MULTIAGENT DESIGN FOR DYNAMIC JOB-SHOP SCHEDULING USING PASSI

Claudio Cubillos, Silvana Roncagliolo and Leonardo Espinoza
Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso, Av. Brasil 2241, Valparaíso, Chile

Keywords: AOSE, Planning & Scheduling, Agent System.

Abstract: The present work details the experience on designing a multiagent system devoted to a dynamic Job Shop setting using the PASSI methodology. The agent system is in charge of the planning and scheduling of jobs and their operations on a set of available machines, while considering the materials assigned to each operation. Dynamicity concerns job orders scheduling on-the-fly and the re-schedule caused by changes to the original plan due to clients, machines or material stocks. The system has been modeled with the PASSI Toolkit (PTK) and implemented over the Jade agent platform.

1 INTRODUCTION

Agent paradigm has leveraged as an important modeling abstraction, in areas such as web and grid services, peer to peer and ambient intelligence architectures just to mention some cases. In a multiagent system (MAS) diverse agents communicate and coordinate generating synergy to pursue a common goal.

Hence as modeling artifact, agent-based systems borrow their key characteristics from us, humans, and our societies. This higher level of abstraction has allowed agents to tackle the increasing complexity of nowadays open software systems where integration, transparency and interoperation among heterogeneous components are a must.

For this technology to get more mature and widespread, the use of agent-oriented software engineering (AOSE) methodologies and tools are a key factor of success. The present work describes the design of a multiagent system using a particular AOSE methodology called PASSI.

The chosen domain for the system corresponds to the job-shop scheduling problem under a dynamic scenario in which job requests coming from clients must be processed on-the-fly and where changes can occur due to changes in the environment (e.g. materials stock-out), problems with the service (e.g. machine delays or break down) or client eventualities (e.g. due-date change, job cancellation).

2 RELATED WORK

Diverse proposals of agent-based systems can be found in literature tackling the job-shop or production scheduling problem.

In (Fischer, 1994) Fischer proposed a hierarchical planning structure consisting of six layers: the layer of the production planning and control system, the layer of the shop floor control system, the task coordination layer, the task planning layer, the task execution layer and the machine control layer.

In (Saad et al., 1995) a Production Reservation approach was proposed by using a bidding mechanism based on the Contract Net Protocol - CNP (Smith, 1978) to generate the production plan. Other recent CNP-based solutions can be found in (Váncza, 2000) (Maturana et al., 1999) (Lim, 2002) and (Usher, 2002).

In AARIA (Parunak et al, 1997), the manufacturing capabilities (e.g. people, machines, and parts) are encapsulated as autonomous agents and use a mixture of heuristic scheduling techniques: forward/backward scheduling simulation scheduling, and intelligent scheduling.

In (Maturana et al., 1999) the adaptive multi-agent manufacturing system architecture called MetaMorph combined the CNP with mediator-centric federation architecture was presented.

Although the above works mention the agent approach for designing the overall system, neither formal models nor AOSE methodologies are used for their design mainly due to a lack of maturity AOSE methodologies at that time.

Cubillos C., Roncagliolo S. and Espinoza L. (2008).
MULTIAGENT DESIGN FOR DYNAMIC JOB-SHOP SCHEDULING USING PASSI.
In *Proceedings of the Fourth International Conference on Web Information Systems and Technologies*, pages 288-291
DOI: 10.5220/0001528502880291
Copyright © SciTePress

Therefore, one of the contributions of the present work is to provide a more formal design of multiagent system devoted to passenger transportation using the PASSI methodology, one of the most relevant agent-oriented software engineering methodologies nowadays.

3 THE DYNAMIC WORK-SHOP PROBLEM

The traditional Job-Shop Scheduling Problem (JSSP), can be described by a set of *n* jobs $\{J_j\}_{1 \leq j \leq n}$ which is to be processed on a set of *m* machines $\{M_r\}_{1 \leq r \leq m}$. Each job has a technological sequence of machines to be processed. The processing of job J_j on machine M_r is called the operation O_{jr}. Operation O_{jr} requires the exclusive use of M_r for an uninterrupted duration p_{jr}, its deterministic processing time, and each operation O_{jr} has pre-assigned materials $\{W_i\}_{1 \leq i \leq k}$. In addition, each job has a due-date $\{D_j\}_{1 \leq j \leq n}$. A schedule is a set of completion times for each operation $\{c_{jr}\}_{1 \leq j \leq n; 1 \leq r \leq m}$ that satisfies those constraints. The considered JSSP involves the scheduling of n jobs J on the m machines M and consuming k materials W while minimizing the total tardiness regarding the due-dates.

On the other hand, the dynamic variant of the problem adds the fact that the jobs to be processed are not known in advance and that they must be scheduled as they arrive. In addition, environmental changes can cause some events to happen, such as a machine delay or breakdown, a client canceling/modifying a job order, among others. Therefore, the objective is to obtain a schedule that also minimizes the number of non-placed jobs. It is one of the most hard NP-complete combinatorial optimization problems.

4 THE AGENT ARCHITECTURE

The multiagent job-shop scheduling system stands over the Jade Agent Platform (Bellifemine et al., 1999), and was modelled with PASSI, an agent-oriented software development methodology, supported by the PASSI ToolKit (PTK, 2005) to be used as an add-in for Rational Rose.

4.1 Agent Identification (A.Id.)

The diagram in Figure 1 shows the identified use cases for this job-shop system and the leveraged agents.

Firstly, the Client agent is a GUI agent in charge of the communication between an actual client and the rest of the system, providing the possibility of generating a job order, and to communicate inbound/outbound eventualities regarding such order due to changes in the environment (e.g. order modification/cancellation from client, order delay/reject from the system).

Order agents are devoted to the job order management, its breakdown into operations, the request of necessary materials for each operation execution to Stock agents, and the request to Machine agents for the scheduling of each operation.

Machine agents encapsulate each real machine, being primarily in charge of its schedule management. This involves processing requests coming from Order agents and performing the scheduling process. It also manages eventualities that can cause schedule changes due to the machine itself (e.g. delay or breakdown), due to material supply problems (e.g. delay on delivery), or even due to client changes (e.g. order cancellation or modification). For all the above, it carries out a search in the solutions state space by implementing an optimization heuristic. In the actual system, a search algorithm presented by (Yoo et al., 2002), inspired in simulated annealing was implemented.

Finally, Stock agents main goal are to maintain adequate levels of materials by generating supply orders (to the Stock Administrator), and to provide the necessary materials for the execution of each operation.

4.2 Roles Identification (R.Id.)

Roles Identification consists in exploring all the possible paths of the preceding Agents' Identification Diagram in Figure 2. In fact, each “<<communicate>>” relationship among two agents can be in one or more scenarios showing interacting agents working together to achieve certain desired system behavior.

As example, the following Figure 3 shows the scenario in which the Client actor requests the system to process its job. Each object in the diagram is described following the <*role*>:<*agent*> convention.

This scenario involves all actors and agents in the system. It starts with the Client requesting its order processing through the GUI (Client agent), the interface generates a *JobOrder* with all the details and forwards the request to the manager role of the Order agent. This agent breaks down the job obtaining an *Operation List* containing the sequence of required operations and related materials. The manager role requests the Stock agent for the availability of the materials needed for each *Opera-*

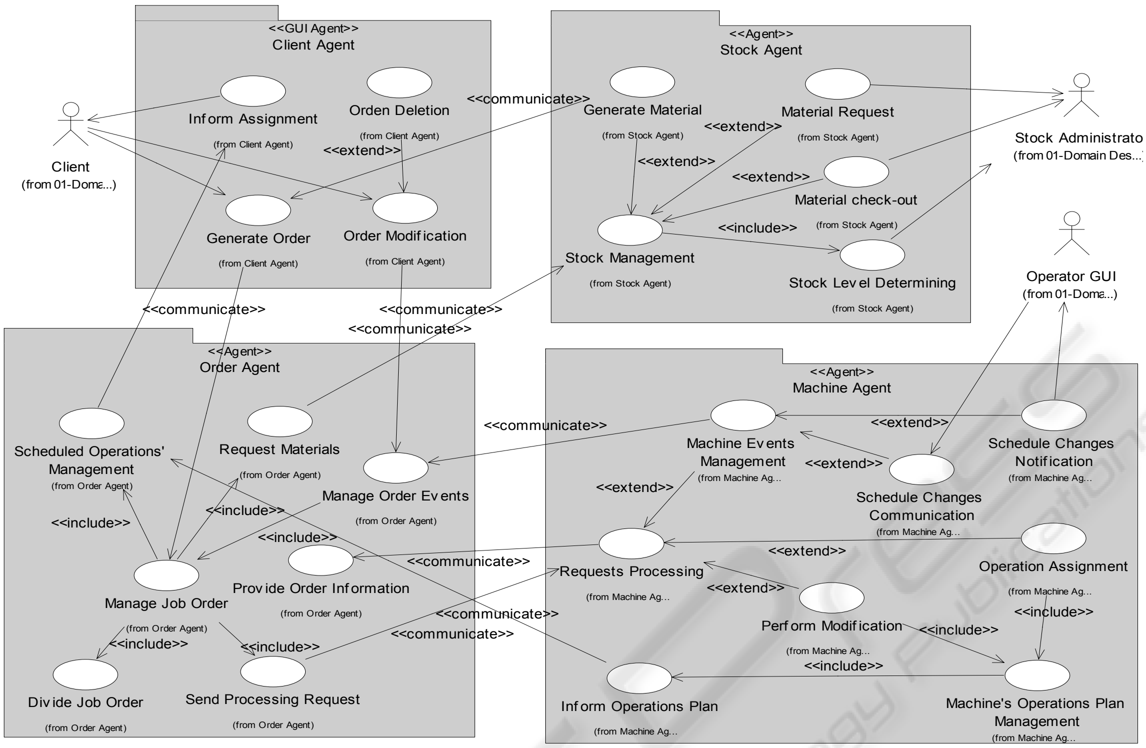

Figure 1: Agent Identification Diagram for the Job-shop scheduling system.

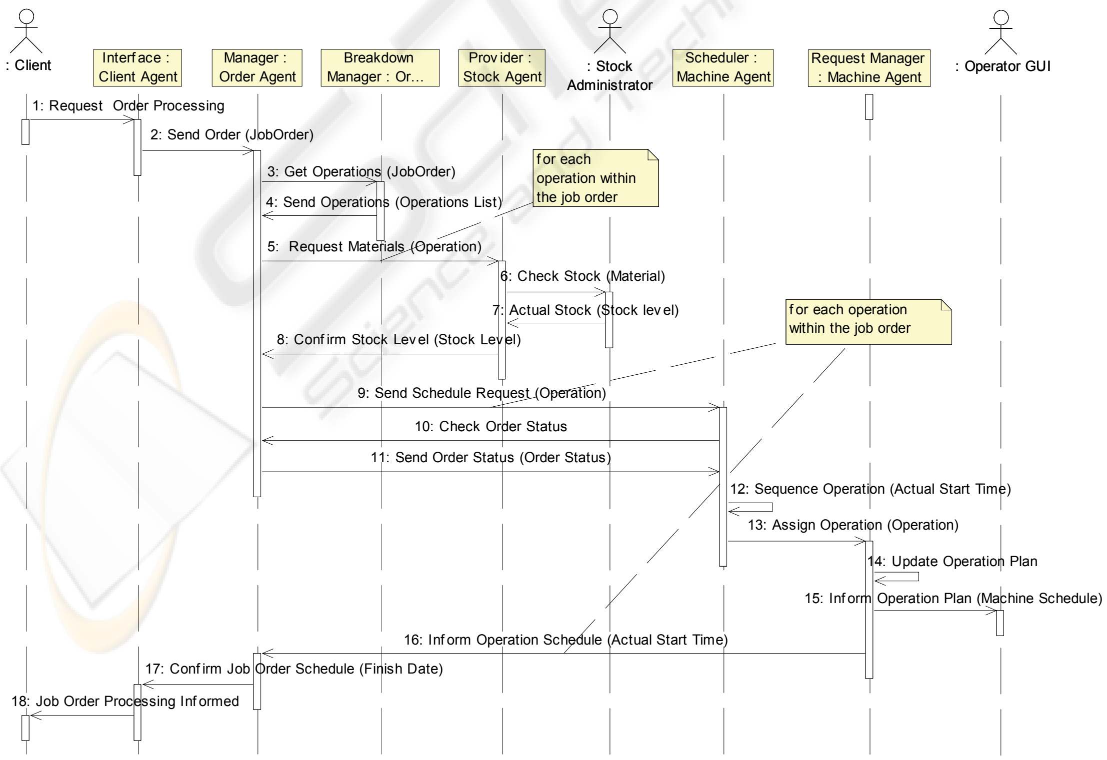

Figure 2: Roles Identification Diagram for the “Client Requests Job Processing” scenario.

SCITEPRESS Science and Technology Publications

tion. On its turn, the Stock agent checks the *Stock Level* of each item in the operation's *Material List* and reserves the required amount.

With the materials availability confirmation the Order agent starts sequentially requesting Machine agents to schedule their operations. The scheduler role of the Machine calls the optimization heuristic to search for feasible alternatives selecting the best one according to the active *UtilityFunction*. Once found, the operation is programmed into the actual *MachineSchedule*, then new operation schedule is informed to the operator (through the Operator GUI actor) and the corresponding Order agent is informed about the scheduled operation indicating its *Actual Start Time*.

Finally, once all operations successfully scheduled, the Order agent informs the Client about the *FinishDate* for the entire job.

5 CONCLUSIONS

The design of an agent-based software architecture for dynamic job shop scheduling was described.

The agent formalization with PASSI promotes the system maintainability, its ability to cope with newer requirements and the possibility to scale and integrate other actors and systems.

Next steps consider the implementation of diverse scheduling algorithms (e.g. genetic algorithm, tabú search, SA) for job-shop in order to refine and demonstrate the extensibility of our solution.

ACKNOWLEDGEMENTS

This work is part of Project No. 209.746/2007 entitled "Coordinación en una sociedad multiagentededicada a la programación y control bajo ambiente dinámico", funded by the Pontifical Catholic University of Valparaíso (www.pucv.cl).

REFERENCES

Bellifemine, F. et al. 1999. JADE - A FIPA Compliant Agent Framework. C-SELT Internal Technical Report.

Burrafato, P., and Cossentino, M. 2002. Designing a multiagent solution for a bookstore with the PASSI methodology. In 4th International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002).

Fischer, K. 1994. The design of an intelligent manufacturing system. Proc. of CKBS'94, University of Keele, England, pp. 83–99.

Lim, M. K. and Zhang, Z. 2002. Iterative multi-agent bidding and co-ordination based on genetic algorithm. Proceeding of 3 Complex Systems, and E-Businesses, Erfurt, pp. 682--689.

Maturana F, Shen Weiming, Norrie D. 1999. MetaMorph: an adaptive agent-based architecture for intelligent manufacturing. International Journal of Production Research, 37: 2159-2173,.

Parunak, H.V.D., Baker, A.D. And Clark, S.J. 1997. The AARIA agent architecture: an example of requirements-driven agent-based system design. Procs. 1st Int. Conf. on Autonomous Agents, Marina del Rey, CA.

PTK. 2007. The PASSI Toolkit (PTK) Available at http://sourceforge.net/projects/ptk. Accessed on 09/10/2005

Saad, A., Biswas, G., Kawamura, K., Johnson, M. E. Salama, A. 1995. Evaluation of contract net-based heterarchical scheduling for flexible manufacturing systems. Procs. Int. Joint Conf. on Artificial Intelligence, Workshop on Intelligent Manufacturing, Montreal, Canada, pp. 310-321.

Smith, R. G. and R. Davis. 1978. Distributed Problem Solving: The Contract Net Approach. Proceedings of the 2nd National Conference of the Canadian Society for Computational Studies of Intelligence.

Usher, J.M. 2002. Negotiation-based Routing in Job Shops via Collaborative Agents. World Manufacturing Congress, Distributed Intelligence in Technology, Economic and Social Applications: An International Symposium, Workshop on Agent-Based Process Planning and Scheduling, Rochester, New York.

Váncza, J. and Márkus, A. 2000. An agent model for incentive-based production scheduling. Computers in. Industry, Vol. 43, Issue 2,pp. 173-187.

Yoo, M. J., Müller, J. P. "Using Multi-agent System for Dynamic Job Shop Scheduling". 4th Int. Conf. on Enterprise Information System (ICEIS 2002), Ciudad Real, Spain, April 2002.