Proceedings of the 1999 IEEE
International Symposium on Assembly and Task Planning
Porto, Portugal - July 1999

DISTRIBUTED TASK SCHEDULING

Paulo Sousa
psousa@dei.isep.ipp.pt

Carlos Ramos
csr@dei.isep.ipp.pt

Departamento de Engenharia Informática
Instituto Superior de Engenharia do Porto (ISEP/IPP)
Rua Dr. António Bernardino de Almeida 4200-072 PORTO - Portugal
Phone: + 351 (02) 834 05 00 - Fax: + 351 (02) 82 11 59

Abstract

This paper proposes an architecture for manufacturing enterprises based on the Holonic concept. The main advantage of this approach is the decentralised, distributed nature of the system as well as its heterarchic organisation as opposed to the rigid hierarchic and centralised nature of Computer Integrated Manufacturing (CIM). The system models Orders, Products, Resources and Tasks and the Process Planning, Scheduling and Order Management functions of the enterprise. A Negotiation Protocol is defined for task assignment that is an extension of the Contract Net protocol. The protocol is based on the Task-Resource abstraction and is suited for negotiation in several situations not only for the dynamic scheduling of manufacturing orders. The negotiation protocol is able to handle exceptions such as machine breakdowns and rush orders.

Introduction

During the last years it was felt a need to deal with new trends on Manufacturing Systems. These new directions include the reduction of order's dimension; increasing product's complexity; increasing product's variety; client's participation in the development process; as well as the reduction of product's life cycles. Thus, Manufacturing Systems need to handle the dynamic nature of demands.

Physically, the Manufacturing System involves several resources (numeric control machines, robots, AGVs, conveyors) and from the logical point of view several tasks can be carried out at the same time. Since Manufacturing Systems correspond to a distributed system from the physical and from the logic point of view, it is natural that the solution for novel Manufacturing systems is a distributed and not centralised one. Due to these reasons the framework of Distributed Artificial Intelligence (DAI) and Holonic Manufacturing Systems (HMS) is proposed. These concepts allow modelling a system as a set of intelligent, autonomous and co-operative elements in order to achieve reconfigurable and extensible systems.

The idea behind Holonic Manufacturing Systems (HMS) is to provide a dynamic and decentralised manufacturing process, in which humans are effectively integrated, so that changes can be made dynamically and continuously. HMS are based on the notion of Holon [Koestler, 1967]: combination from Greek *holos* (whole) with the suffix *on* which, as in prot*on* or neutr*on*, suggests a particle or part.

A Holon means simultaneously a whole and a part of the whole. Thus a Holon can be made up of other holons. A Holon is autonomous and co-operative and sometimes intelligent. Each production unit (e.g. Numeric Control machine) can be a Holon and these holons co-operate with each other in order to manufacture products.

As it was observed by Valckenaers et al. (1994) it is expected that rigid, static and hierarchical manufacturing systems will give way to systems that are more adaptable to rapid change. As an alternative to hierarchy it appears the concept of holarchy as a system of holons that can co-operate to achieve a goal or objective. A HMS is a holarchy that integrates the entire range of manufacturing activities from order booking through design, production, and marketing to achieve the agile manufacturing enterprise.

The paper is organised as follows. Section 2 presents the developed framework for holonic systems and the architecture of an holonic manufacturing enterprise. In section 3 a negotiation protocol is presented for the task allocation problem. Section 4 introduces some considerations regarding exceptions. Finally conclusions are drawn in section 5.

Holonic Architecture

Related Work

Research works in this field differs in what is chosen as building blocks. Bongaerts et al. (1996) identified three types of holons a HMS should have: (i) Products; (ii) Resources and (iii) Orders. According to them, *"for a minimalist implementation of a manufacturing system it* *suffices to have a holarchy of product related holons, resource related holons and order related holons*".

Kouiss et al. (1997) presented a multi-agent system for dynamic scheduling in Flexible Manufacturing Systems (FMS). Each agent models a *Work Centre* in the FMS, and schedules operations by locally applying dispatch rules.

AARIA's architecture [Parunak et al., 1997] uses agents representing *Parts*, *Resources* and *Unit Processes* for scheduling in discrete manufacturing. Each agent is equally intelligent and responsible. Parts move through Unit Processes and Buffers. Each Unit Process picks up one or more part from buffers, and employs resources to produce an output part.

Gou and Luh (1997) follow a more modular approach using eight types of holons: (i) *Product*; (ii) *Part*; (iii) *Machine-Type*; (iv) *Machine*; (v) *Cell Coordinator*; (vi) *Cell*; (vii) *Factory Coordinator*; and (viii) *Factory*. In their work, Product Holons represent manufacturing orders, and are divided in Part Holons, which represent a stage in the manufacturing process of that product. A Cell Holon is an holarchy composed of Cell Coordinator, Machine, Machine-type and Part Holons.

Enterprise Architecture

This section extends the work done by Sousa and Ramos (1998) by further explaindeing each holon of the manufacturing system.

Figure 1 represents a manufacturing system, showing several functional units of the enterprise: Process Planning (PPH), Order Management (OMH) and Production Scheduling & Control (SH). This upper level holons aggregate several types of holons needed for their jobs. As the figure shows, some holons participate in more than one upper level Holon. The "basic" holons modelled by the system are: Customer Orders (COH); Products (PH); Resources (RH) and Tasks (TH). Two assistant holons exist: the Task Manager (TM) and the Directory Service (118).

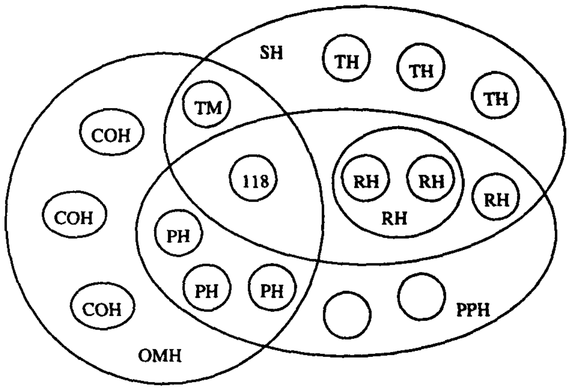

Figure 1 - Holonic Architecture for a Manufacturing System

The *118 Holon* acts as the central directory for the enterprise. Each Holon registers its abilities in this database, and asks for other holons when ever necessary.

The *Customer Holon* represents the customer's interests and manages all data regarding the customer's account (e.g. ABC rank, total sales value, etc.). This holon is also responsible for quantifying the customer's satisfaction value for each action the enterprise takes directly relating him (e.g. order delivery delay).

An *Order Holon* is created for each order a customer makes. It represents a demand for production to the enterprise. An order can consist of several products and specific conditions regarding quality, price or any other constraints the customer imposes.

A *Product Holon* represents each item the enterprise is able to manufacture. It also represents the logistic concerned with stocks and manufacturability. If for instance an operation is not available in the system (e.g. due to tool breakdown) the Product Holon is responsible for providing an alternative and informing it to the Task Holons.

The list of `Plan` terms represents alternative plans according to different optimisation criterions (e.g. total processing time, material cost, etc.) to produce the item. Plans are generated by a system called TPMS - Task Planning for Manufacturing Systems [Rocha and Ramos, 1996]. Each plan has a list of tuples indicating for each operation in the plan the predecessor and successor operations, as well as any preferences relative to resources.

A *Resource Holon* represents the current situation (execution status; delivered activity; and activity to be done) of a physical (or human) resource. The activity of the resource is represented into an agenda - the sequence of operations to be carried out (Ramos et al., 1995). A resource holon can represent a basic resource (e.g. a milling machine) or a work cell (e.g. a milling machine, a vision system and a robot) in which case it aggregates two or more Resource Holons, one for each resource the work centre has. In the manufacturing case the resource's abilities are the operations it can perform (e.g. drill, mill, etc).

An example of Resources and agendas is the following:

```
(milling_1, [], [cell_2],
[mill_100rpm, mill_350rpm,
mill_500rpm],
[
(1998.05.22-09:50, 10s,
mill_350rpm, task_2704_98,
done),
(1998.05.22-10:00, 1260s,
mill_100rpm, task_2394_98,
doing),
(1998.05.22-12:00, 20s,
mill_100rpm, task_5769_98,
to_do)
])
```

Task Holons represent the possibilities to execute a plan for a task into the plan structure [Ramos, 1996]. A Task Holon is used to make *n* items of a certain *product*. A Task Holon is either originated by a customer's order or for stock balance. A Task Holon instantiate a plan to achieve that goal, contracting resources for executing each operation it needs. A Task Holon has a birth and death, during its existence each of these holons is responsible for monitoring the progress of the work. After completion of the task, the Task Holon "dies" and enters a cryogenic state - long term information about a dead holon that is able to reborn (e.g. for data mining).

The *Task Manager* is the user interface of the system. This Holon is responsible for launching Task Holons whenever a new task is ordered. Besides that, the Task Manager is responsible for dealing with dynamic changes of task conditions (e.g. when the user changes the deadline of a task). This holon is not a central control point, task's control is delegated to Task Holons.

The *Process Planning Holon* aggregates Resource Holons and Product Holons and is responsible for the generation of production plans for each product, taking into account the existing resources and engineering data from CAD systems used to develop the product.

The *Order Management Holon* handles customers' orders, aggregating Customer Holons, Order Holons and Product Holons.

The *Production Scheduling & Control Holon* aggregates Resource Holons and Task Holons for the dynamic scheduling of manufacturing tasks, as well as execution control of those tasks.

Infrastructure

The developed Holons were built using the HFW (Holonic Framework) infrastructure defined by Silva et al. (1998). Figure 2 illustrates the internal structure of a HFW Holon, consisting of two levels: the framework level, and the application level.

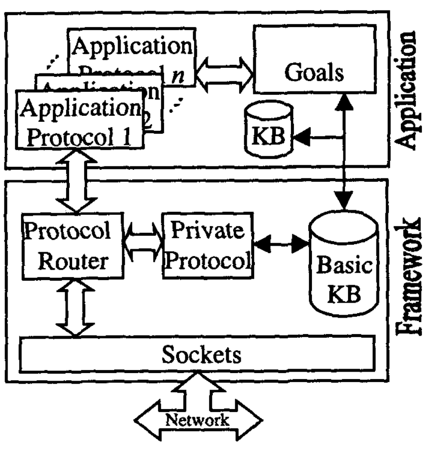

Figure 2 - Framework's basic structure

The framework provides asynchronous communications, directory services and support for several protocols. A *protocol* is a set of messages and rules (defining which messages make sense at any given time), that allows two holons to communicate with each other. The use of object oriented mechanisms allows communication messages to be perfectly routed and processed. The *directory services* are implemented in a per-holon basis, and act as a knowledge base of other holons (their location, abilities and confidence degree).

The framework is also prepared to have a special kind of Holon that implements a centralised directory of every Holon in the system, their location and abilities. This centralised directory does not imply the absence of per-holon directories since those are still needed for each Holon to save its own degree of confidence on each other Holon it co-operates with. For this feature each Holon should register its abilities on start-up and dynamically update that information, if necessary, along the execution period.

At the application level, each specific Holon is responsible for defining its protocol(s) and sensor/actuator procedures. It's also responsibility of this level to interact with the user if necessary. The application is event-driven by the framework, so it is easy to implement the protocols as a finite state machine with arriving messages as input.

Negotiation Protocol for Task Assignment

The protocol is divided in 4 phases: (i) task announcement; (ii) forward influence; (iii) backward influence; and (iv) contracting.

Figures 3 and 4 are diagrams of the exchanged messages in this phase. First, the Task Manager creates a new Task Holon and announces it what to do. The recently created Task Holon will ask the Process Planning Holon for a basic plan - and some alternative plans too - for the manufacturing of the requested item. The task holon will then contact all the resources able to perform each operation, informing about resources contacted for predecessor and successor operations.

When a Request Announcement message arrives, the holon looks at its agenda and schedules the requested operation. The scheduling is done based on locally defined dispatch rules (and globally defined scheduling policies). The scheduling algorithm has been adopted from a method developed for centralised dynamic scheduling that is well described in [Ramos et al., 1995].

After that, Resource Holons with no predecessors will pass a "influenced" list of time intervals to the successor Resource. This list is obtained by shifting the reserved intervals by the duration of the operation. For each *List of Intervals* message received, the Resource Holon will combine (e.g. intersect) it with its agenda obtaining a new one, that will be the basis for the "influenced" list of intervals it will pass to successor Resource Holons. The forward influence phase stops when all resources with no successor receive the list of intervals from their predecessors.

The backward influence phase is similar to the forward influence but in the opposite direction. The backward influence phase ends when the Resource Holons for operations with no predecessor receive the list of intervals from the operations succeeding them.

When each Resource Holon has the final agenda, it responds to the Task Holon with a ***Resource Bid*** message, which includes information about the time intervals where the operation can be scheduled.

After receiving all Resource Bid messages the Task Holon selects one Resource Holon that is able to handle each operation. This selection is based on heuristic rules (e.g. the resource having more free time till the deadline). If the co-ordination between resources is not possible, the Task Holon tries to change the selection of Resource Holons for the operations.

Handling Exceptions

Renegotiation is needed whenever exceptions arise (e.g. a resource malfunction), a task's priority changes, or a new priority task enters the system.

When a Resource Holon detects a malfunction that can not be recovered, it decides to inform each Task Holon that has contracted its work. After receiving the

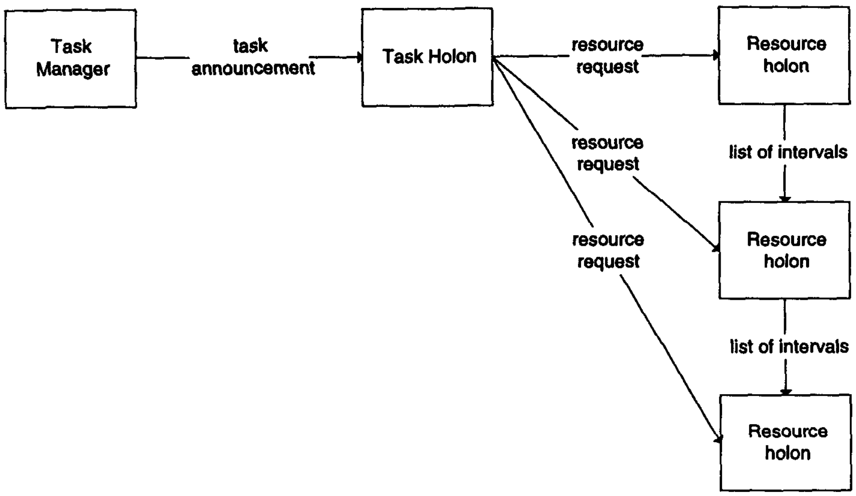

Figure 4 - Task announcement and forward influence

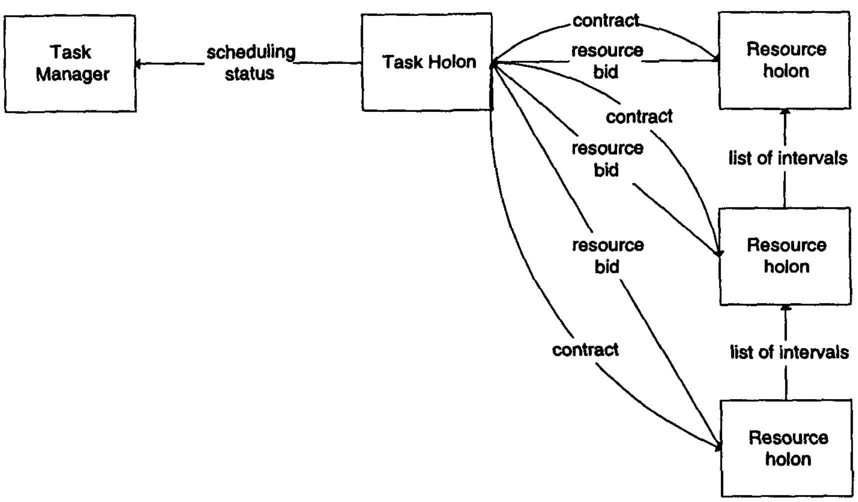

Figure 4 - Backward Influence and contracting

Machine Fault message from the Resource Holon, the Task Holon will start negotiations with other Resource Holons capable of performing those operations it needs.

In some cases the Resource Holon may analyse the situation and decide not to inform the Task Holons, but instead tries to recover from the malfunction and execute its work with a slight delay.

If a new task enters the system and has to be scheduled in some time window already occupied by other task, a special kind of renegotiation is needed. The new Task Holon has to persuade (or demand) the other Task Holons to give it the resource time it needs. Figure 5 shows a resource's agenda before and after a situation like this.

Conclusions and Future Work

An architecture, based on holonic concepts, was proposed. This architecture resembles the distributed nature of manufacturing, thus allowing for a better modelling of the system.

While other co-operative communities operate with Agents representing resources or systems, the architecture proposed here combines Resource-based Holons with Task-based Holons. The main advantage is the easy access to task activities that are supported in Task-based Holons.

The Negotiation Protocol based on the Contract Net Protocol is suitable for the dynamic scheduling of manufacturing tasks. This Negotiation Protocol is able to deal with deadlines and exceptions. Exceptions or dynamic changes to the system can trigger a renegotiation phase. This renegotiation allows the system to adapt itself.

Future Work includes running the system with some publicly available benchmarks for a better comparison with other distributed and centralised researches.

Another point of action will be testing the system with real data from a manufacturing enterprise (negotiations are being made with companies in the foundry and textile industry).

References

Bongaerts, L.; Wyns, J.; Detand, J.; Van Brussel, H. and Valckenaers, P. (1996). *Identification of Manufacturing Holons.* Proceedings of 1st European Workshop on Agent Oriented Systems in Manufacturing, pp. 57-73. Berlin, Germany, 26-27 September.

Davis, R. and Smith, R. (1983). *Negotiation as a metaphor for distributed problem solving.* Artificial Intelligence, vol. 20, n. 1, pp. 63-109.

Gou, L. and Luh, P. (1997) *Holonic Manufacturing Scheduling: Architecture, Cooperation Mechanism, and Implementation.* Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Tokyo, Japan, June 16-20.

Koestler, A. (1967). *The Ghost in the Machine.* Hutchinson & Co, London.

Kouiss, K.; Pierreval, H. and Mebarki, N. (1997) *Using multi-agent architecture in FMS for dynamic scheduling.* Journal of Intelligent Manufacturing, vol. 8, pp. 41-47. Chapman & Hall.

Parunak, H.; Baker, A. and Clark, S. (1997). *The AARIA Agent Architecture.* Proceedings of the International Conference on Autonomous Agents (ICAA'97), Marina del Rey, CA, February 6-8.

Ramos, C. (1996). *A Holonic Approach for Task Scheduling in Manufacturing Systems.* IEEE International conference on robotics and Automation, Minneapolis, USA.

Ramos, C.; Almeida, A. and Vale, Z. (1995). *Scheduling Manufacturing Tasks considering Due Dates: a new method based on Behaviours and Agendas.* International. Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems; Melbourne, Australia.

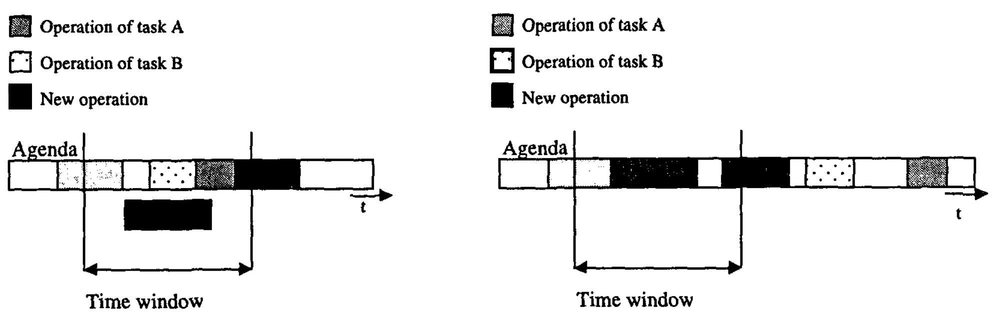

Figure 5 - Rescheduling for a new operation (a) the request; (b) reallocation

Rocha, J. and Ramos, C. (1996). *Plan Representation and Heuristic Generation of Operation Sequences for Manufacturing Tasks.* Data and Knowledge Systems for Manufacturing and Engineering; Tempe, Arizona, USA.

Silva, N.; Sousa, P. and Ramos, C. (1998) *A Holonic Manufacturing System Implementation.* Proceedings of the Advanced Summer Institute (ASI'98). Bremen, Germany, 14-17 July.

Sousa, P. and Ramos, C. (1998). *A Dynamic Scheduling Holon for Manufacturing Orders.* Journal of Intelligent Manufacturing - special issue on agent based manufacturing, vol. 9 n. 2, pp. 107-112. Chapman & Hall. ISSN 0956-5515.

Valckenaers, P. and F. Bonneville, H. Van Brussel, L. Bongaerts and J. Wyns, 1994, Results of the Holonic Control System Benchmark at K.U. Leuven, Rensselaer's 4th International Conference on Computer Integrated Manufacturing and Automation Technology.