DISCRETE EVENT SYSTEMS MODELING AND CONTROL OF A MANUFACTURING TESTBED

R. J. Leduc and W. M. Wonham

Systems Control Group
Department of Electrical and Computer Engineering
University of Toronto
Toronto, Ont., M5S 1A4
email: leduc@control.toronto.edu, wonham@control.toronto.edu

Abstract - In this paper, we describe the manufacturing testbed we have built to investigate the implementation of RW supervisors on Programmable Logic Controllers (PLC). We discuss the modeling of the testbed and the design of its controllers. Finally, we present several theorems for verifying controllability and non-blocking on large systems.

I. Introduction

With current technology, we are capable of building extremely large and complex systems. It is becoming increasing difficult to design controllers that guarantee the proper operation of such systems. For Discrete Event Systems, the theory developed by Ramadge and Wonham ([4] and [7]) - referred to as "RW supervisory control theory" - enables us to design controllers for a given system that are mathematically guaranteed to be always controllable and to never deadlock. This is highly desirable in systems that are safety-intensive.

Our current work addresses the conversion of a theoretical supervisor to a physical implementation on a Programmable Logic Controller (PLC). This has been investigated previously by Brandin ([2] and [1]). To this end a testbed that physically simulates a manufacturing workcell was built around an Allen-Bradley PLC. In the remainder of this paper we describe the modeling of the testbed, and the design of its controllers. Finally, we present several results that identify reduced plant models for verifying controllability and nonblocking.

II. Description of Testbed

The testbed is designed to simulate a manufacturing workcell, in particular, problems of routing and collision. The configuration of the testbed is similar to a portion of the product routing design of the Motorola Fusion Factory [6]. Figure 1 shows a block diagram of the testbed.

The testbed is composed primarily of model railroad components. The tracks are laid out to resemble a set of three interacting work units. The two trains simulate Automated Guided Vehicles (AGV) that provide and remove material to/from each manufacturing unit. Each unit has a small electric crane to simulate robots loading and unloading the AGV. Finally, the system contains several remote track switches (6) to control the paths of the trains, plus sensors (26) that detect the presence and identity of each train.

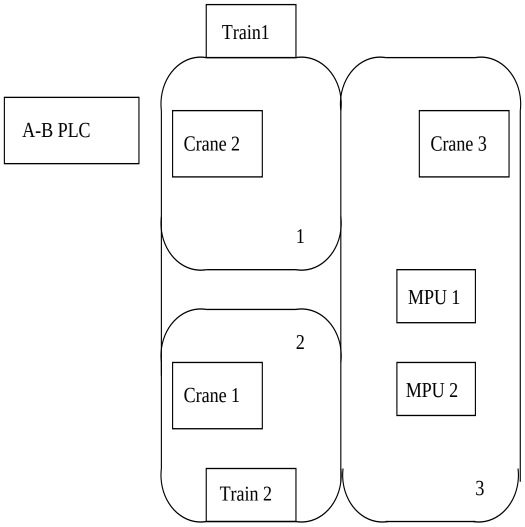

Figure 1: Layout of Testbed

The testbed is controlled by two embedded MC68332 processors and an Allen-Bradley PLC. The microprocessors control the trains and cranes directly, plus they have a discrete interface for communication with the PLC. The PLC is responsible for the control of the overall system. For more information on PLC, refer to [3] and [5].

III. Plant Model

The plant model for the testbed consists of the following basic elements: trains, cranes, sensors and switches. They are shown in Figure 2. These models are augmented by additional specifications that define how the elements interact with one another. In total, there are 108 modular specifications.

The interaction models show the interdependencies of

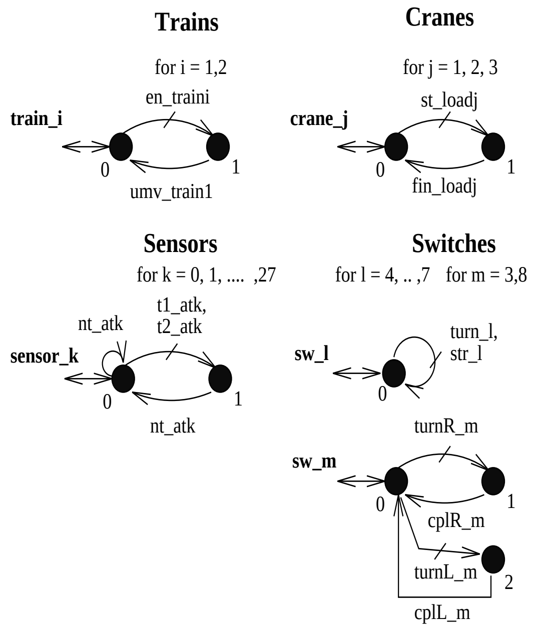

Figure 2: Fundamental Plant Models

the basic models. The first series of models show how the sensors are dependent on train movement. If all trains are stationary, then every sensor is disabled. This is depicted in Figure 3. The next series of models is the sensor's interdependencies. With respect to the starting position of a particular train, sensors can only be reached in a particular order, dictated by their physical location on the testbed.

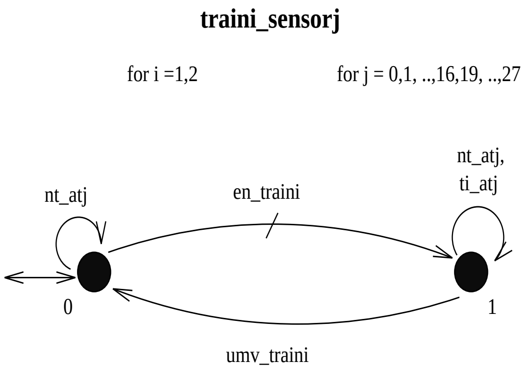

Figure 3: Sensor Dependency for Trains

The next series of models detail the sensor's dependency on the switches. Depending on the current location of the trains, certain sensors are inaccessible if the switch prevents the train's approach. An example of this model is given in Figure 4. The figure shows how adjacent sensors are affected by the current position of switch 6, with respect to recognizing train 2.

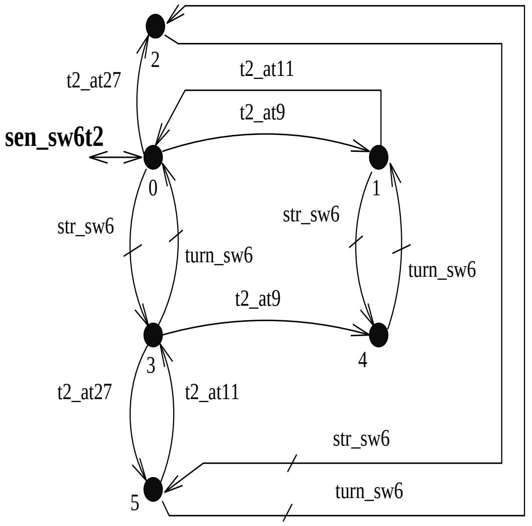

Figure 4: Sensor Dependency for Switch 6

The final series of models is slightly different from the previous series. This series, the switch request handlers, manages how the controllers interact with the switches. The handlers disable the switches' activation events until they are specifically requested. The request handler for switch 3 is depicted in Figure 5.

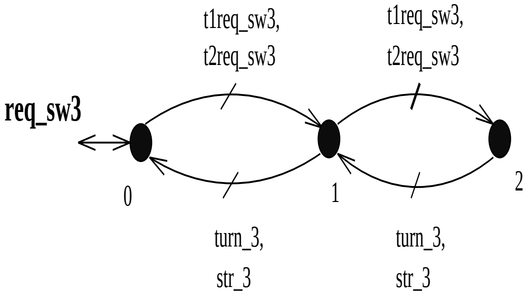

Figure 5: Switch Request Handler

IV. Control Specifications

The testbed's safety and operation specifications are as follows:

1. Prevent trains from colliding
2. Ensure switches are set correctly while train traverses them
3. Enforce specified route
4. Synchronize trains to permit loading by cranes

The route for train 2 is as follows: traverse loop 2, loop 1, loop 3 and then repeat indefinitely. The route for train 1 is similar.

V. DES Supervisors

After the plant model was developed, 29 modular supervisors were designed to enforce the given control specifications. The controllers work as follows. The routing controllers track the progress of each train and activate track switches when necessary. Each supervisor manipulates one switch for a particular train. The overall switching pattern generates the train's path. Figure 6 shows the routing controller for switch 7, train 2.

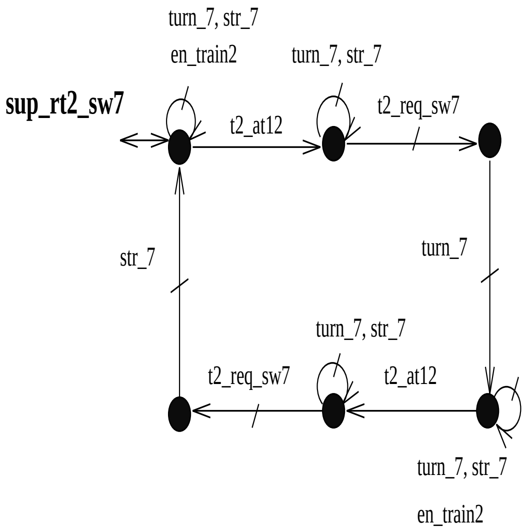

Figure 6: Routing Controller

A crane controller monitors its immediate sensor for a train. When a train arrives, the controller stops it and activates the crane, which then proceeds to load the train. Upon completion of the train's task, the supervisor releases the train. The crane controller for crane 1 is shown in Figure 7.

The final class of supervisors are the collision protection controllers. Each track section has its own controller that essentially permits only one train to occupy that portion of the track at any given time. Figure 8 shows the collision controller for the track section containing sensors 23 and 24.

VI. Model Reduction Theorems

After modeling the testbed, it was quickly apparent that conventional methods would be ineffective due to the large size of the composite model (on the order of 10^{16} states). To handle this, we developed several theorems for verifying controllability and nonblocking on reduced models of the plant. These are stated below.

Let the plant be composed of $n \geq 1$ plant models.

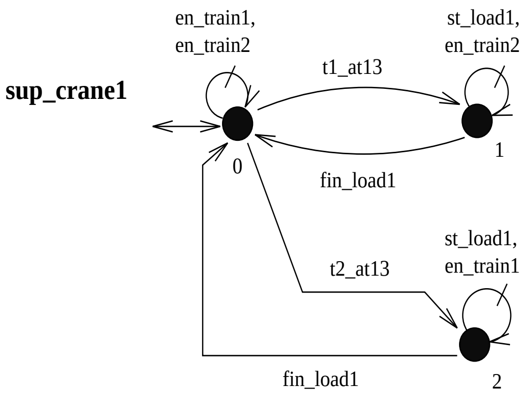

Figure 7: Supervisor for Crane 1

Let $I = \{1, 2, \ldots, n\}$ be an index set for the n plant models. Define the n plant models as follows:

$G_i = (\Sigma_i, Y_i, \eta_i, Y_{o_i}, Y_{m_i}), i \in I$

$Plant = sync(G_1, G_2, \ldots, G_n) = (\Sigma, Y, \eta, Y_o, Y_m)$

Let $J = \{1, 2, \ldots, m\}$ be an index set for the $m \geq 1$ DES supervisors for plant. Define the m supervisors as follows:

$S_j = (\Sigma, X_j, \xi_j, X_{o_j}, X_{m_j}), j \in J$

Definition: A sub-plant, G_{sub}, is any group formed from the n models that define Plant. Let $I_{sub} \subseteq I$ be the index set that represents the sub-plant.

Formally, $G_{sub} = sync(G_{i_1}, G_{i_2}, \ldots, G_{i_k})$, $I_{sub} = \{i_1, i_2, \ldots, i_k\}$, $k \leq n$.

Definition: Arbitrary events for supervisor S_j, $j \in J$, are events self-looped at every state in the supervisor. Let $\Sigma_{arb} \subseteq \Sigma$ represent these events.

Formally, for $\sigma \in \Sigma$,

$\sigma \in \Sigma_{arb_j}$ iff $(\forall x_j \in X_j)\, \xi_j(x_j, \sigma)! \wedge \xi_j(x_j, \sigma) = x_j$

Definition: Essential events for supervisor S_j, $j \in J$, are those events in Σ not in Σ_{arb_j}. Let $\Sigma_{ess_j} = \Sigma - \Sigma_{arb_j}$ represent these events.

Definition: Plants G_{i_1} and G_{i_2}, i_1 and $i_2 \in I$, are decoupled if they have no common events ($\Sigma_{i_1} \cap \Sigma_{i_2} = \emptyset$).

Definition: A fundamental plant for supervisor S_j, $j \in J$, is a minimal (in terms of index set size) sub-plant that contains every event in Σ_{ess_j} ($\Sigma_{ess_j} \subseteq \Sigma_{fund_j}$) and is decoupled from the remaining sub-plant. Let G_{fund_j} represent such a sub-plant and let $I_{fund_j} \subseteq I$ be its index set. Let G_{rem_j} represent the corresponding remaining sub-plant and $I_{rem_j} = I - I_{fund_j}$ be its index set.

Definition: A candidate plant for supervisor S_j, $j \in J$, is any sub-plant that contains every event in Σ_{ess_j} ($\Sigma_{ess_j} \subseteq \Sigma_{cand_j}$). Let G_{cand_j} denote such a sub-plant and let $I_{cand_j} \subseteq I$ be its index set.

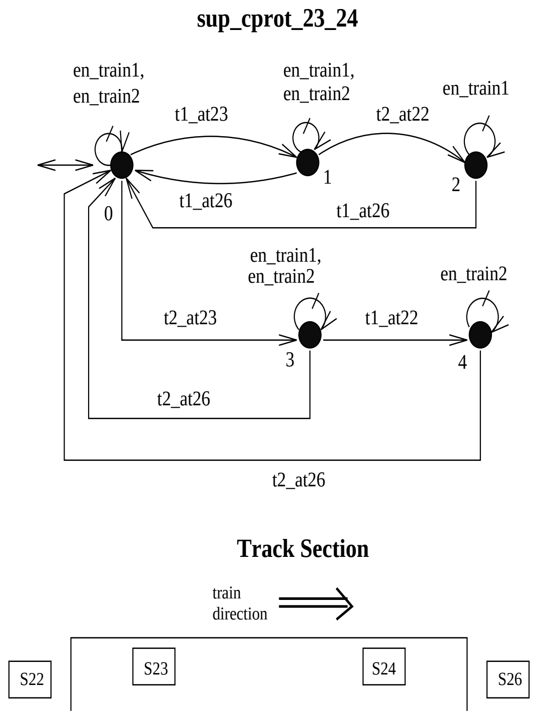

Figure 8: Collision Protection Supervisor

Definition: The essential plant for supervisor S_j, $j \in J$, is the sub-plant that contains every plant model that has at least one event in Σ_{ess_j}, and nothing else. Let G_{ess_j} represent this plant and let $I_{ess_j} \subseteq I$ be its index set.

Formally, for $i \in I$

$i \in I_{ess_j} \leftrightarrow (\exists \sigma \in \Sigma_i)\, \sigma \in \Sigma_{ess_j}$

The first two theorems below identify sub-plants that can be used for verifying controllability of the entire plant. Unfortunately, G_{fund}, for a plant with high sub-model interaction, is usually too large thus a candidate plant must be selected. The limitation of Theorem 2 is that several possible sub-plants qualify as G_{cand}, many of which may fail to contain enough information to determine controllability. A good choice is G_{ess} since it is easily identifiable and is likely to succeed.

Theorem 1 *S_1 is controllable for Plant iff it is controllable for G_{fund_1}.*

Theorem 2 *If S_1 is controllable for G_{cand_1} then S_1 is controllable for Plant.*

The next theorem identifies sub-plants that can be used for testing non-blocking for individual controllers. This can be combined iteratively with Theorem 4 to verify that $((S_1 \wedge S_2 \wedge \ldots \wedge S_m) \,/\, Plant)$ is non-blocking.

Theorem 3 *If S_1, G_{fund_1} and G_{rem_1} are trim then S_1 is non-blocking for Plant iff S_1 is non-blocking for G_{fund_1}.*

Theorem 4 *If $(S_1 \,/\, G_{fund_1})$ and $(S_2 \,/\, G_{rem_1})$ are non-blocking and G_{fund_1} is decoupled from G_{fund_2} then $(S_1 \wedge S_2 \,/\, Plant)$ is non-blocking.*

We were able to successfully use Theorem 2 to verify controllability for the testbed's 29 modular supervisors. We employed candidate plants as small as (16 states, 72 transitions) and as large as (720 states, 5144 transitions). Since $L(S_j)$, $j \in J$, is a closed language, the intersection of these languages is controllable. Thus, the conjunction of the m modular supervisors is controllable.

As yet we have been unsuccessful in computing non-blocking for the testbed due to the high interaction among the sub-plants. The reduced plant was only slightly smaller than the testbed itself.

VII. Conclusion

In conclusion, we have designed and built a PLC based testbed for investigating the implementation of DES Supervisors. We have modeled the system and designed modular controllers for it. Finally, we developed several theorems that we used to verify that the supervisors are controllable. These theorems and the theorems on non-blocking can be implemented as a generic algorithm for application to large systems.

VIII. References

[1] B. A. Brandin. *Real-Time Supervisory Control of Automated Manufacturing Systems.* PhD thesis, Department of Electrical Engineering, University of Toronto, 1993. Also appears as Systems Control Group technical report # 9302, Department of Electrical Engineering, University of Toronto, February 1993.

[2] B.A. Brandin, W.M. Wonham, and B. Benhabib. Discrete-event systems supervisory control applied to the management of manufacturing workcells. In *7th International Conference on Computer Aided Manufacturing Engineering*, pages 527–536, Cookeville, Elsevier, 1991.

[3] G. Michel. *Programmable Logic Controllers- Architecture and Applications.* Wiley, 1990.

[4] P. Ramadge and W. Wonham. Supervisory control of a class of discrete-event processes. *SIAM J. Control Optim*, 25(1):206–230, 1987.

[5] J. Stenerson. *Fundamentals of Programmable Logic Controllers, Sensors, and Communications.* Prentice Hall, Inc., 1993.

[6] R. Strobel and A. Johnson. Pocket pagers in lots of one. *IEEE Spectrum*, pages 29–32, Sept 1993.

[7] W. Wonham and P. Ramadge. On the supremal controllable sublanguage of a given language. *SIAM J. Control Optim*, 25(3):637–659, 1987.