Building Closed-loop Models for Discrete Controller Design (Extended Abstract)

Hans-Michael Hanisch
Chair of Automation Technology
Martin Luther University of Halle-Wittenberg
06099 Halle (Saale)
Germany
aut.informatik.uni-halle.de
hans-michael.hanisch@informatik.uni-halle.de

This contribution is dedicated to design of discrete controllers in manufacturing at the sensor/actuator level as a subset of discrete controller design.

Closed-loop modeling is the core issue of automatic control and makes it a unique science. It must be therefore a major concern in the community dealing with discrete controller design to come up with closed-loop models as well.

In automation technology in general, any controller is just a means to realize some desired behavior of the controlled object (plant). It is therefore only natural that deep understanding of the plant is the basis of any controller design. Also the specifications are given in terms of the plant.

This might be different in other fields of discrete controller design problems, as, for example, in aerospace and automotive industry, in telecommunication and others, but if we come back to manufacturing, where the plant is dominating, it is questionable whether a formal model of just the controller in open loop is a sound basis for design and verification.

It is often misunderstood what a controller actually is. What we currently see is a tendency that researchers coming from computer science getting into this field without a real deep understanding of control.

In fact, here are three types of phenomena that must be included into a closed-loop model. They come from different “worlds”:

1. The “natural world” including phenomena of heat and mass transfer, mechanical phenomena etc. that cannot be arbitrarily changed due to the wishes on a human being, as - for example - the chemical processes within a reactor. Such phenomena are created by “Mother nature”, and they are yet not understood in detail.
2. The “constructed world” including applied phenomena taken from the “natural world”, as - for example - electrical machines, conveyors, pneumatic valves and cylinders etc.
3. The “programmed world” including any digital data processing device, as - for example - Personal Computers, controllers etc.

What makes automation technology somehow “unique” is the fact that phenomena of all three “worlds” must be taken into account to perform a controller design process correctly.

When it comes to controller design, experts with knowledge of all three worlds have to collaborate and therefore to communicate. This has been - and is still - the major problem of controller design. It is often misunderstood - also in the computer science community - that what is designed to act as a controller is merely a means - and nothing more - to achieve a desired behavior of the plant. The author of this presentation has been active in this field for more than two decades and has often seen the lack

Figure 1: The Framework

of communication and the consequences that arose from this gap.

In general, the situation in continuous control is not that frustrating as in discrete control. The reason is that almost all of the experts in continuous control share a common model, namely algebraic and differential equations coming from the classical fields of mathematics.

Things look completely different in discrete event control. Although it is often claimed that Automata and Petri nets would play the same role in discrete controller design as the classical models in continuous control, it is questionable whether this is really the case, at least at the sensor/actuator level.

From an engineering point of view, it is highly desirable to have a methodology along with appropriate formal modeling mechanisms and design technologies at hand, that allow model-based engineering rather than a design from scratch. They should allow such features as the following ones.

- real modularity
- hierarchy
- compositionality
- reusability

Furthermore, the models should clearly identify the structure of the plant and the interfaces between plant and controller(s). If we look into reality, plant

and controller(s) do not interact by flow of tokens (as it is often assumed in Petri net models) or by common events or shared variables (as it is often assumed in automata models). It sounds trivial, but in reality plant and controller(s) interact via boolean or integer-valued data from/to sensors and actuators. It therefore should be obvious to any engineer with some basic knowledge of control systems design that the models must provide a concept of input and output signals.

This presentation therefore proposes a way of modeling that is orientated on a block-diagram way of modeling that is common to any control engineer. Encapsulation of dynamic behavior inside the modules is ensured. The interactions between modules are only accomplished by means of signals, as it is usually the case in automation technology.

Such a formalism has been developed in the group of the author and was called Net Condition/Event systems, Signal/Event nets etc. The formalism provides two kinds of input and output signals, namely condition signals that carry state information and event signals that carry state transition information. Composition is performed by connecting input and output signals of the same type of several modules.

Based on this modeling formalism, controller verification in closed loop is accomplished (see Figure 1).

Figure 2: A Workbench Designed for Closed-loop Modeling and Verification

First, a model of the plant behavior is generated semi-automatically using a library of plant component models. The only model components that must be manually added are the work piece properties since these are the unique components characterizing the specific plant and the process it performs.

Second, a model of the controller(s) is generated directly from the control code.

Third, plant and controller models are composed to a model of the closed-loop behavior.

Forth, safety as well as process specifications are generated using tools that were developed in the author's group. These tools translate the specifications into CTL formulas.

With these prerequisites at hand, one can do model-checking to verify the specifications in closed loop.

The method works for controller design following both, the IEC 61499 and the IEC 61131 International Standard.

Figure 2 gives some illustration on what is possible up to now by means of several tools that have been developed in the group of the author.

It is not claimed that the proposed methodology is able to solve any problem - especially when it comes to hybrid models with high complexity. It is, however, a formalism that enables communication between experts of different fields and supports model engineering in a component-based way.

The major focus of the presentation is the systematic modeling of the closed-loop behavior. The presentation will critically review the current state of the methodologies and will discuss the weaknesses. Based on this discussion, some further directions for research and development are given. This is intended to go beyond the work of the author's workgroup and provide some generalization on further development in this field.

Examples of real manufacturing systems in laboratory scale illustrate the contents throughout the whole presentation so that even someone who is not an expert in this particular field can - hopefully - easily understand what the presentation is about.