Intelligent Production Machines and Systems
D.T. Pham, E.E. Eldukhri and A.J. Soroka (Eds.)

Simulation of manufacturing processes using multi-agent technology

Jiří Vokřínek, Dušan Pavlíček, and Radek Šmerák

Gerstner Laboratory, Department of Cybernetics, Czech Technical University in Prague, Technická 2, CZ 166 27 Prague, Czech Republic

Abstract

Testing on real equipment is very expensive. Simulation of manufacturing processes helps find out best possible configuration of the production process even if the real equipment doesn't exist. Extension or reconfiguration of the existing processes can be tested and evaluated before investment takes place. Validation of the plans in the simulated environment can also bring significant benefits. Proposed simulation system makes use of advantages of the agent technologies such as high flexibility and modular architecture, openness and easy reconfiguration. Simulation system allows testing several configurations of production processes and locates bottlenecks or possible problems. Using different planning techniques with different configurations allows adapting the planning algorithms and finding the best plan for the appropriate configuration. All the experimental results can be analytically evaluated and used for improving production processes. On the other hand, real data gathered from the real processes shall be applied back on the simulation model to improve its accuracy.

1. Introduction

Modern industry processes challenge the needs of effective automation. Manufacturing has to be transferred into high flexibility and low-cost automated process. Agent technologies are well suited for the domain of simulation and planning of the production process. Agent technologies provide high flexibility and modular architecture. The multi-agent system is suitable for *i)* middle and high level planning of the production process on short or long-term period with respect to optimality criteria, *ii)* low-level real-time control and planning with production feedback, *iii)* configuration and simulation of the production process to find the most expedient structure and plan. The concept of agentification makes it possible to utilize the existing software and/or hardware solutions as well as human resources and robots. The agent can model (control) all the parts of the manufacturing process such as transport system, assembly robots, manual assembly stations, intelligent sensors, storage systems or databases.

2 Agent-based planning and simulation

A multi-agent approach is new and very efficient in the field of production planning and simulation. The system is implemented as a community of so called agents that communicate and co-operate with each other.

Contrary to a standard centralized system [1, 2], a multi-agent system supports easy integration of existing software and hardware [3, 4, 5, 6]. Each existing system that should be involved in the production process is "agentified" – equipped with an interface that allows it to behave as an agent. The multi-agent co-operation is easier if the agents comply with FIPA standards for the inter-operation of heterogeneous software agents [7, 8]. The multi-agent technology is scalable to a wide range of planning and simulation problems from small factories to large enterprises with hundreds of connected units.

It is obvious that the emulation provided by a multi-agent system can identify problems of the entire manufacturing process as well as it can help improve the performance of the process without influencing the real equipment that is actually not required for the purposes of testing. Modularity and easy reconfiguration of the multi-agent solution makes it possible to test several configurations and improve the efficiency of the process.

The proposed architecture is built on the multi-agent technology. The system is highly configurable and allows the user to model different configurations of the process. The simulation units (represented by the agents) contain particular parameters and constrains of the simulated units (e.g. production time, inputs, outputs, distribution of errors, etc.). The agent approach allows the integration of a new unit by simple means – the new agent is added. This makes it possible to change the configuration very easily, even in runtime. Each agent finds all its partners, which can provide him with the inputs or, more generally, with all agents that are able to provide inputs or to process its outputs. The configuration of the production process is self-organizing and it is defined by the parameters of the simulation units (agents). We recognize three main simulation units (agents): the processing unit, the stocks (buffers) and the transport units.

3 Time Synchronization

Time flow is one of the most complicated problems in simulation. Desired simulated time granularity usually goes to seconds or minutes, while the simulation period goes to weeks or months. The special *time synchronization agent* (TSA) is necessary. However, detailed simulation (e.g. with second or minute steps) of a long-time period can take a significant amount of real time. The synchronization agent has to be able to change the granularity of the simulated time in order to meet the best possible ratio between the required level of detail and the duration of the simulation.

The proposed system is based on *parallel discrete even simulation* paradigm. The problem is to keep the time flow in distributed agents environment synchronized and to guarantee proper causality of the events in the system. We propose the following time synchronization model for manufacturing simulation using multi-agent technology. The *time synchronization agent* is the centralizing point of the system and it is responsible for the time flow control in the entire system.

3.1 Confirmed Time Windows Synchronization

The time flow simulation consists of two synchronizing methods – *local synchronization* and *time windows* extension.

The local synchronization part is a non-central-oriented synchronization (it doesn't use TSA). It uses time stamps for each operation in the manufacturing process. Individual agents operate asynchronously and when a causal synchronizing event occurs (e.g. a low or a high amount of components in the buffers – input/output buffer under/overflow), they wait for the others. This method maximizes CPU utilization but running asynchronous agents may cause problems when the need of simulation interruption or parameter changes occurs (Figure 1-a).

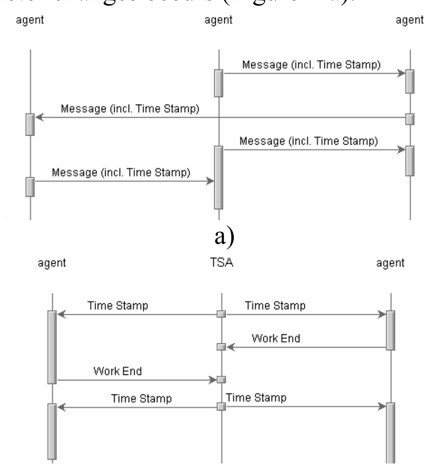


Figure 1: Confirmed time windows synchronization

The time windows extension of the previous method ensures the entire system to be synchronized at certain points of the simulation. Individual agents operate asynchronously using local synchronization within the time windows defined by the TSA. When an agent reaches the end of the time window, it sends a *confirmation* message to the TSA. TSA is responsible for collecting all the confirmation messages and subsequently it opens a new time window. This extension keeps the CPU load maximized. Due to the asynchronous nature of agents' operation taking place within the time windows, simulation can only be interrupted between two time windows (after collecting the confirmations and before opening of the new time window) (Figure 1-b).

The strong points of this approach are *fail-proof* operation (in terms of causality failure due to bad time-synchronization), possibility of *simulation interruption and continuation* between two time windows, very good *CPU utilization*, and the possibility of *time compression and dilatation* (one agent operates faster than the other, e.g. an idle machine consumes much less computation time than an operating machine). The weak points are *time non-uniformity* (each agent operates with different simulation time at the same moment) and *time non-linearity* (the speed of time flow is not constant).

In our case, the most important criterion is the fail-proof operation, the possibility of interruption and the CPU utilization. We accept message overhead of this method because confirmation messages can be extended by the information needed for other important components (see section 4.3).

4. System architecture

The architecture of the system can be divided into three functional parts:

i) simulation agents,
ii) time synchronization and control, and
iii) analytical and visualization part.

Simulation agents represent all the important production units. There is also a special set of agents for time simulation and control and for the visualization component. Together they form a manufacturing simulation multi-agent system.

The system is implemented in JAVA using the A-Globe agent platform [10]. This platform provides all the necessary support with minimum overheads and a very fast messaging system [9, 10].

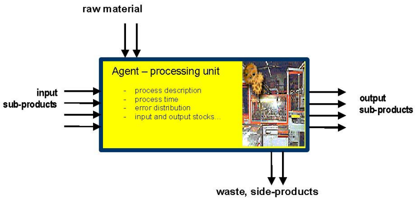


Figure 2: Agent as a general manufacturing unit.

4.1 Simulation agents

Simulation agents represent relevant manufacturing components. All the desired functionality is covered by an agent. We recognize three main classes of simulation agents – the processing unit, the stock, and the transport unit.

4.1.1. Agent structure

Each *manufacturing processing unit* is described by its main parameters (see Figure 2) such as inputs (raw material, sub-products), outputs (sub-products, waste), processing time, error distribution, size of input and output stocks, etc. The agent's behavior is to transform the inputs to the outputs upon the specified conditions.

Stocks (buffers) are a special case of processing units with reduced behavior. Stocks don't provide any transformation of the input sub-products to the output sub-products. They contain only parameters defining the size of the stock. The output product is the same as the input product.

Transport units are available for modeling time delays needed for transporting the sub-products (conveyor belts, etc.). The parameters of transport units are mainly time and the capacity. Once again, transport units can be modeled using reduced simulation agents.

Each simulation agent is equipped with a description of its desired role in the system. It is defined in the XML configuration file. The parameters of the agent are

- a set of input sub-products/raw materials
- a set of output sub-products/waste
- a set of production recipes
- input/output buffers capacities and critical limits
- failure conditions
- position of the machinery represented by the agent in the simulated world

Each *recipe* is defined by input components (with possible supplements), output components (with the error distribution of spoilage), and the processing time. During the simulation, an agent tries to keep its *input buffers* above critical limit and its *output buffers* below critical limit. When the agent's input buffers contain enough sub-products to start a certain receipt and the output buffers are able to accommodate the produced sub-products, the receipt can be executed. If more receipts can be executed at the same time, one of them is chosen randomly. The process of receipt execution is periodical as the simulation runs and any receipt can be carried out. When the *failure condition* is reached, the agent stops execution of the receipt and waits for the 'machine repairing'.

4.1.2. Contracting and Negotiation

Each transport of material or negotiation about sub-products availability is provided by inter-agent communication. The main goal of the agent's *con-*

tract module is to arrange the flow of material among agents. The *Contract-net-protocol* [11, 16] is used for contracting the producers and consumers of the agent's sub-products. The contracting behavior is triggered whenever the input or output buffer volume reaches its critical limit. After successful contracting, the agreed amount of sub-products is virtually transferred from the output buffer of the producer to the input buffer of the consumer. The duration of this transfer depends on the distance between agents in the simulated world.

Another negotiation protocol used is *query-inform* [12]. It is used for discovering the sub-products availability, respectively the time when a particular sub-product will be available in the producer's output buffer.

Every message contains a *time stamp* for time synchronization purposes. It is important for maintaining causality in the asynchronous time-flow environment. The contract module is aware of the last time stamp known to every co-operating agent and the future availability of the sub-products.

4.2 Time synchronization and control

Time synchronization of all simulation agents is provided by *confirmed time windows synchronization* method (see chapter 3.1).

Every agent is connected to the *time synchronization agent* (TSA). There is only one instance of this agent in the system. TSA is responsible for opening and closing the time windows. In fact, the TSA sends only a message responsible for opening a new time window. Information of the time window closing possibility is gathered from GEA (see bellow). This information is subscribed and when TSA receives confirmation of finishing all the activities within the current time window it then opens a new time window by broadcasting the opening message for the new time window (this message contains time stamps of the beginning and the end of this window). Just before this message is sent, it is the only moment when the simulation can be interrupted and TSA checks for requests for interruption provided by *simulation control module* (see below) before opening new time window.

The *Global Environment Agent (GEA)* is used for collecting all the distributed data of the entire system. It is responsible for manufacturing process visualization, simulation control, synchronization and visualization (see chapter 4.3).

Simulation Control Module (SCM) allows the user to control parameters of the simulation, such as to start, to pause and to stop the simulation, to adjust the time windows duration and to set the simulation starting and finishing time (see Figure 3). It also shows the current simulation time of the Logical Visio (see bellow). The time window settings consist of two parameters – *window size* (simulation time within one time window, e.g. 60s) and *window duration* (the corresponding minimal 'real' time, e.g. 3s). A combination of these two parameters defines the *speed factor* of the simulation in relation to the real time (simulation runs 20 times faster in this case). Once the user interrupts the simulation (by pressing the pause button) the time window parameters can be modified.

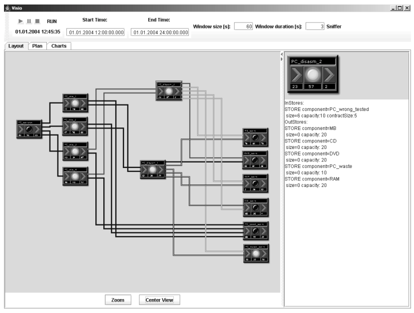


Figure 3: Logical Visio

4.3. Visualization and data evaluation

The simulation environment is presented to the user in a single compact application window. The time synchronization agent control is integrated in its bottom part (see Figure 3). It allows the user to control the necessary parameters of the simulation. All simulation units are clearly presented on the screen together with the inter-agent communication links and material flow. The simulation system presents internal data to the user in three different ways:

- schematic logical visualization,
- analytical data visualization,
- 3D simulation visualization.

Besides the 'on-screen presenting' features, all data can be stored in an external file for future analyses. All the visualization components are connected to the Global Environment Agent.

4.3.1 Logical Visio

The main screen of the simulation system contains three visualization tabs (see Figure 3). The first one ('Layout') belongs to the Visio that shows the logical structure of the simulation process. In the top-most part of the screen, there are simulation control components that are always accessible. The main part of the Visio (left) shows the logical structure of the system. On the right side of the screen the details of the selected agent are shown. Each agent is presented as a box with its name, the input/output buffer relative loads, machine load and the agent's current state (working, idle or broken). A more detailed description of an agent can be seen in the right-hand part of the screen. Agents are connected by lines that represent connections between machines as well as the communication or production flow (highlighted by different colors).

4.3.2. Analytical Visio

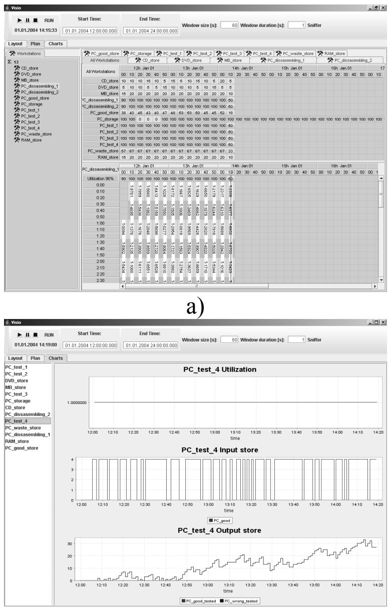
The other two tabs ('Plan' and 'Charts') belong to analytical visualization.

The *plan part* shows the processing activity of the simulation agents. It is able to show the load of the agents during the simulation in the time-table charts (see Figure 4-a). This Visio was adapted from the multi-agent planning system ExPlanTech developed by Gerstner Laboratory [5] and it is able to show not only the part of the production that has already been finished, but also the future plans (this can be used once the system is extended with the planning module). It is possible to show the overall time-table or inspect the individual, more detailed agent's time-tables.

The *charts part* shows several charts of the agents' activity and state (see Figure 4-b). For each individual agent, there are charts that show utilization of input/output buffers and machine load. Every chart shows the minimal, the maximal and the end value for each time window.

4.3.3. 3D Visio

The game-like visualization is provided for an in-depth inspection of the material flow and machines' activity in the manufacturing process (see Figure 5). It provides the user with intuitive overall information about the system configuration, together with detailed information about individual agents and sub-products. This component is based on Crystal Space game engine [14] and it was previously successfully deployed for several multi-agent systems [13].



a)

b)

Figure 4: Analytical Visio.

4.3.4. Log Manager

The log manager collects all the information in the system within each time window. When all the information that belongs to the open time window is collected, the GEA is informed about the possibility of closing the current time window. All the information is stored and it is available for visualization components (i.e. immediately for the Analytical Visio and at the end of each time window for the others). The complete log is saved to a file for future use and further analyses.

5 Conclusions and future work

Agent technologies provide high flexibility and modular architecture with possibility of distributed processing. Simulation provides us with the possibility to test several configurations of production processes and to find out bottlenecks or possible problems. The system provides very fast and reliable experimental environment that makes it possible to choose the most expedient structure from the set of tested configurations. The agents running in the simulation define the structure of the process. The structure can be simply changed by adding or removing individual agents.

The natural features of multi-agent system such as openness guarantee easy extension in the future when new requirements on the simulation system rises. For example, when new specific machine have to be simulated, the special agent should be developed. This agent will be able to cover all the unforeseen parameters and constrains and it is simply added to the existing configuration without any influence to the rest of the system.

The system can be easily integrated with the existing planning system or with newly developed planning modules. Using different planning techniques with different configurations allows adapting the planning algorithms and finding the best plan for the appropriate configuration. We plan to develop set of planning modules for well-known planning strategies in the future.

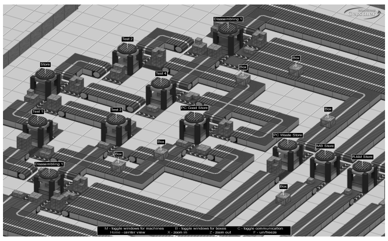


Figure 5: 3D Visio. The game-like visualization

All the experimental results can be analytically evaluated and used for improving production processes. On the other hand, real data gathered from the real processes shall be applied back on the simulation model for improving its accuracy.

Proposed simulation system can help find out the best possible configuration of the production process even if the real equipment does not exist yet. Extension or reconfiguration of the existing processes can be tested and evaluated before investment take place. Validation of the plans in the simulated environment can also bring significant benefits.

6 Acknowledgements

This research has been supported by CONCERN (CONex Central European Electr(on)ics Recycling Network of Excellence), I*PROMS Network of Excellence and the Ministry of Education, Youth and Sports of the Czech Republic grant No. MSM 6840770013.

References

[1] Sadeh, N.,, Yoichiro N..: Focused Simulated Annealing Search: An Application to Job Shop Scheduling. Annals of Operations Research, 60, pp. 77-103, 1996

[2] Sadeh, N., Fox, M.S.: Variable and Value Ordering Heuristics for the Job Shop Constraint Satisfaction Problem. Artificial Intelligence 86, pp.1-41, 1996

[3] Sadeh, N., Hildum, D., Kjenstad, D., and Tseng, A.: MASCOT: An Agent-Based Architecture for Dynamic Supply Chain Creation and Coordination, Production Planning & Control. Volume 12, No.3, 2001

[4] Mařík V., Pěchouček M., Hazdra T., Štěpánková O.: ProPlanT -Multi-Agent System for Production Planning. In: Proc. of XVth European Meeting on Cybernetics and Systems Research, Vienna 1998, pp. 725-730

[5] Pěchouček M, Vokřínek J, Hodík J, et al.: ExPlanTech: Multi-agent framework for production planning, simulation and supply chain management, LECTURE NOTES IN COMPUTER SCIENCE 3187: 287-300 2004

[6] Mařík, V., Vrba, P., Fletcher, M.: Agent-Based Simulation: Mast Case Study. In Emerging Solutions for Future Manufacturing Systems. New York: Springer, 2004, p. 61-72. ISBN 0-387-22828-4.

[7] Mařík, V., Pěchouček, M., Vrba, P., Hrdonka, V.: FIPA Standards and Holonic Manufacturing. In Agent-Based Manufacturing. Berlin: Springer, 2003, p. 89-119. ISBN 3-540-44069-0.

[8] FIPA - Foundation for Intelligent Physical Agents. http://www.fipa.org.

[9] Vrba, P.: JAVA-Based Agent platforms evaluation, Holonic and Multi-Agent Systems for Manufacturing, LNAI 2744, Springer-Verlag Berlin Heidelberg, 2003, pp. 47-58

[10] Šišlák, D., Rollo, M., Pěchouček, M.: A-Globe: Agent Platform with Inaccessibility and Mobility Support. In Cooperative Information Agents VIII. Berlin: Springer, 2004, p. 199-214. ISBN 3-540-23170-6.

[11] Smith, R.: The contract net protocol: High-level communication and control in a distributed problem solver. IEEE Transactions on Computers, 1980, C-29(12):1104-1113.

[12] FIPA Query Interaction Protocol Specification: FIPA00027. Foundation for intelligent physical agents http://www.fipa.org/specs/fipa00027/XC00027E.pdf

[13] Šišlák, D., Pěchouček, M., Rollo, M., Rehák, M., Tožička, J.: Simulating Agents´ Mobility and Inaccessibility with A-Globe Multi-agent System. In Multiagent System Technologies. Berlin: Springer, 2004, s. 258-272. ISBN 3-540-23222-2.

[14] SpaceCrystal - free and portable 3D Game Development Kit , http://crystal.sourceforge.net