Petri Net Based Diagnosis for Construction Design

Calin Ciufudean, Adrian Graur, Constantin Filote, Daniel Popescu, and Camelia Petrescu

Abstract— **In this paper we propose a Petri net within a diagnosis system for construction design. We assimilate the construction process with an assembly process that composes parts and/or subassemblies into a unique product. We assume that the assembly supervisor (AS) system is distributed, and it solves several local AS attached to the nodes of the Petri net model of the assembly process. The research issues that we address in this work include the modeling of assembly process, determination of cost-effective assembly planes for efficient building, and real time adoption of a plan to a given product to be assembled. This work extends the known assembly Petri nets to a powerful framework enabling to derive the diagnosis of assembly process whose path may vary, and the objective function is maximized. The presented approach can be used to evaluate transient and steady-state performances of alternative design based on a construction example. Possible extensions of the work are also discussed.**

Index Terms— **Petri nets, diagnosis, assembly process, objective function.**

I. Introduction

In this paper we focus on the diagnosis of asynchronous systems. Typical examples are construction systems, such as shown in Fig.1. In Fig.1., the supervisor system is distributed, and it involves several local supervisors, attached to some nodes of the construction network. Each local supervisor has only a partial view of the overall system. The different local supervisors have their own local time, but they are not synchronized. Alarms are reported to the global supervisor, and this supervisor performs diagnosis. We notice that events may be correctly ordered by local supervisors, but communicating observations via network causes a loss of synchronization, and results in a non-deterministic supervisor. Model-based fault detection and diagnosis schemes have been investigated in great detail [1-6]. These references use linear, time-based dynamic models, and also Petri nets and discrete-event models. In any model based detection scheme, model prediction provides the basis for comparison with measured process behavior, and key issue is the appropriate selection and placement of local supervisors.

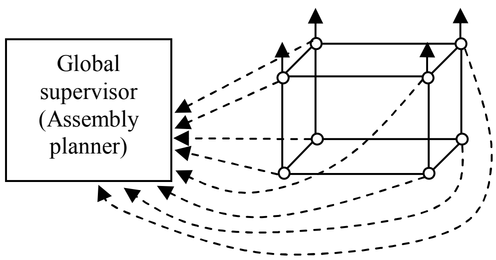

Fig.1. Supervision of a construction system

Fault analysis and diagnosis tools are commonly developed as a stand alone addition to the operation of a machine or section of a process plant. In real systems, where the local jobs are coordinated by local supervisors, such as shown in Fig.1, the problem of matching planning instances is NP-hard in general cases [6]. Adapting a process plan by changing its intermediate goals has implications for subsequent assembly functions, including production planning and scheduling. Alternative process plans can be exploited in real time to react to machine failures, in order to avoid having bottleneck machines, and to enable adaptive production planning of failure-prone construction systems. The work proposes an adaptive process-planning scheme that can manage process changes and adapt the process to specific assembly conditions. In order to solve this problem, the paper proposes a methodology for design and implementation of an adaptive assembly planner based on assembly Petri nets (APN). The advantages of using APN's include the following [7]:
- allowing the dynamic behavior to be visualized;
- representing both the assembly process and system resources in a single presentation for diagnosis and easy control implementation;
- allowing a linear programming formulation to find optimal assembly plans.

Calin Ciufudean, is with the Electrical Engineering Department "Stefan cel Mare" University of Suceava, 1 University Str., RO-720225, Suceava, Romania (phone: +40723255571; fax: +40-230-524801; e-mail: calin@eed.usv.ro,).
Adrian Graur, Constantin Filote are with the Electrical Engineering Department "Stefan cel Mare" University of Suceava, 1 University Str., RO-720225, Suceava, Romania (e-mail: adriang@eed.usv.ro, filote@eed.usv.ro).
Daniel Popescu is with Technical University of Constructions Bucharest 66 P.Protopopescu Bvd., Ro-73232, Bucharest, Romania (e-mail: dpopescu@instal.utcb.ro).
Camelia. Petrescu is with the ICPE Electrical Engineering CO., 313 Splaiul Unirii 3, RO-016348, Bucharest, Romania (e-mail: metrologie@icpe.ro).

The proposed planner is integrated, as shown in Fig.1, with an assembly system, and this principial scheme is represented in Fig.2. Input to the system consists of raw materials; output is the finite product (construction) and what remain to be dumped, secondary raw materials, s.a.

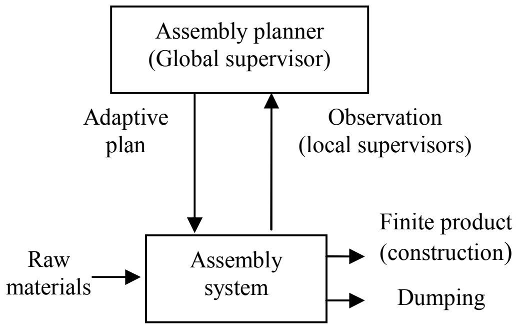

Fig.2 Adaptive assembly system

The assembly planner supplies a predictive plan for each product, respectively a plan that was generated based on previous data. During assembly execution, observations made by local supervisors are transferred directly to the global supervisor (assembly planner - see Fig.1). They are used to update predicted values of each component and respective assembly costs. Hence, the assembly system adapts the predictive process plan to the new data and generates an adapted plan that may lead to a new termination goal. The objectives of this paper are to present a method for developing an adaptive planner and to illustrate assembly process planning via a specific design with execution success rate and respective costs. Section 2 describes the assembly Petri net model and planning algorithm. Section 3 presents a design and implementation methodology for an assembly system.

II. MODELING THE CONSTRUCTION PROCESS: ASSEMBLY SUPERVISOR PETRI NETS

The paper extends the known assembly Petri nets (APN) into assembly supervisor Petri nets (ASPN). Thus ASPN's can accurately describe the construction topology, mating relations and precedence relations. In an ASPN a transition (assembly process) and a place (a product, or its subassembly) are associated with utility information (cost/benefit). Each transition is also associated with pre-firing and post-firing values. The pre-firing value is a decisional value which indicates the priority level for a transition to fire [8], respectively its associated assembly operation to perform. The post-firing value represents a probability that indicates the success rate of its assembly operation, which is updated based on the observations received from the local supervisors.

The ASPN can estimate the assembly performance, e.g. net profit, and also decides the best actions among various corrective actions, in order to maximize the profit. ASPN offers a good framework enabling to drive the optimal assembly process plan whose intermediary goals may vary and the objective function is maximized. An assembly supervisor Petri net is defined as: ASPN = $(P, T, W, Mo, f_1, f_2, v_d, v_p)$, where P and T are finite sets of places and transitions, respectively; $W \subseteq (PXT) \cup (TXP)$ is a set of directed arcs;
$M_0 : P \rightarrow N$ is the initial marking, where N is the set of nonnegative numbers.
The set of input (output) transitions of a place $p \in P$ is denoted by ^0p (p^0). The set of input (output) places of a transition $t \in T$ is denoted by ^0t (t^0). We also have:
$f_1 : P \rightarrow R^+$ is the resources utility function assigned to a place, where R is the set of nonnegative real numbers;
$f_2 : T \rightarrow R^+$ is the cost function assigned to a transition, where R^+ is the set of nonnegative real numbers;
$v_d : T \rightarrow N$ is a decisional value assigned to a transition. This value is assigned according to a planning algorithm. Value v_1 is used to decide firing priority of the transitions;
$v_p : T \rightarrow [0,1]$ is a probability value associated with a transition, that is updated according to the sensing result of the corresponding assembly operation. The value $v_p(t)$ represents the success rate of an assembly operation. The value $1 - v_p(t)$ represents the failure rate.

We notice that in an ASPN model, a place with multiple output transitions represents a subassembly with multiple ways to be assembled. These different assembly choices should determine a common set of assembled parts. Multiple output transitions from a place form a Logic-OR relation and multiple output places from a transition form a Logic-AND relation. Both place and transition utility functions are used to generate an optimal assembly plan. The decision and probability value of transitions are used to execute, respectively to adapt the construction plan. ASPN defined in such a way belongs to the class of free-choice Petri nets [9]. As shown in Fig.1 local supervisors modeled with ASPN's are coordinated by a global supervisor (GS). That means that the global ASPN model has the structure given in Fig.3, where p_{si}, i = 1,...,n are the partial subassemblies of the construction, and p_f represents the final product (construction).

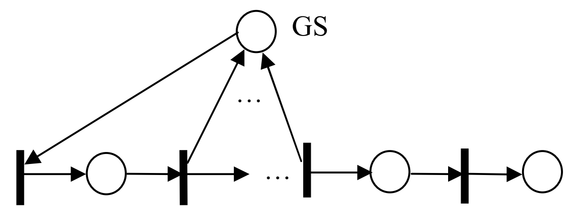

Fig.3 Global ASPN structure

Obviously, a construction planning schedule is to determine the best order of assembly operations, i.e. transition firings. In order to reject the uncertainty of assembly operations different-level priorities are assigned to different assembly alternatives for all the subassemblies [9].

In an ASPN, a place with multiple transitions implies assembly methods, each of them having its own way value (v_w). Introducing the v_d (decisional value) to each transition enables an easy determination of assembly order. For example let place p having k output transitions: t_1, t_2, ..., t_k and their values v_{wi}, i = 1,. ...,k.

The priority levels to transitions t_i are assigned accordingly to the following relation:

$$v_d(t_j) = i \quad (1)$$

If v_{wi} is the i^{th} smallest among v_{wi}, i = 1,...,k .

When an assembly operation fails (e.g. the ASPN diagnosis is revealing a bottleneck or a too expensive way of firing transitions), the assembly planner selects a transition with the next largest v_d value, and so on. The v_p values assigned to transitions are designed to adapt the ASPN for the maximum expected assembled value. Initially, all v_p are set accordingly to assembly planner (designer) experience. During execution, for each assembly operation, the number of successes are recorded and different v_{pi}, i = 1,...,k are re-adjusted with an exponential rate $e^{-N/N_s \cdot t_i}$ where N is the number of transitions fired for a subassembly, and N_s is the number of successes.
We notice that v_d is assigned based on both v_p and v_w for each transition.

III. DESIGN AND IMPLEMENTATION METHOD

Using the proposed ASPN and algorithms, we have the following design and implementation steps for a construction system:
a) Construct an ASPN given the product information;
b) Associate all the data with places and transitions in the ASPN;
c) Run the ASPN based on the construction resources

In order to understand these steps we implement an assembly (construction) system as a plausible example. We have raw material type A,B,C,D,E (e.g. parts A,B,C,D,E), and the possible way to assembly these parts to obtain the final subassembly F is depicted in Fig.4. We noted with TD the places that symbolize the dumping materials. In Fig.4, to each location we assign the utility function $f_1(P_1 \div P_{11})$ = (1,2,3,...,11). For example, the final product obviously has the greatest value, and the other values were assigned arbitrarily in this application. The assembly cost in this example is as follows: $f_2(t_1 \div t_9)$ = (1,2,...,9). Each transition has allocated in Fig.4 the respective cost.

Once we have the cost/benefits values of places and transitions we can find the optimal assembly plan (e.g. the optimal way in ASPN). The general job-shop scheduling problem has been shown to be NP-complete. Therefore, we resort to the heuristic search algorithm to solve this problem.

We use a heuristic best-first search procedure known as A* algorithm [10]. This algorithm is the following one:

Step 1. Place initial marking M_0 on the list VALID

Step 2. If VALID is empty terminate with failure.

Step 3. Choose a marking M from the list VALID with maximal cost f(M) and move it from the list VALID to the list NON-VALID.
Step 4. If M is the final marking, construct the searched way from the initial marking to the final marking, and terminate.

Step 5. Calculate $v_d(t_j)$ (see section 2 - relation (1)) and generate the successor markings for each enabled transition, and set the way from successors to M.

Step 6. For each successor marking M', do the following:
1. if marking M' is not already on list VALID or list NON-VALID, then put M' on list VALID;

2. if marking M' is already on list VALID and a way with a higher benefit is found, then direct its pointer along the current way;

3. if marking M' is already on list NON-VALID and a way with a higher benefit is found, then direct its pointer along the current way and move M' from list NON-VALID to the list VALID.

Step 7. Go to step 2

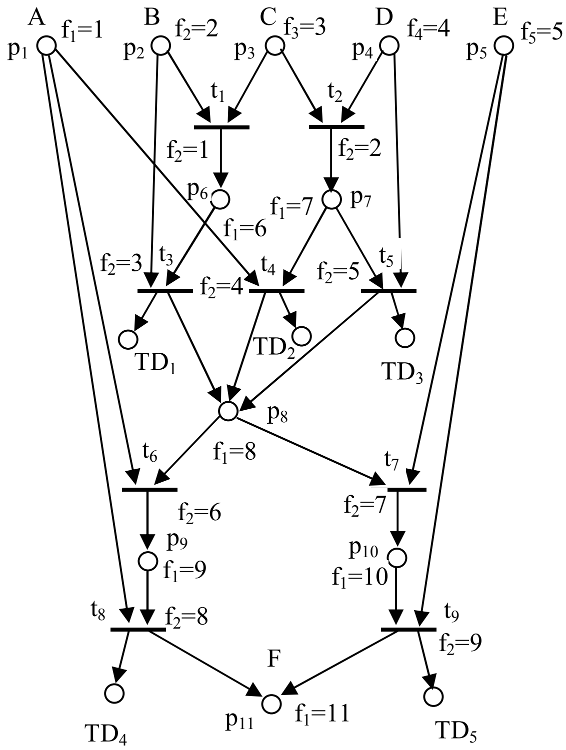

Fig.4. ASPN of a construction subassembly

For an assembly plan with n operations the complexity of this algorithm is O(bn), where b is the capacity of list VALID.

The algorithm complexity also depends upon the total number of nodes in ASPN as well as the total number of raw materials to perform the assembly operations concurrently.

For the example in Fig.4 the optimal assembly plan involves the transition (t_2, t_4, t_7, t_9). This way may be updated in accordance to the values v_{di}, and v_{pj}, where i = 1, ..., 11, and j=1, ..., 9, and to the marking of places TD_k, where k=1, ...,5. For simplicity, values v_{wj} were not updated.

IV. CONCLUSION

The approach presented in this paper is suited to distributed and asynchronous systems, such as construction ones, in which no global state and no global time is available, and therefore a partial order model is considered. This work proposes a methodology for design and implementation of adaptive assembly systems. In order to model the planning problem, ASPN is introduced, with two functions: one attached to places and the other to transitions. The first is a resource utility function, which represents the value of a subassembly, or a part to be used, and the second function represents the cost of performing a particular assembly operation.

To incorporate the uncertainty caused by different assembly conditions and the quality of resources, a probability value it is assigned to each transition. Probability values can be updated during process execution. Future research will focus on adapting colored Petri nets to the presented approach.

REFERENCES

[1] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, D. Tewneketzis, *Diagnosability of Discrete Event Systems,* IEEE Trans. on Autom. Contr., vol. 40, No.9, pp.1555-1575, 1995.

[2] A. Jalcin, Th. O. Boucher, *An Architecture for Flexible Manufacturing Cells with Alternate Machining and Alternate Sequencing*, IEEE Trans. Robot. Autom. vol. 15, no. 6, pp. 1126-1130, 1999

[3] Z. Liu, D. Towsley, *Stochastic Scheduling in Tree-networks*, Advances Appl. Probability, vol. 20, pp. 222-241, 1994

[4] M. Lahmar, H. Ergan, S. Benjafaar, *Resequencing and Feature Asignment on an Automated Assembly Line,* IEEE Trans. on Rob. and Autom., vol. 19, No.1, pp. 89-102, 2003.

[5] N. Viswandham, Y. Narahari, T.L.Johnson, *Deadlock Prevention and Deadlock Avoidance in Flexible Manufacturing Systems Using Petri Net Models,* IEEE Trans. Robotics Automat., vol. 6, no. 6, pp. 438-456, 1990.

[6] B. Nebel, J. Koehler, *Plan Reuse Versus Plan Generation - A Theoretical and Empirical Analysis*, Artif Intell., vol. 76, pp. 427-454, 1995.

[7] T. Suzuki, T. Kanehara, A. Inaba, S. Okuma, *On Algebraic and Graph Structural Properties of Assembly Petri Net,* in Proc. IEEE Int. Conf. Robot. Automat., Atlanta, G. A., pp. 507-514, 1993.

[8] E. Zussman, M. Ch. Zhou, *A Methodology for Modeling and Adaptive Planning of Disassembly Processes*, IEEE Trans. on Robot. and Autom., vol. 15, no. 1, pp. 190-193, 1999.

[9] E. Zussman, M. Ch. Zhou, *Design and Implementation of an Adaptive Process Planner for Disassembly Processes*, IEEE Trans. on Robot. and Autom., vol. 16, no. 2, pp. 171-179, 2000.

[10] R. V. Rogers, K. Preston White jr., *Algebraic Mathematical Programming and Network Models of the Deterministic Job-Shop Scheduling Problem*, IEEE Trans. Syst. Man. and Cybern., vol. 21, no. 3, pp. 350-361, 1991.