

# Reengineering Manufacturing Processes for Agility

*A. Kusiak and D. He*

*Intelligent Systems Laboratory*

*Department of Industrial Engineering*

*The University of Iowa*

*Iowa City, IA 52242 - 1527, USA*

*Tel. 1-319-335 5934, Fax: 1-319-335 5424*

*e-mail: ankusiak@icaen.uiowa.edu*

## **Abstract**

Agility can be achieved through the partnership of different companies sharing resources, information, and manufacturing capabilities. In a virtual enterprise, products are manufactured by a network of geographically distributed manufacturing partners. In order to achieve the necessary degree of agility, the distributed manufacturing processes have to be efficiently coordinated to timely react to the changing environment. In this paper, an approach to reengineering manufacturing processes is presented. The approach involves the Integrated Definition 3 (IDEF3) methodology, decomposing the manufacturing process based on similarity of resources, and scheduling the resultant Petri nets. Specifically, a sequential cluster identification algorithm is developed to decompose a manufacturing system represented as an IDEF3 model. A scheduling approach is presented to generate an aggregated schedule from sub Petri net models. The computational experience shows that the methodology developed in this paper reduces the computational time of solving the scheduling problem without significantly affecting its solution quality.

## **Keywords**

Reengineering, Agile Manufacturing, Scheduling, Virtual Enterprise

## **1 INTRODUCTION**

The markets for manufactured goods are characterized by uncertainty and rapidly changing demands. In order to succeed in such an environment, manufacturing companies must efficiently reduce the time needed to design, prototype, test, manufacture, and deliver to the market a high quality product. Agility can be achieved by partnering of numerous companies and sharing the resources, information, and manufacturing capabilities to form a virtual enterprise where the products are manufactured by a network of different and geographically

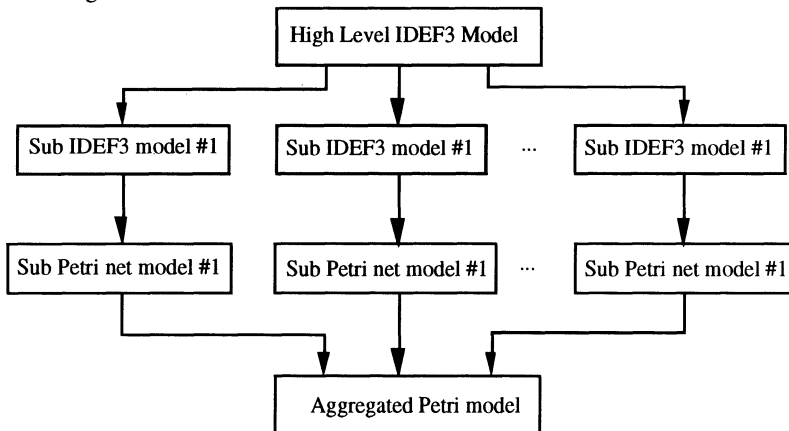
distributed partners. In order to achieve the necessary agility, this virtual enterprise has to be properly coordinated to timely react to the changing environment. This new virtual enterprise can be created by reengineering the existing partner companies.

Petri nets are an important tool for modeling and scheduling manufacturing processes (Peterson 1981, Narahari and Viswanadham 1985, Crockett et al. 1987, Kasturia et al. 1988, Al-Jaar and Desrochers 1990, Jafari 1990, Lee and DiCesare 1992). However, one of the disadvantages of the Petri net representation of a manufacturing process is the complexity issue involved in its model building. Therefore, a simple representation of manufacturing processes is needed to facilitate modeling the virtual manufacturing processes. The Integrated Definition 3 (IDEF3) methodology offers a graphic representation tool for the functions and their interconnections in a manufacturing system. Manufacturing operations (activities), processing times, resource requirements, process routes, and precedence relationships among operations can be easily represented with an IDEF3 model. For a large and complex process, the IDEF3 representation is easier to use than the corresponding Petri net model. The IDEF3 representation of a manufacturing process provides an effective man-machine interface. Despite its advantages, IDEF3 has a limited analytical capability required for control purposes. Combining IDEF3 and Petri net models enhances the performance of modeling and analyzing a virtual manufacturing process.

In this paper, a methodology for reengineering of manufacturing processes for agility is proposed. The remainder of the paper is organized as follows. In Section 2, a methodology for reengineering for agility is discussed. The results of computational experiments of the methodology are presented in Section 3. Section 4 concludes the paper.

## 2 METHODOLOGY

The basic idea of the methodology for reengineering for agility is shown in Figure 1. According to the methodology, the structure of a virtual manufacturing process is first described by a high level IDEF3 model. The structural information of the virtual manufacturing process includes manufacturing activities (operations), operation times, resources required by each manufacturing activity, and precedence relationships among the manufacturing activities.



**Figure 1** The framework of the methodology

Based on the information included in the IDEF3 model, the activities are then clustered into groups. The activities in a group can be further represented by a sub IDEF3 model that includes precedence constraints, operation times, and resource requirements. Based on the sub IDEF3 model for each group of activities, a sub Petri net model is generated. These sub Petri net models are then aggregated into an aggregated Petri net model by connecting the sub Petri net models. The aggregated Petri net model is constructed based on the data from the high level IDEF3 model. Therefore, instead of building a complex Petri net model, smaller and simpler Petri net models are built and analyzed. The aggregated Petri model provides a basis for aggregate Petri net scheduling.

## **2.1 Decomposition of IDEF3 Models**

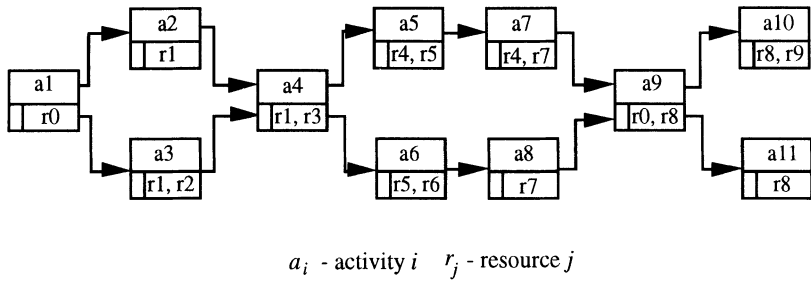
### **2.1.1 IDEF3 Methodology**

The IDEF3 methodology is a powerful tool for modeling a wide variety of systems which use hardware, software, and people to perform activities (U.S. Air Force 1981 and Mayer et al. 1992). IDEF0 provides a structured representation of functions, information, and objects that are interrelated. IDEF3 was created specifically to reflect the sequence of activities, e.g., performed in a manufacturing system. The IDEF3 process flow description is made up of units of behaviors (UOBs), links, and junction boxes. A UOB represents an activity occurring in the process. For example, loading or unloading parts and transporting parts are activities that may be represented as UOBs in a process model. For complete description of IDEF0 and IDEF3 see U.S. Air Force (1981) and Mayer et al. (1992). In this paper, the IDEF3 methodology with a UOB from the IDEF0 methodology is used and is referred to throughout the paper as the IDEF3 methodology (see Kusiak et al. 1994).

### **2.1.2 Decomposition of IDEF3 Models**

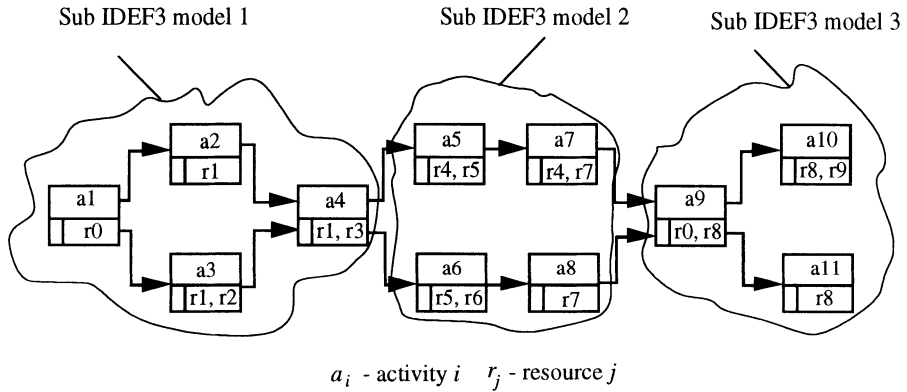
An IDEF3 model of a manufacturing process is decomposed into groups of activities in order to reduce the degree of resource sharing among these groups. The reason for considering the resource sharing criterion in the decomposition is that resource sharing is an issue in Petri net modeling (Zhou and DiCesare 1990, 1991). Sharing resources in a manufacturing system, e.g., operators, machine tools, robots, tools, fixtures, and so on often causes conflicts. The existence of shared resources is a major reason for Petri nets to contain deadlocks. One condition for correct operation of a manufacturing system is that its Petri net model be deadlock-free. Resource sharing may also increase the scheduling complexity of a Petri net because the transition firing conflicts caused by shared resources need to be resolved. The research on synthesizing Petri nets that are deadlock-free has received a significant attention in the literature (e.g., Datta and Ghosh 1984, Krogh and Beck 1986, Ahuja and Valavanis 1988, Koh and DiCesare 1990). To consider the issue of resource sharing in the synthesis of Petri nets, methods for synthesis of Petri nets for manufacturing systems with shared resources were proposed (Zhou and DiCesare 1991, 1992). One problem associated with Petri net synthesis is that an expertise is required in Petri net modeling and its effectiveness strongly depends on the complexity of a Petri net model. Instead of dealing directly with Petri net models, the methodology developed in this paper suggests that the system be decomposed with a consideration of resource sharing at the IDEF3 representation level. By decomposing the system based on resource sharing, difficulties in Petri net modeling and analysis, e.g., scheduling can be avoided. If two groups of activities do not share resources, then the corresponding sub Petri net models are not connected by the common resource places and their transition firing sequences can be determined independently. It is desired that a manufacturing system be decomposed so that the number of resources shared by different groups of activities is minimized. To illustrate the decomposition of an IDEF3 model based on the resource sharing, consider a virtual manufacturing process represented by the IDEF3

model in Figure 2.



**Figure 2** The IDEF3 model of a virtual manufacturing process

The sub IDEF3 models of the IDEF3 model in Figure 2 are shown in Figure 3. The only resource shared by two groups of activities is  $r_0$ .



**Figure 3** The decomposed IDEF3 model

The benefits of decomposing a manufacturing system represented with IDEF3 are twofold. First, as the decomposition is not performed on a Petri net, the difficulty associated with the complexity of Petri net modeling can be avoided and the decomposition task can be easily automated. Second, the decomposed IDEF3 models are easy to interpret by the shop floor operators.

Before the decomposition methodology is presented, two assumptions are introduced. The first assumption is that the IDEF3 model is acyclic, i.e., no cycles exist in the model. This type of IDEF3 model can be obtained through reengineering of the process model (Kusiak et al. 1994). The second assumption is that the precedence relationship among activities represented with an IDEF3 model is not violated when the IDEF3 model is decomposed.

In order to perform the decomposition, an activity-resource incidence matrix is obtained based on the information included in the IDEF3 model. In an activity-resource

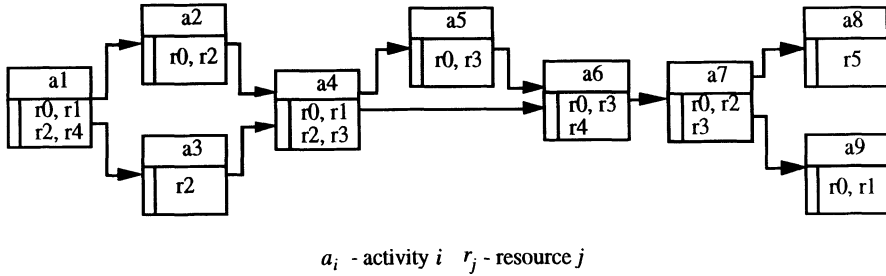
incidence matrix, the rows represent activities and the columns represent resources. The element of the matrix is "1" if a resource is required by the corresponding activity. Otherwise, the element is "0". For the sake of simplicity, elements of 0 are not shown in the activity-resource incidence matrices throughout the paper. For example, consider the IDEF3 model of a virtual manufacturing process in Figure 2. The corresponding activity-resource incidence matrix is shown in Figure 4.

	r0	r1	r2	r3	r4	r5	r6	r7	r8	r9
a1	1									
a2		1								
a3		1	1							
a4		1		1						
a5					1	1				
a6						1	1			
a7					1			1		
a8								1		
a9	1								1	
a10									1	1
a11									1	

**Figure 4** The activity-resource incidence matrix

To decompose an IDEF3 model using the resource sharing criterion and precedence constraints, a sequential cluster identification algorithm (SCI) is developed. The SCI algorithm is different from the cluster identification (CI) algorithm (Kusiak 1990) in two heuristic rules. The first rule is the precedence constraint preservation (PCP) rule. This rule is to preserve the precedence relationships among activities when the IDEF3 model is decomposed. The symbol  $\rightarrow$  is used to represent the precedence relationship between any two activities:  $a \rightarrow b$  means that activity  $a$  precedes activity  $b$  in a virtual manufacturing process, i.e., activity  $b$  will not start until activity  $a$  is completed. The PCP rule is stated as follows: if the precedence relationship  $a \rightarrow b \rightarrow c$  exists and activities  $a$  and  $b$  are not included in the same activity group, then activities  $a$  and  $c$  are to be included in different groups.

The second rule is the critical resource sharing (CRS) rule. The CRS rule is to enhance the decomposition of the IDEF3 model. The decomposition of an activity-resource incidence matrix may not lead to mutually exclusive clusters, i.e., all activities are grouped in one group. The latter is due to some resources being shared by most activities of a virtual manufacturing process, e.g., pallets and fixtures. To avoid this problem, the critical resources are removed from the activity-resource incidence matrix before the matrix is decomposed. The CRS rule is implemented with two procedures. The first procedure is the sequential sharing (SS) procedure. If a resource is shared by a predefined number of consecutive activities, then this resource is removed from the activity-resource incidence matrix. This predefined number of consecutive activities can be determined based on the experience. The second procedure is the frequent sharing (FS) procedure. The FS procedure is to compute the usage frequency of a resource and remove the resource from the activity-resource matrix if its usage frequency exceeds a predefined value. This predefined value is determined based on the knowledge of the manufacturing process. Consider the IDEF3 model of a virtual manufacturing process in Figure 5 where resource  $r_0$  is shared by seven consecutive activities ( $a_1, a_2, a_4, a_5, a_6, a_7, a_9$ ) and has a usage frequency of  $7/9 = 0.78$ . If the predefined number of consecutive activities is 5 and the usage frequency is 0.7, then resource  $r_0$  is removed from the activity-resource incidence matrix before its decomposition.



**Figure 5** The IDEF3 model of a virtual manufacturing process

Let  $K'$  be the set of activities and  $K$  the maximum sub set of activities in  $K'$  that satisfy the PCP rule. The SCI algorithm is presented next.

**The Sequential Cluster Identification (SCI) Algorithm**

**Step 0.** Set iteration number  $k = 1$ . Set  $K' = K = 0$ .

**Step 1.** Select the first row of incidence matrix  $A^{(k)}$ , place it in  $K'$ , and draw a horizontal line  $h_i$

through it ( $A^{(k)}$  is read: matrix  $A$  at iteration  $k$ ).

**Step 2.** For each entry 1 crossed by the horizontal line  $h_i$  draw a vertical line  $v_j$ .

**Step 3.** Select rows corresponding to the entry 1 crossed once by a vertical line. Place them in set  $K'$ . Select the maximum subset  $K$  satisfying PCP rule from set  $K'$ .  
Set  $K' = K$ .

For each row in set  $K$  which has not yet been crossed by a horizontal line, draw a horizontal line  $h_i$ .

**Step 4.** For each row in set  $K$ , do:

(1) For each entry 1 crossed once by a vertical line, draw a horizontal line  $h_k$ .

(2) For entry 1 crossed once by the horizontal line  $h_k$  draw a vertical line  $v_k$ .

**Step 5.** Repeat Step 3 and 4 until each row in  $K$  has no crossed-once entries 1. All crossed-twice entries 1 in  $A^{(k)}$  form activity group  $A_k$  and resource group  $R_k$ .

**Step 6.** Transform the incidence matrix  $A^{(k)}$  into  $A^{(k+1)}$  by removing all the rows and columns

having no crossed-once entries 1.

**Step 7.** If matrix  $A^{(k+1)} = 0$  (where 0 denotes a matrix with all elements equal to zero), stop; otherwise set  $k = k+1$  and go to Step 1.

**Example**

Consider the manufacturing process represented by the IDEF3 model in Figure 5. The corresponding activity-resource incidence matrix is generated.

$$A = \begin{matrix} & r0 & r1 & r2 & r3 & r4 & r5 \\ \begin{matrix} a1 \\ a2 \\ a3 \\ a4 \\ a5 \\ a6 \\ a7 \\ a8 \\ a9 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & & 1 \\ 1 & & 1 & & \\ & 1 & & & \\ 1 & 1 & & & \\ 1 & & & 1 & \\ 1 & & & 1 & 1 \\ 1 & & 1 & & \\ & & & & 1 \\ 1 & 1 & & & \end{bmatrix} \end{matrix}$$

As discussed before, resource  $r_0$  is removed from the above matrix according to the FS procedure. The result is shown in matrix (1).

$$A^{(1)} = \begin{matrix} & r1 & r2 & r3 & r4 & r5 \\ \begin{matrix} a1 \\ a2 \\ a3 \\ a4 \\ a5 \\ a6 \\ a7 \\ a8 \\ a9 \end{matrix} & \begin{bmatrix} 1 & 1 & & 1 \\ & 1 & & \\ & 1 & & \\ 1 & 1 & 1 & \\ & 1 & & 1 \\ & & 1 & 1 \\ & 1 & 1 & \\ & & & 1 \\ 1 & & & \end{bmatrix} \end{matrix} \quad (1)$$

**Step 0.** Set iteration  $k = 1$ . Set  $K' = K = 0$ .

**Step 1.** Row 1 of matrix (1) is selected and placed in set  $K' = \{1\}$ . A horizontal line  $h_1$  is drawn. The result of Step 1 is shown in matrix (2).

**Step 2.** Three vertical lines  $v_1$ ,  $v_2$ , and  $v_4$  are drawn. The result of Step 2 is shown in matrix (2).

$$A^{(1)} = \begin{matrix} & r1 & r2 & r3 & r4 & r5 \\ \begin{matrix} a1 \\ a2 \\ a3 \\ a4 \\ a5 \\ a6 \\ a7 \\ a8 \\ a9 \end{matrix} & \begin{bmatrix} 1 & 1 & & 1 & \\ \vdots & 1 & & & \\ \vdots & 1 & & & \\ 1 & 1 & 1 & & \\ \vdots & \vdots & 1 & & \\ \vdots & \vdots & 1 & 1 & \\ \vdots & 1 & 1 & & \\ \vdots & \vdots & & & 1 \\ 1 & \vdots & & & \end{bmatrix} \end{matrix} \quad \begin{matrix} h_1 \\ \\ \\ v_1 & v_2 & & v_4 \end{matrix} \quad (2)$$

**Step 3.** Rows 2, 3, 4, 6, 7, and 9 are selected and placed in  $K' = \{1, 2, 3, 4, 6, 7, 9\}$ . The maximum subset  $K = \{1, 2, 3, 4\}$  is obtained by applying the PCP rule. Set  $K' = K = \{1, 2, 3, 4\}$ . For each row in set  $K$ , a horizontal line is drawn, as shown in matrix (3).

$$A^{(1)} = \begin{array}{c} \begin{array}{ccccc} & r1 & r2 & r3 & r4 & r5 \\ \begin{array}{c} a1 \\ a2 \\ a3 \\ a4 \\ a5 \\ a6 \\ a7 \\ a8 \\ a9 \end{array} & \left[ \begin{array}{ccccc} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{array} \right] & \begin{array}{c} h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \end{array} \quad (3)$$

$\begin{array}{ccccc} & v_1 & v_2 & & v_4 \end{array}$

**Step 4.** Since row 4 has an entry 1 crossed once at column 3, a vertical line  $v_3$  is drawn. The result is shown in matrix (4).

$$A^{(1)} = \begin{array}{c} \begin{array}{ccccc} & r1 & r2 & r3 & r4 & r5 \\ \begin{array}{c} a1 \\ a2 \\ a3 \\ a4 \\ a5 \\ a6 \\ a7 \\ a8 \\ a9 \end{array} & \left[ \begin{array}{ccccc} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{array} \right] & \begin{array}{c} h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \end{array} \quad (4)$$

$\begin{array}{ccccc} & v_1 & v_2 & v_3 & v_4 \end{array}$

**Step 5.** Since the entry 1 at rows 5, 6, and 7 is crossed once by vertical line  $v_3$ , rows 5, 6, and 7 are placed in  $K' = \{1, 2, 3, 4, 5, 6, 7\}$ . The maximum subset  $K = \{1, 2, 3, 4, 5, 6, 7\}$  is obtained by applying the PCP rule. Set  $K' = K = \{1, 2, 3, 4, 5, 6, 7\}$ . Draw horizontal lines  $h_5, h_6$ , and  $h_7$  as shown in matrix (5).

$$A^{(1)} = \begin{array}{c} \begin{array}{ccccc} & r1 & r2 & r3 & r4 & r5 \\ \begin{array}{c} a1 \\ a2 \\ a3 \\ a4 \\ a5 \\ a6 \\ a7 \\ a8 \\ a9 \end{array} & \left[ \begin{array}{ccccc} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{array} \right] & \begin{array}{c} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \end{array} \end{array} \quad (5)$$

$\begin{array}{ccccc} & v_1 & v_2 & v_3 & v_4 \end{array}$

Since all rows in  $K$  have no crossed-once entries 1, they form an activity group  $A_1$ . The obtained activity and resource groups are:

Activity group:  $A_1 = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$ .

Resource group:  $R_1 = \{r_1, r_2, r_3, r_4\}$ .

**Step 6.** After removing all the rows and columns that have no crossed-once entries 1, matrix (5) is transformed into matrix (6).



$$A^{(2)} = \begin{matrix} & r1 & r4 \\ \begin{matrix} a8 \\ a9 \end{matrix} & \begin{bmatrix} 1 & 1 \end{bmatrix} \end{matrix} \quad (6)$$

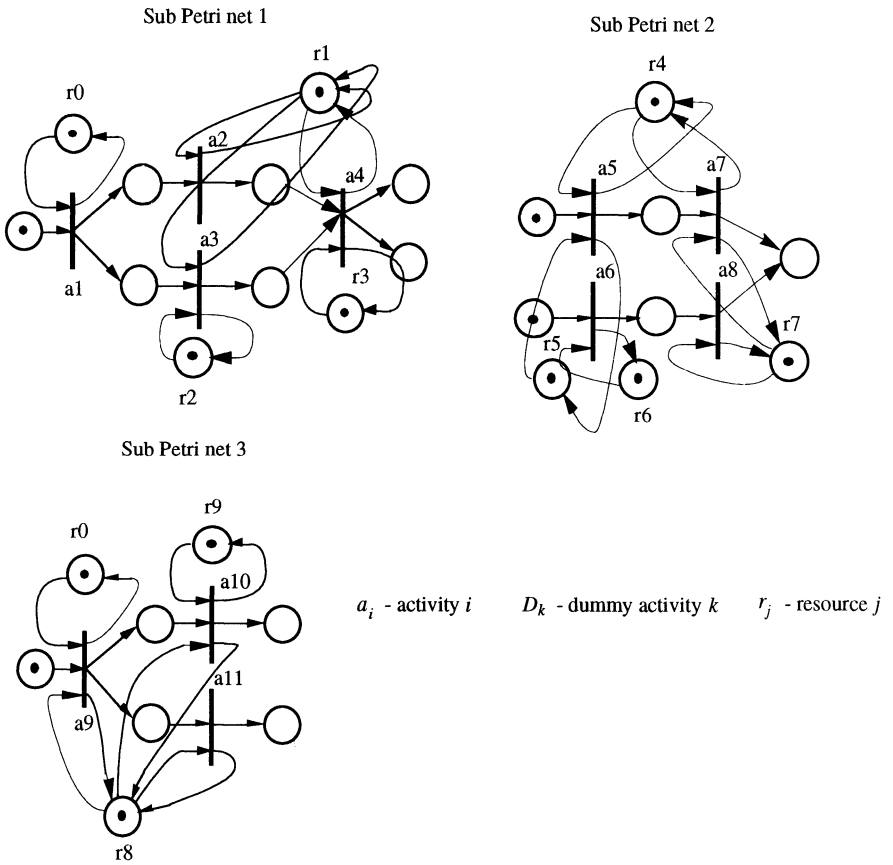
After two more iterations, two activity and resource groups are obtained:

$A_2 = \{a_8\}$ ,  $R_2 = \{r_4\}$ ;

$A_3 = \{a_9\}$ ,  $R_3 = \{r_1\}$ .

Note that resource  $r_1$  is the only resource shared by two different groups  $A_1$  and  $A_3$ .

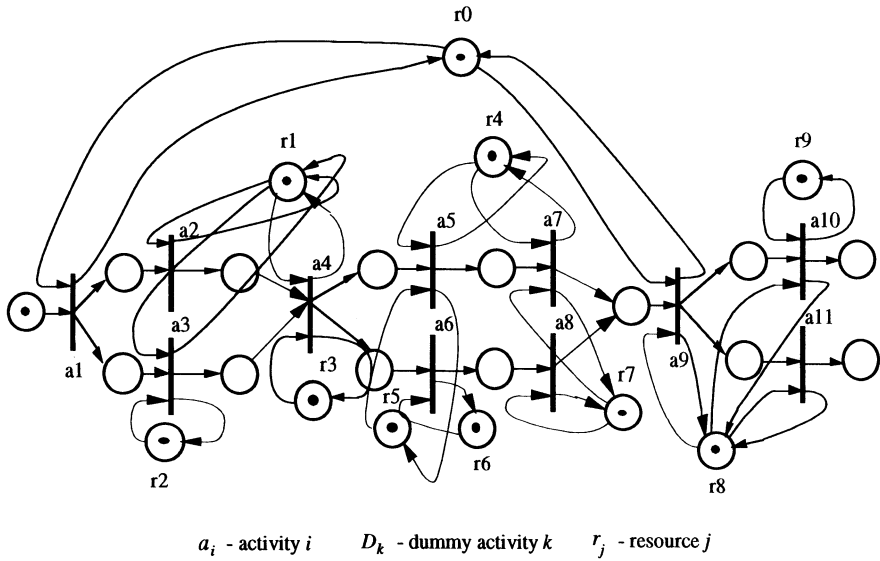
### 2.2.2 The Aggregation of Sub Petri Net Models



**Figure 6** The sub Petri net models corresponding to the sub IDEF3 models in Figure 3

Each sub IDEF3 model can be transformed into a sub Petri net model. The mechanism for transforming a task graph into a Petri net model is described in Strege and Loydl (1995). Since an IDEF3 model contains the production data as a task graph, it can be also converted into a Petri net model using the principles described in Strege and Loydl (1995). After the sub Petri net models have been generated, they are transformed into an aggregated Petri net model. The aggregation is performed by connecting the sub Petri nets based on the precedence and resource sharing information provided by the IDEF3 model. The precedence relationship among activities appear in the aggregated Petri net model in the form of arcs connecting transitions and places. If the activities corresponding to the transitions in different sub Petri nets share the same resources, then these transitions in the aggregated Petri net are connected by the same resource places. For example, consider the sub Petri nets in Figure 6 generated for the sub IDEF3 models in Figure 4.

The corresponding aggregated Petri net for the high level IDEF3 model is illustrated in Figure 7.



**Figure 7** The aggregated Petri net model

### 2.2.3 Aggregate Petri Net Scheduling

Scheduling a Petri net determines the sequence of the transition firing for each resource so that the makespan of the underlying manufacturing process is minimized. Once a Petri net model of the manufacturing system is obtained, a production schedule can be determined by computing the firing sequence of the Petri net model. Theoretically, an optimal schedule can be obtained by generating the reachability graph and finding a path from the initial marking to the final marking. However, even for a simple Petri net, an entire reachability graph may be too large. Instead of generating the entire reachability graph, heuristic algorithms to generate only the necessary portion of the reachability graph are used to find an acceptable path. In this paper, the scheduling system described in Strege and Pham (1994) is used to determine the firing sequence  $s_{ri}$  and  $s_{pj}$  for each resource place  $r_i$  and job place  $p_j$ , respectively. The scheduling system used for determining the partial firing sequence in each sub Petri net is a

combination of heuristic scheduling rules and a deadlock recovery procedure. The heuristic scheduling rules include the first-in-first-out (FIFO), last-in-last-out (LIFO), and shortest processing time (SPT) rules. Whenever a deadlock is encountered in scheduling, a deadlock recovery procedure is initiated. It combines a deterministic backtracking and a stochastic (Monte Carlo) forward tracking steps. It attempts to reach a deadlock-free state from a deadlock state in the minimum number of changes. A detailed description of the scheduling system is presented in Strege and Pham (1994).

Analogous to the decomposition of IDEF3 models, a Petri net scheduling approach is developed. Instead of generating a schedule for the entire Petri net, a partial firing sequence of each sub Petri net is determined. Then these partial firing sequences are transformed into an aggregated schedule. The Petri net scheduling approach works as follows. First, the transition firing sequences of resources in each sub Petri net are determined. When determining the transition firing sequences in each sub Petri net, the resources shared are scheduled in each sub Petri net as if they were exclusively used in each sub Petri net. Then the transition firing sequences of the shared resources are determined. For resources that are not shared by other sub Petri nets, their firing sequences can be directly used in the aggregated schedule. The transition firing sequences of shared resources have to be determined when partial firing sequences of sub Petri nets are aggregated. In determining the firing sequence of shared resources, simulation is performed. The purpose of simulation is to determine a feasible firing sequence of each shared resource. To determine the feasible firing sequences of the shared resources, firing conflicts have to be resolved. Rules for solving these conflicts have been developed (Strege and Pham 1994). The scheduling rules used include the first-in-first-out (FIFO), last-in-last-out (LIFO), and shortest processing time (SPT) rules.

The decomposition of a virtual manufacturing process and the construction of sub Petri nets reduce the complexity of Petri net scheduling. When the manufacturing system is decomposed, small and simple sub Petri nets are built. Note that the sub Petri nets do not represent the actual subsystems, but they are useful in determining a "good" schedule for the system. Finding a firing sequence for a small and simple Petri net is easy. Since the sub Petri nets can be scheduled simultaneously and the schedules for these sub Petri nets can be aggregated, the time required to find a firing sequence for the aggregated net can be reduced dramatically.

### 3 COMPUTATIONAL EXPERIENCE

In order to evaluate the performance of the methodology developed in this paper, a computational study has been performed. In the study, numerous computational experiments were performed. The structures of the manufacturing processes were randomly generated. The IDEF3 models representing the manufacturing processes were decomposed by the SCI algorithm. The sub Petri nets were then generated and transformed into aggregate Petri net models. The number of resources shared in a virtual manufacturing process was set proportional to the number of activities in the IDEF3 models (transitions in the Petri nets). The number of activities ranged from 71 to 122. The number of tokens used in simulating the Petri net schedules ranged from 10 to 120. Each token represented a job to be performed in the system. As in the computational study for the flexible disassembly cell, two types of Petri net schedules were generated and the relative difference between the makespans of the two schedules was computed. The computational times for generating a schedule for the entire Petri net and the schedules for sub Petri nets of the corresponding Petri net were recorded. The results of computational study are provided in Table 1.

For each experiment in Table 1, the makespans of the corresponding schedules are almost identical. However, computing sub Petri net schedules takes much less time than computing a schedule for the entire Petri net.

The results of computational study shows that the methodology developed in this paper reduces the computational time of generating a Petri net schedule without a significant change of its quality.

4 CONCLUSIONS

Agility can be achieved by partnering numerous companies and sharing resources, information, and manufacturing capabilities. In a virtual enterprise, the products are manufactured by a network of geographically distributed partners companies. In order to achieve the necessary agility, distributed manufacturing process has to be efficiently coordinated to timely react to the changing environment. In this paper, a reengineering approach was discussed. The approach includes building a structured representation of a virtual manufacturing process with the Integrated Definition 3 (IDEF3) methodology, decomposing the manufacturing process based on the similarity of resources, and scheduling the Petri nets. Specifically, a sequential cluster identification algorithm was developed to decompose a manufacturing system represented as an IDEF3 model. A scheduling approach was presented to generate an aggregated schedule from sub Petri net models. The computational experience shows that the methodology developed in this paper reduces the computational time complexity of the scheduling problem without significantly affecting the solution quality.

**Table 1** The results of computational experiments

Experiment No.	CPU time (second)		Relative Difference in Makespan $\frac{M_2^c - M_1^d}{M_2} 100\%$
	Schedule 1 <sup>a</sup>	Schedule 2 <sup>b</sup>	
1	14.4	1.1	1.8
2	35.5	1.6	0.6
3	37	1.7	0.9

- a: the schedule generated for the entire Petri net
- b: the schedule generated by aggregating the partial schedule of sub Petri nets
- c: the makespan of Schedule 2
- d: the makespan of Schedule 1

5 REFERENCES

Ahuja, J. S. and Valavanis, K. P. (1988). "A Hierarchical Modeling Methodology for Flexible Manufacturing Systems Using Extended Petri Nets", *Proceedings of International Conference of Computer Integrated Manufacturing*, Troy, NY, pp. 350 - 356.

- Al-Jaar, R. Y. and Desrochers, A. A. (1990). "Petri Nets in Automation and Manufacturing", in G. H. Saridis (Ed), *Advances in Automation and Robotics*, JAI Press, CT, Vol. 2.
- Crockett, D., Desrochers, F., DiCesare, and Ward, T. (1987). "Implementation of A Petri Net Controller for A Machining Workstation", *Proceeding of IEEE Conference on Robotics and Automation*, Raleigh, NC, pp. 1861 - 1867.
- Datta, A. and Ghosh, S. (1984). "Synthesis of A Class of Deadlock-free Petri Nets", *Journal of Association of Computing Mathematics*, Vol. 31, No. 3, pp. 486 - 506.
- Datta, A. and Ghosh, S. (1986). "Modular Synthesis of Deadlock-free Control Structures", in Goos and Hartmanis (Eds), *Foundations of Software Technology and Theoretical Computer Science*, Vol. 241, Springer-Verlag, New York, pp. 288 - 318.
- Jafari, M. A. (1990). "Petri Net Based Shop Floor Controller and Recovery Analysis", *Proceedings of the IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, pp. 532 - 537.
- Kasturia, E., DiCesare, F., and Desrochers, A. (1988). "Real-time Control of Multilevel Manufacturing Systems Using Colored Petri Nets", *Proceedings of IEEE Conference on Robotics and Automation*, Philadelphia, PA, pp. 1114 - 1119.
- Koh, I. and DiCesare, F. (1990). "Transformation Methods for Generalized Petri Nets and Their Applications in Flexible Manufacturing Systems", *Proceedings of International Conference on Computer Integrated Manufacturing*, Tony, NY, pp. 363 - 371.
- Krogh, B. H., and Beck, C. L. (1986). "Synthesis of Place/Transition Nets in Simulation and Control of Manufacturing Systems", *Proceedings of 4th IFAC/IFORS Symposium: Large Scale Systems* (Zurich).
- Kusiak, A. (1990). *Intelligent Manufacturing Systems*, Prentice Hall: Englewood Cliffs, N.J.
- Kusiak, A., Larson, T. N., and Wang, J. (1994). "Reengineering of Design and Manufacturing Processes", *Computers and Industrial Engineering*, Vol. 26, No. 3, pp. 521-536.
- Lee, D. Y. and DiCesare, F. (1992). "FMS Scheduling Using Petri Nets and Heuristic Search", *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France.
- Mayer, R. J., Cullinane, T. P., deWitte, P. S., Knappenberger, W. B., Perakath, B., and Wells, M. S. (1992). *Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report*, Armstrong Laboratory, Wright-Patterson AFB, Ohio 45433, AL-TR-1992-0057.
- Narahari, Y. and Viswanadham, N. (1985). "A Petri Net Approach to the Modeling and Analysis of Flexible Manufacturing Systems", *Annals of Operations Research*, Vol. 3, pp. 449 - 472.
- Peterson, J. L. (1981). *Petri Net Theory and the Modeling of Systems*, Prentice Hall, Englewood Cliffs, New Jersey.
- Strege, B. and Loydl, H. (1995). "Automatic Transformation of Simple User Commands to Hierarchical Petri Nets", *Proceedings of Systems, Man, and Cybernetics*, Vancouver, Canada.
- Strege, B. and Pham Q. B. (1994). "Autonomous Treatment of Special System States in Flexible Manufacturing Cells by Integration of Knowledge Based Reaction Planning, Scheduling and Petri Net Modifications", *Proceedings of The 10th ISPE/FAC International Conference on CAD/CAM, Robotics and Factories of the Future*, Ottawa, Canada, pp. 762-767.
- U.S. Air Force (1981). *Integrated Computer Aided Manufacturing (ICAM) Architecture Part II, Volume IV-Functional Modeling Manual (IDEF0)*, Air Force Materials Laboratory, Wright-Patterson AFB, Ohio 45433, AFWAL-tr-81-81-4023.

- Zhou, M. C. and DiCesare, F. (1990). "A Petri Net Design Method for Automated Manufacturing Systems with Shared Resources", *Proceedings of the IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, pp. 526 - 531.
- Zhou, M. C. and DiCesare, F. (1991). "Parallel and Sequential Mutual Exclusions for Petri Net Modeling for Manufacturing Systems with Shared Resources", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 4, pp. 515 - 527.
- Zhou, M. C. and DiCesare, F. (1992). "A Hybrid Methodology for Synthesis of Petri Net Models for Manufacturing Systems", *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 3, pp. 350 -361.

Additional references are available at World Wide Web:  
<http://www.icaen.uiowa.edu/~ankusiak>