DESIGN AND SIMULATION OF A HOLONIC ASSEMBLY SYSTEM

Peter BOBER, Michal GIRMAN
Laboratory of Industrial Engineering, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic, Phone: +55/602 2281, E-mail: peter.bober@tuke.sk, michal.girman@tuke.sk

SUMMARY

This paper defines an overall structure and behaviour of components of a holonic assembly system. The required properties of the holonic assembly system are autonomy and possibility of automatic reconfiguration in case of failures and relocations of individual elements - holons. The assembly system consists of input buffer, transportation, and universal assembly stations. Rule based distributed scheduling is used for assembly task planning. Desired properties are achieved by mutual interaction of holons. Designed structure and scheduling rules are verified by simulation. The proposed scheduling rule gives good output rate already for 10 tasks available at a time as compared to quasi optimal output rate when all 200 work orders were scheduled at once.

Keywords: *holonic manufacturing, distributed scheduling, holon, simulation*

1. INTRODUCTION

Trend in the development of process management and control is dictated by the requirements of flexibility and agility. Holonic manufacturing system concepts are used to introduce this flexibility for manufacturing of high variety low-volume products. Logically and physically independent intelligent entities (called holons) act both autonomously and cooperatively to manage the shop-floor in a distributed fashion [1].

The properties of the whole are determined by properties of individual components and their interaction. A basic task in building of holonic systems is to design individual holon properties and mutual interaction in order to create the required behaviour of the whole.

According Fletcher [2] a holon comprises:

- A low-level process and machine control system that handles hard realtime decisions concerning the manufacturing hardware. This control system would be typically built using the emerging IEC 61499 standard for open distributed manufacturing systems.
- A high-level control system responsible for planning, scheduling, configuration and other non-realtime decision making actions that demand knowledge. This is usually modelled by intelligent software agents (see also Ulieru [3]).

The following paragraphs deal with the design and modelling of properties and structure of the holonic assembly system. We abstract from technical low-level process and machine control as well as software implementation.

2. HOLONIC ASSEMBLY SYSTEM

Assembly shop-floor is used as a typical example where holonic assembly system can be introduced. It consists of storage area (input buffer), transportation, and universal assembly stations (*UAS*). This shop-floor should be able to assembly arrived parts automatically. Parts for one final product are stored in one container. In our model, each *UAS* is able to assembly all final products but the assembly time varies among *UAS*. The number of active *UASs* can vary because of breakdowns, maintenance work, or redistribution of assembly capacity among several shop-floors.

3. DESIGN OF HOLONIC ASSEMBLY SYSTEM

3.1 Structural Design

The required behaviour of shop-floor is a result of the structure and mutual interaction of system's elements. The system's structure is built upon functional grouping. The shop-floor consists of three types of holons:

- Coordinator holon
- Transportation holon
- Universal Assembly Station (UAS)

Coordinator holon and *Transportation* holon are persistent. The availability and number of *UASs* is changing according external circumstances (break down, maintenance work, reallocation of assembly capacity, etc.). Fig. 1 shows the structure, material, and information flow among holons.

3.2 Coordinator

The Coordinator holon is the central point for information exchange. It registers the presence of other elements of the system namely *UASs* and ensures assembly task planning. *Coordinator* is not responsible for central planning; however, it coordinates the scheduling algorithm of individual *UASs* as explained in 3.5.

Coordinator tasks:

- register presence and state of *UASs*,
- receive and store assembly tasks (containers with parts),

- coordinate task scheduling,
- move selected container to *Transportation*,
- update container, *UASs*, and request lists.

Events:
- change of state of *UAS*,
- new task (container) arrival,
- requesting tasks from *UAS*.

Local information resources:
- list of *UASs*,
- list of tasks (containers),
- list of requests.

Dynamic behaviour of *Coordinator* holon is modelled by the state diagram on fig. 2.

3.3 Transportation

Distribution of containers (assembly tasks) is provided by *Transportation* holon. Container routing is automatic, based on container destination address. If assigned target is not ready to receive a container, then the container is returned back to *Coordinator* holon. State diagram of *Transportation* is on fig. 3.

Transportation tasks:
- routing of containers,
- detection of presence and failure (state) of *UASs*.

Events:
- container arrival to *Transportation*,
- container departure from *Transportation*,
- container delivery to *UAS*,
- container from *UAS*,
- change of state of *UAS*.

Local information resources:
- list of *UASs* (connection points and state),
- list of containers in *Transportation*.

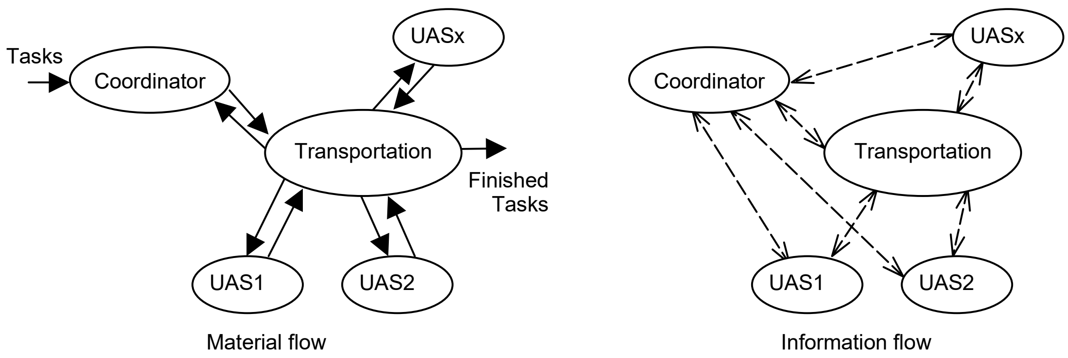

Fig. 1 Material and information flow in holonic assembly system

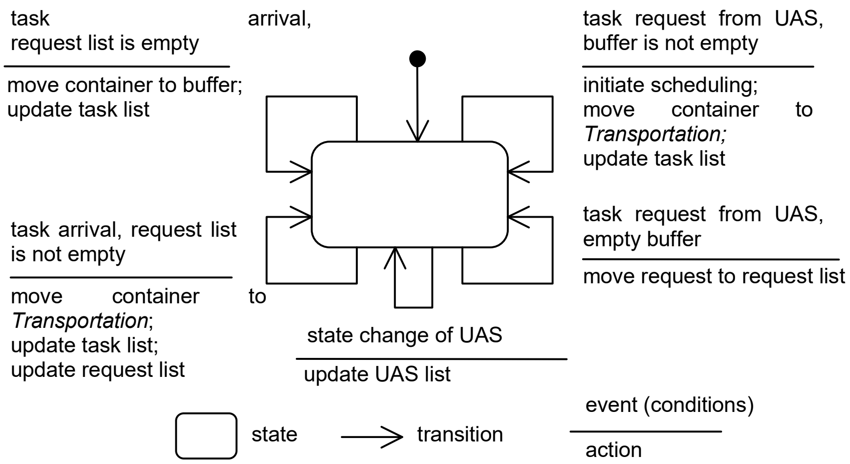

Fig. 2 State diagram of *Coordinator* holon

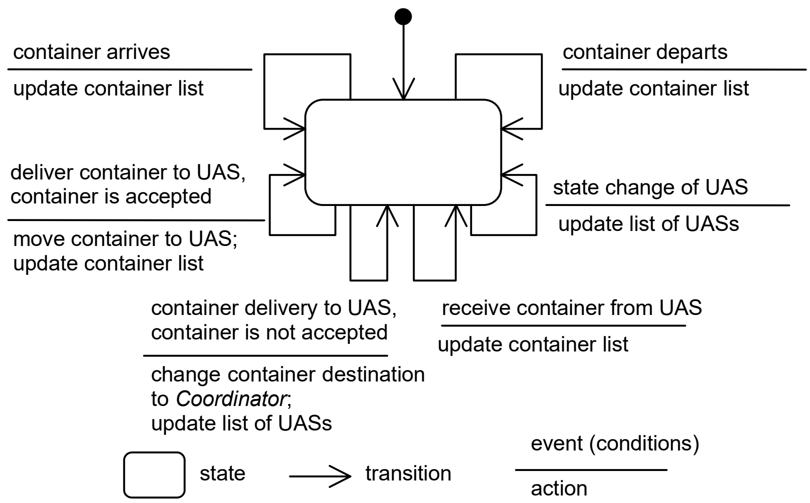

Fig. 3 State diagram of *Transportation* holon

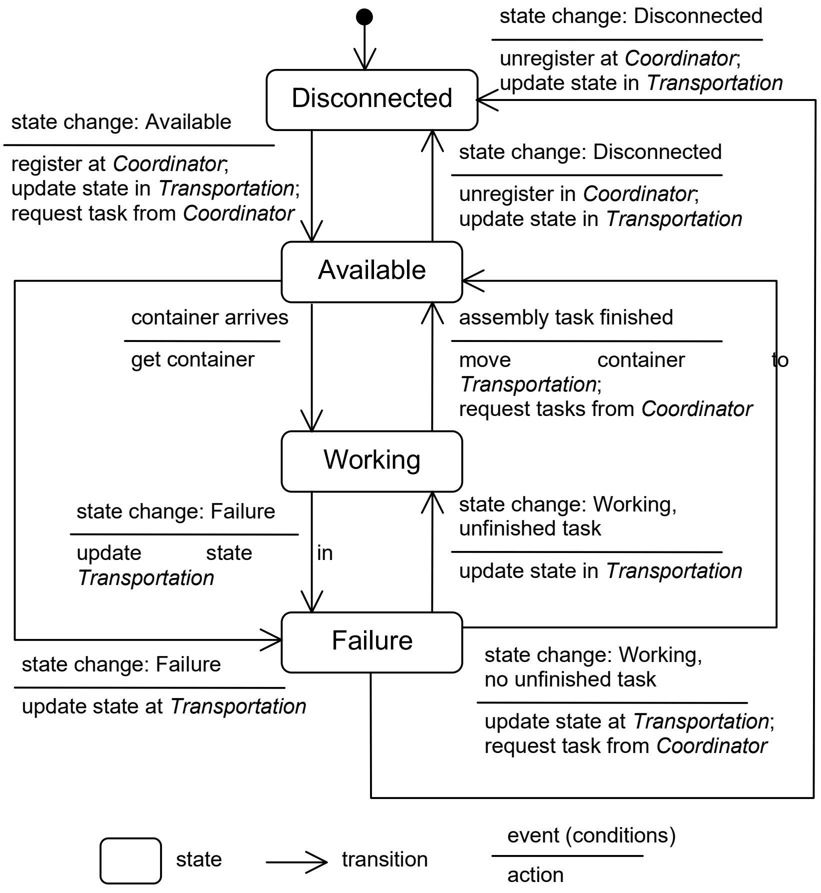

Fig. 4 State diagram of *UAS*

3.4 Universal Assembly Station (UAS)

Universal assembly station is able to assemble the products from supplied parts. *UASs* request available assembly task from *Coordinator* holon and get an opportunity to participate in the distributed scheduling by choosing assembly tasks according to a local criterion (objective function). State diagram of *UAS* is on fig. 4.

UAS tasks:
- perform assembly task,
- schedule tasks.

Events:
- container arrival,
- assembly task finished,
- change of state (Disconnected, Available, Working, Failure).

Local information resources:

- state of *UAS*,
- current assembly task.

3.5 Distributed Scheduling

The arriving containers are distributed to individual *UASs* according to a schedule. The quality of a schedule is evaluated by the average output rate (which is inversely proportional to the average assembly time) during the observed period. Optimal scheduling exists only for stable conditions. In case of a failure of an element, rescheduling must take place. Unlike in centralized planning with global objective function for optimisation, decentralized scheduling uses local objective functions. In our study, each *UAS* has its own scheduling algorithm and objective function.

This paper presents rule based strategy for scheduling. Each *UAS* j is choosing task i where assembly time T_{ij} is minimal for that particular *UAS* j:

$$T_{ij} = \min(T_{1j}, T_{2j}, ..., T_{nj}) \qquad (1)$$

where

T_{ij} is assembly time of container i for *UAS* j,

n is number of available assembly tasks for planning at the moment when *UAS* j requests a task.

The rule is implemented in the following scenario:

1. *UAS* which is free or just finished a previous task asks the *Coordinator* holon for the next task.
2. *Coordinator* holon sends back the list of available tasks in the input buffer at that moment. If there is no task then *UAS's* request is moved to request list and *UAS* waits for a task.
3. *UAS* choose one task according to its local criterion.
4. *Coordinator* sets destination of container which corresponds to chosen task to *UAS* and send it to *Transportation*.
5. *Transportation* ensures delivery of container to *UAS*.

This scenario has modifications in case of *UAS* breakdown, maintenance work, or disconnection as well as for the case of an empty container buffer. All different scenarios can be derived from state diagrams of the holonic system elements.

4. SIMULATION RESULTS

Simulation software SIMPLE++ was used for the simulation of the holonic assembly system. The total number of assembly tasks was 200 and number of *UASs* was 7. Assembly tasks arrived one by one and time between arrivals was normally distributed ($\mu = 6$ min, $\sigma = 2$ min, $t > 0$). Assembly time for a particular task was different for each *UAS*. The mean value for all 200 tasks and 7 *UASs* was 55 minutes, with minimum of 21 and maximum of 101 minutes.

A restricted number of available assembly tasks was considered for scheduling when request from individual *UAS* came. Figure 5 illustrate part of the computer model.

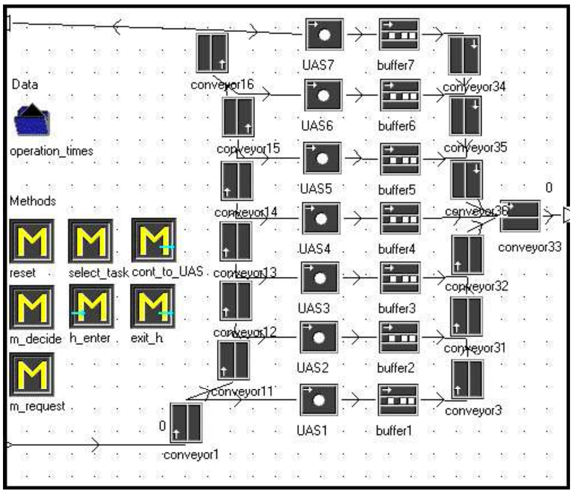

Fig. 5 Illustration of computer model – *Transportation* and *UASs* (*Transportation* holon consists of conveyor 1-33)

The following experiments were performed to demonstrate the properties of the designed holonic assembly system:

1. *UASs* operate all the time, and the number of available tasks for scheduling is restricted to a certain maximal value defined by size n of the input buffer. That is, the list of available tasks offered to individual *UAS*s arisen from the content of the buffer. Seven simulation runs were done for different maximum value from $n = 1$ to 200. The input buffer was filled up at the beginning of each simulation run.
2. Random *UASs* breakdown. Tasks that are not accepted are returned to *Coordinator* holon and are available for other *UASs*. The rest of the conditions are identical with experiment 1.

Results of the experiment 1 are in fig. 6. The chart shows the average output rate of the holonic assembly system for various numbers of available tasks. The output rate varies with number of available tasks for scheduling. We conclude that the proposed scheduling rule is satisfactory already for a relatively small number of available tasks ($n = 10$). Quasi optimal output rate was obtained by a genetic algorithm for all 200 tasks.

Fig. 7 shows the system's behaviour in case of *UASs* breakdown, maintenance work, or temporary inaccessibility due to capacity redistribution among shop-floors (experiment 2). Time t_1 between breakdowns of each *UAS* and duration of breakdowns t_2 were random variables with lognormal distribution ($\mu_1 = 10$ *hours*,

$\sigma_1 = 5$ *hours* ; $\mu_2 = 2$ *hours* , $\sigma_2 = 1$ *hour*). The *UASs* breakdowns slowed down the assembly.

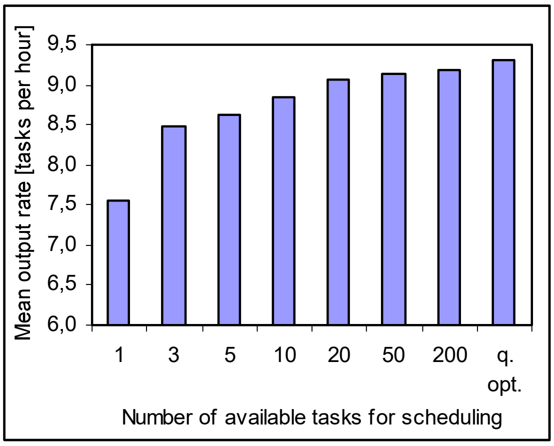

Fig. 6 Mean output rate versus number of available tasks for scheduling (q. opt is the quasi optimal value obtained by the genetic algorithm)

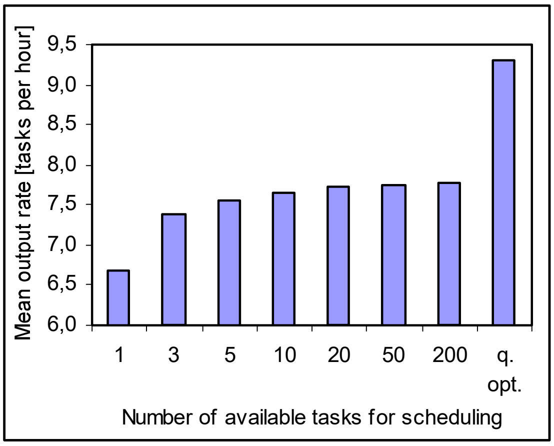

Fig. 7 Mean output rate at random breakdowns of *UASs* versus number of available tasks for scheduling (q. opt is quasi optimal value without breakdowns)

5. CONCLUSION

The proposed holonic assembly system can dynamically redistribute assembly tasks according to the available number of universal assembly stations. The holonic assembly system consists of *Coordinator* holon, *Transportation* holon and *Universal assembly stations* whose behaviour was described in details by state diagrams. The system's structure was based on functional grouping. The behaviour of the entire system assures high output rate (short assembly time) by decentralised rule based scheduling which used restricted number of available tasks. Simulation proved function of such systems even in conditions of breakdown, maintenance work or capacity redistribution of universal assembly stations. The results showed that already a relatively low number of available tasks for scheduling gives output rates comparable to those obtained by a genetic algorithm.

ACKNOWLEDGMENTS

The paper shows some results of the research project VEGA 1/1093/04 "Research of Basic Properties of Holonic Manufacturing Systems" (Scientific Grant Agency of the Ministry of Education of Slovak Republic and the Slovak Academy of Sciences).

REFERENCES

[1] Fletcher, M., Brennan, R.W., Xu, Y.: How Intelligent Manufacturing Holons Configure Themselves. In Proceedings from IASTED International Conference on Intelligent Systems and Control, Clearwater, Florida. November 19-22, 2001 pp. 37-42. http://iit-iti.nrc-cnrc.gc.ca/iit-publications-iti/docs/NRC-44931.pdf.

[2] Fletcher, M. (2000): Some Remarks on Agent-based Holonic Assembly Systems. International Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce (MAMA'2000), International ICSC Congress on Intelligent Systems and Applications (ISA'2000), Wollongong, Australia, December 11-15, 2000, Vol 1, pp. 710-716.

[3] Ulieru, M.: FIPA-ENABLED HOLONIC ENTERPRISE. http://www.enel.ucalgary.ca/People/Ulieru/Publications/Nice-paper.doc, 2003 (18.3.2004).

[4] Matuszek, D. – Plinta, D. – Bubeník, P.: Modelling and Simulation in Managing Production processes. In: Computer integrated manufacturing, Editors: Skolud, B., Krenczyk, D., Warszawa: Wydaw. Naukowo-Techniczne, 2003, pp. 361-368, ISBN 83-204-2850-5

BIOGRAPHY

Peter Bober graduated (MSc.) in 1987 with distinction at the Department of Technical Cybernetics of the Faculty of Electrical Engineering at Technical University of Košice. He defended his PhD. in the field of electric drives in 1993; his thesis title was "Digital Control of Induction Drive". From 1994 to 1997 he held a position of assistant professor at the Department of Electric Drives. Since 1997 he is working at the Laboratory of Industrial Engineering. His scientific research focuses on modelling and simulation in various application areas.

Michal Girman graduated (MSc.) in 1970 at Faculty of Electrical Engineering, Brno University of Technology. He defended his PhD. at The Faculty of Electrical Engineering, Technical University of Košice in 1981. Since 1970 he is working at the Faculty of Electrical Engineering of Technical University of Košice as assistant professor and from 1983 as an associate professor. Since 1997 he is the head of the Laboratory of Industrial Engineering. His scientific research focuses on production control and production control systems.