard, a function block is defined as an event trigger component containing its own data. To validate the temporal behavior of an application, we have to take into account its scheduling on the execution support. We propose an abstraction of the component behavior taking into account all its possible executions. Moreover, we propose to verify the temporal correctness of the application with regard to global temporal properties (end to end delays). Thanks to such characterization, we show that it is possible to check deadlines for the application to ensure its correctness. To reach this goal, we transform the application into a dependant tasks model(Copyright © 2005 IFAC).

Keywords: Function Blocks, IEC 61499, Real Time, schedulability analysis, offline scheduling.

1. INTRODUCTION

Industrial control applications need to satisfy not only functional properties but also temporal ones. To validate a priori the correctness of such applications, one of the key issues is to model their temporal behaviors. To manage the design complexity, several methods based on components have been proposed (Pecos 2005, holobloc 2005, Stewart *et al.* 1997, Articus 1996). In such approaches, evaluating temporal behavior requires also a model of the execution support.

In the control systems field, the IEC 61499 standard (WG6 2003, WG6 2004) is a component-based methodology allowing to design applications as well as the execution support (Crnkovic and Larsson 2002). In the standard, the Function Block is defined as an event trigger component (Lewis n.d.). It is a reusable functional unit of software owning data. A control application is specified by a “function blocks network" which can be distributed on one or more devices. Several complex applications have been specified using function blocks (Ifak 2005). Moreover, reusable function blocks libraries already exist (holobloc 2005).

The standard allows to validate static interoperability between blocks. Nevertheless, temporal behavior depends on dynamic data. Therefore, it is difficult to a priori validate temporal interoperability. We propose an abstraction of each block behavior taking into account all its possible executions. Such abstraction allows to compute an upper bound of the application execution time. We show that it is possible to verify the temporal correctness of the application with regard to global temporal properties (end to end delays).

To validate temporal interoperability, we propose a schedulability analysis of a function blocks network distributed on one device. To perform such analysis, we propose to transform the function blocks application into a particular dependent tasks model in the order to take advantages of the results in this field.

Some works are proposed about schedulability analysis of systems in one processor. In the non-preemptive case, (Cucu *et al.* 2002) presents an optimal algorithm to schedule systems with precedence, periodicity and latency constraints (end to end deadline). In addition, (Cucu and Sorel 2003) proves the existence of a hyper-period T for synchronous systems allowing the proposed algorithm to find a schedule applying it from 0 to T, instead of from 0 to ∞. On another side, (Jeffay *et al.* 1991) gives a schedulability condition for a set of periodic and sporadic operations with arbitrary release times. (Howell and k. Venkatrao 1995) studies the complexity for the problem of non-preemptively scheduling of periodic and sporadic tasks on one processor using inserted idle times. Using part of this result, we show that it is possible to generate a safe off-line scheduling for a function blocks network. Moreover, the proposed schedulability analysis allows to define a strategy for a function block to adapt its internal behavior according to the scheduling.

In the next section 2, we briefly present the IEC 61499 standard. Then we present our behavior characterization of an application. The section 4 deals with its transformation into a tasks model. In the section 5, we propose a generation of tasks deadlines according to the application delays. Finally, we present a schedulability analysis based on the generated deadlines.

2. THE IEC 61499 STANDARD

We present the main concepts of the IEC 61499 Function Blocks standard (WG6 2003, WG6 2004). This standard is an extension of the IEC 61131.3 (IEC61131-3 1993) for the Programmable Logic Controllers. We can divide its description into two parts: the architecture description and the block behavior through the events selection mechanism.

2.1 Architecture description

An application function block (FB) (figure 1) is a functional unit of software that supports some functionalities of an application. It is composed by an interface and an implementation. The interface contains data/event inputs and outputs supporting the interaction with the environment. Events are responsible for the activation of the function block while data contain valued information.

The implementation consists of a body and a head. The body is composed of internal data and algorithms implementing the block functionalities. Each algorithm gets values in the input data channel and produces values in the output data ones. They are programmed in structured text (ST) language (IEC61131-3 1993).

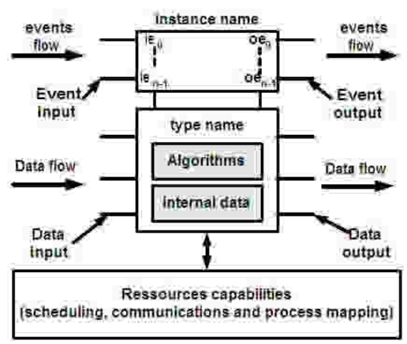

Figure 1. An IEC 61499 function block

The block head is connected to event flows. It selects the sequence of algorithms to execute with regard to an input event occurrence. The selection mechanism of an event occurrence is encoded in a state machine called the Execution Control Chart (ECC). At the end of the algorithms execution, the ECC sends the corresponding output event occurrences.

In the standard, a function blocks network defines the functional architecture of a control application. Each function block event input (resp. output) is linked to an event output (resp. input) by a channel. Otherwise, it corresponds to a global application input (resp. output). Data inputs and outputs follow the same rules (figure 2).

The execution support architecture (i.e. the industrial control system) is defined by a devices network. A device is composed of one processing unit,