A Novel Time Advancing Mechanism for Agent-Oriented Supply Chain Simulation

Xu Xu, Jie Lin
School of Economics and Management, Laboratory of Intelligence Decision and Decision Support
Tongji University, Shanghai, P. R. China
Email: 08xuxu@tongji.edu.cn, jielinfd@263.net

Abstract— **Among the techniques supporting a multi-decision context, as a supply chain is, distribution simulation can undoubtedly play an important role in a co-operative environment. The distribution simulation for supply chain has its advantages to find and solve bottleneck of supply chain. Considering the design of distribution simulation platform for supply chain, the realization of time synchronization and time advance will be the key points. This paper proposes an advancing mechanism that integrates High Level Architecture with multi-agent distributed simulation to meet time management in supply chain simulation, i.e., a 'heterogeneous' system is built during our research. We present some preliminary experimental results which illustrate the performance of our advancing mechanism on our platform established. Finally, the experimental results have demonstrated the novel time advancing mechanism can be successfully applied in supply chain contexts, and surely prove that it is feasible and scientific.**

Index Terms— **distributed simulation, supply chain, high level architecture, agent, time management**

I. INTRODUCTION

Modern industrial enterprises operate in a rapidly changing world, stressed by even more global competition, managing unforeseeable markets, supervising geographically distributed production plants, striving for the provision of outstanding products and high quality customer service [1].

In the last years, globally companies, as well as Small and Medium Enterprises (SMEs), are realizing that the efficiency of their own business is heavily dependent on the collaboration and co-ordination with their suppliers as well as with their customers [2]. This external perspective is termed in literature under the concept of supply chain management (SCM), which is concerned with the strategic approach of dealing with trans-corporate logistics planning and operation on an integrated basis [3].

From the IT perspective, a new wave of solutions is arising with the main type to overcome all the physical, organizational and informational hurdles which can seriously jeopardize any co-operation effort [4]. Advanced Planning and scheduling (APS) systems aim to step over the intra-company integration supplied by Enterprise Resource Planning (ERP) systems by providing a common inter-organizational SCM platform, which supports the logistics chain along the whole product life-cycle, From its initial forecast data, to its planning and scheduling, and finally to its transportation and distribution to the end customer [5]. Despite the various solutions currently available on the market, the common feature of the APS products reside on the intensive usage of quantitative methods in order to provide users with the best solution at time.

Among these quantitative methods, simulation is undoubtedly one of the most powerful technique to apply as a decision support tool, within a supply chain environment.

In the industrial area, simulation has been mainly used for decades as an important support for production engineers in validating new lay-out choices and correct sizing of a production plant. Nowadays, simulation knowledge is considered one of the most important competition to acquire and develop with modern enterprises in different processes (marketing, manufacturing) [6]. In particular, supply chain is a typical environment where simulation can be considered a useful device.

The simulation of physical systems (e.g. supply chain) is an important tool for researchers that allows them to analyze the behavior or/and performance of the system considered and to verify new ideas. It is natural to model supply chain as a set of computing processes which then can be handled by distributed machines or processors. For the last two decades, distributed simulation has been an active research area. Distributed simulations not only reduce the computation time and permit to execute large programs which cannot be executed on a single machine, but they first of all reflect better the structure of the physical system to be simulated.

There has been considerable recent interest in agent-based simulation, simulation based on autonomous software and/or hardware components (agents) which cooperate within an environment to perform some task. An agent can be viewed as a self-contained, concurrently executing thread of control that encapsulates some state and communicates with its environment and possibly other agents via some sort of message passing [7]. The environment of an agent is that part of the world or computational system 'inhabited' by the agent, the environment may contain

---

Xu Xu and Jie Lin are co-first authors.

The research is supported by the National High-Tech. R&D Program for CIMS, China (No.2007AA04Z151), Program for New Century Excellent Talents in University, China (No.NCET-06-0377), Shanghai Leading Academic Discipline Project (No.B310) and the National Natural Science Foundation, China (No.70531020).

doi:10.4304/jcp.4.12.1301-1308

other agents whose environment of a given agent.

Simulation has traditionally played an important role in agent research and a wide range of test-beds have been developed to support the design and analysis of agent architectures and systems. There has been considerable recent interest in agent-based platform which cooperate within an environment to perform some tasks, such as Swarm, Repast, Ascape. By far, there has no mature distributed simulation platform not only to satisfy the needs of supply chain, but also to solve the bottleneck of supply chain. The work suffers the following problem. The problem is how to integrate distributed simulation with multi-agent to build the platform. Such combination is an essential step in establishing an simulation platform, the behavior and environment in which they are all embedded. Taking all into consideration is how to establish an advancing mechanism to satisfy time synchronization and time advancing. These are our paper's key points.

In summary, aim of this paper is to find and realize time advancing mechanism for agent-oriented supply chain Simulation.

The paper is organized as follows. First, background and related work are illustrated. Second, time management research is summarized. Third, time advancing mechanism is proposed in order to build agent-oriented supply chain simulation. Fourth, preliminary experimental results are presented and reported. Finally, conclusion and future work from the authors are provided.

II. BACKGROUND AND RELATED WORK

In this section, we briefly describe the relation between distributed simulation and SCM, traditional distributed simulation protocols, and multi-agent distributed simulation.

A. The Relation between Distributed Simulation and SCM

Generally, simulation of supply chain can be carried out according to two structural paradigms: using only one simulation model, executed over a single computer (local simulation) or implementing more models, executed over more computers in a distributed fashion.

Distributed simulation: Distributed simulation is concerned with execution of simulations on geographically distributed computers interconnected via a network, local or wide [8]. The need of a distributed execution of a simulation across multiple computers derives from four main reasons [9], [10]: (1) produce execution simulation time, (2) reproduce a system geographic distribution, (3) integrate different simulation models that already exist and integrate different simulation tools and languages, and (4) increase tolerance to simulation failures.

SCM: SCM involves managing the flow of material and information through multiple stages of manufacturing, transportation and distribution with the objective of maintaining low inventories without compromising customer service level. The effective practice of SCM is critical to participating companies especially in today's business trend whereby companies are geographically distributed throughout the globe.

Traditionally, SCM involves only a single enterprise with multiple facilities and distribution centers. But in recent years, the scope of SCM has evolved to cross the enterprise boundaries, as vertical integration is no longer the emphasis of large corporations [11].

Commercial simulation tools for supply chain planning have been released in recent years, for example the Supply chain Analyzer by IBM and the integrated tools of simulation and optimization by i2. This illustrates the importance and applicability of simulation to supply chain planning. The main area of application of supply chain simulation has been on performing what-if analysis, by varying various aspects of the chain. Building a detailed model of the supply chain does not pose a problem when the chain involves only a single enterprise. In contrast, not many participating companies are willing to share detailed model information when the chain crosses the enterprise boundaries. This obstructs the use of simulation in supply chain planning.

Distributed simulation techniques as the enabling technology to eliminate this obstacle. Distributed simulation technology allows each participating corporation to run their own simulation model at their own site. Detailed model (application codes and data) information is encapsulated within the corporation itself and the participating corporations only need to define essential information flows from one model to another. i.e., some of the features of distributed simulation were recognized as important benefits for enabling sound simulation models in support of SCM. The distributed approach has been progressively considered the most viable. This is due to its undeniable advantages [12], [13]: (1) the possibility of developing complex models preserving proprietary information associated with individual systems, (2) the correspondence between model and node, guaranteeing a real and updated representation of the single industrial units, and (3) a reduction of the simulation time, taking advantage of the additional computing power of the distributed processors.

In the specific field of supply chain simulation, traditionally carried out with the use of local models. To realize the model encapsulation, traditional distributed simulation protocols should be discussed.

B. Traditional Distributed Simulation Protocols

Distributed simulation has been largely used for many years in military applications, in which computers and executables have been joined together through tools such as distributed interactive simulation (DIS) protocol, aggregate level simulation protocol (ALSP), and the high level architecture (HLA). These architectures, all developed in the military field, are the basis of distributed simulation. Weaterly et al. [14] and Page and Smith [15] provide a complete description of them; they are briefly illustrated in the next paragraphs.

The Distributed Interactive Simulation Protocol: The Distributed Interactive Simulation (DIS) protocol became an IEEE standard in 1993 and its objective was to create synthetic, virtual representations of warfare environments through a systematic connecting of separate simulation subcomponents which reside in distributed and multiple locations. Generally, DIS simulations run in real-time with an elevated degree of detail. Communication in a DIS environment is based on the protocol date unit (PDU), a set of encoded bits that communicate entity state and other information identified as useful within the protocol (weapons fire events, etc.). A process known as 'dead reckoning' (esteemed position) is used to reduce the number of entities introduced into the network during runtime, by allowing entity state extrapolation between updates.

The Aggregate Level Simulation Protocol: The Aggregate Level Simulation Protocol (ALSP) was targeted towards support for the inter-operation of aggregate-level, logical-time simulations. ALSP provides an ASCII-based message-passing protocol and software infrastructure that coordinates the advance of simulation time, enforces adherence to a common object model of the shared simulation state, and arbitrates contests over the right to modify that shared state.

The High Level Architecture: High Level Architecture (HLA) [9] is the most known distributed framework. It represents both a generalisation and extension of DIS and ALSP [9]. The HLA is defined by three components: (1) a common model definition and specification formalism, (2) a collection of services describing the HLA runtime environment, and (3) a set of rules governing compliance with the architecture. HLA is designed with a high degree of flexibility, permitting an arbitrary mixture of fidelity and resolution. The notion of federation is at the heart of HLA, that is, a collection of federates-simulators and other systems-that inter-operate using the protocols described by the architecture. In a typical federation execution, a federate joins the federation, indicates its operating parameters (information that federates provide to the federation and information that they expect to receive from that federation) and then participates in the evolution of the federation state until the federate departs the federation, or the simulation ends [16]. In 1996 the HLA was endorsed as the standard for all US DoD M&S. Today it still represents the most widely used architecture for distributed simulation. Many scientific papers analyse perspectives and the uses of HLA in different applications (navy, transportation, environment, videogames, etc.) [17], [18], included the industrial production field [19], [20].

In HLA, each individual model is a federate. A collection of federates that form the whole simulation system is a federation. To apply this to a supply chain simulation, the federate is thus the basic simulation model of each individual company. Each company defines only data that they are willing to share in the Simulation Object Model (SOM) using the object model template (OMT) of the HLA. The simulation time synchronization of federates is achieved automatically through the time management services of HLA. Time management will be discussed in the next section.

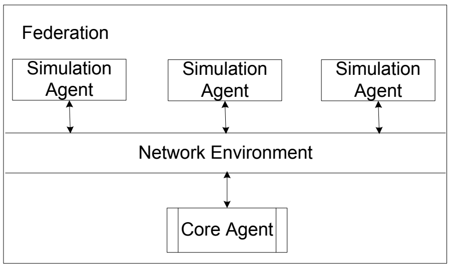


Figure 1. Multi-agent distribution model.

C. Multi-Agent Distributed Simulation

Agent-based simulation offers advantages when independently developed components must inter-operate in a heterogeneous environment, e.g., the Internet, and agent-based simulation are increasingly being applied in a wide range of areas including business process modeling and military simulations. A multi-agent simulation is an simulation composed of multiple interacting intelligent agents. Multi-agent simulation can be used to solve problems which are difficult or impossible for an individual agent or monolithic simulation to solve.

What is required is a general, distributed simulation framework for multi-agent. Such a framework, capable of supporting a wide variety of agents and environment, would facilitate generalization by ensuring that different implementations are subject to identical assumptions. In addition, the use of distributed simulation techniques would allow exploiting the processing power to study larger and more complex multi-agent.

The aim of the study is to simulate a wide range of agent-based supply chain, from a single agent in a complex environment, e.g., an agent controlling a production process, to many agents in s simple environment, e.g., an environment consisting almost entirely of other agents such as transportation agent, buffer agent in network environment.

Figure 1 shows the simplified multi-agent distribution model. Supply chain simulation can be integrated through distributed simulation once each module is implemented.

III. TIME MANAGEMENT RESEARCH

A. Time Models

Time models are inspired by research in the distributed simulation community, where they are used implicitly to assign logical time stamps to all events occurring in the simulation [21]. In software simulations, the logical time stamp of an event corresponds to the physical time the event was observed in the real world which is being simulated.

A time model captures the execution requirements in terms of logical time, according to the semantic properties of the multi-agent application. More precisely, a time model defines how the duration of various activities in a multi-agent is related to logical time [22], and these logical durations of activities are used as a means to determine the relative execution order of all activities within a multi-agent platform. In this way a time model allows the developer to describe the required execution in a platform-independent way. The execution of an application on a particular execution platform must be controlled according to the defined time model.

B. Time Management Mechanism

By specifying time models, it is important to define execution requirements for a multi-agent platform. By relating multi-agent activities to logical time. Multi-agent simulation additionally need time management mechanisms to ensure that all activities are executed according to the time model specification. These mechanisms avoid that any event with a logical time in the future can have influence on things with a logical time in the past, even in the presence of arbitrary network delays or computer loads. In other words, time management preserves causality dictated by logical time.

Distributed simulation communities have been investigating the consistency of logical time in simulations for a long time. All events happening are ordered and hence causally related by means of the global notion of logical time [23]. Therefore various time management mechanisms have been developed to prevent causality errors:

Execution directed by clock. In this approach the logical time of the system is discredited in a number of intervals of equal size. The interval size is called time step. Global synchronization schemes force all entities to advance together in a lock-step mode, and hence the execution of the system proceeds synchronously. In the case of multi-agent simulation, a drawback is that synchronous execution forces all agents to act at the pace of the slowest one, which severely limits execution speed [24]. Moreover, since a central authority must control and keep track of the execution of all agents in the system, the cost of synchronous approaches increases rapidly as the number of agents grows.

Execution directed by events. In this case, events are generated by all entities [25], and each event has a precise logical time stamp which allows sorting them. During execution, the next event to be processed is the one with the smallest logical time stamp, ensuring causality and thereby skipping periods of inactivity. However in a distributed context, a system is modeled as a group of communicating entities, referred to as logical processes (or LPs).

Each LP contains its own logical clock (indicating its local logical time) and all LPs process events asynchronously and advance at different rates, which allow a significant speedup, but may cause causality errors. Hence, for asynchronous execution additional synchronization is needed to ensure that each LP processes messages in increasing logical time order:

Conservative synchronization. In conservative synchronization [26] each LP only processes events when it can guarantee that no causality errors (out of (logical) time order messages) will occur. This causes some LPs to block, possibly leading to deadlock. The performance of conservative synchronization techniques relies heavily on the concept of lookahead, but the autonomous behavior of agents could severely restrict the ability to predict events [27]. Moreover, to determine whether it is safe for an agent to process an event, information about all other agents must be taken into account, limiting the scalability of this approach.

Optimistic synchronization. In optimistic approaches, causality errors are allowed, but some roll-back mechanism to recover from causality violations is defined (e.g. time warp [28]). While this approach is feasible for simulations, providing rollback for multi-agent simulation applications in general is not feasible at all. Moreover, the cost imposed by the roll-back mechanisms can easily outweigh the benefits [29] and increases rapidly as the number of agents grows.

C. Time Stamp Order

The features of time stamp order can be summarized: (1) a message will be held until run-time infrastructure (RTI) can guarantee that no message having a smaller time stamp will later be received, (2) no message will be delivered to a federate in its past, (3) it is useful for classical discrete event simulations, and (4) conflict resolution (ordering of concurrent messages) is deterministic.

D. Lookahead

For Time Stamp Order using conservative synchronization, a federate promises the RTI that it will predict attribute updates and interactions at least L time units ahead of time, RTI can use this value to determine when it can safely allow a federate to advance its time, and a federate may retract a event.

Lookahead is often derived from: (1) physical limitations of federates, (2) tolerances to temporal inaccuracies, (3) Time step increment, (4) non-preemptive behaviour, and (5) pre-computed simulation activities.

The pre-requisite to using the time management services of HLA is that each federate must define a nonzero lookahead value. In HLA, lookahead is associated with a federate. It is a value which determines the next earliest time that the federate will send an external event. Before a federate executes each event, the simulation time of this event has to be checked against the time granted by the RTI earlier. If the simulation time is not greater than that previously granted, the event is then simulated. Otherwise, the federate needs to request for time advancement from the RTI.

Lower Bound on Time Stamp (LBTS). Earliest time the RTI expects it could possibly deliver a message to a particular federate, $LBTS(F) = Min(T_i + L_i)$ over all federates i can send F a time stamp ordered message, and

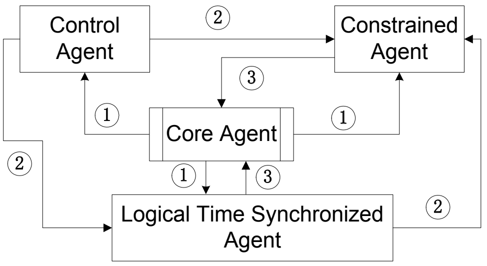


Figure 2. The procedure of time advancing.

when LBTS(F) exceeds the time advance requested by F, the RTI can grant F its time advance.

IV. PROPOSED TIME ADVANCING MECHANISM

Taking into consideration all of the above, this paper adopts conservative synchronization to study the strategy of time management mechanism in agent-oriented supply chain simulation. The time management of federate can be classified into three types:

Time regulating or logical time aggressive. Federate participates in determining the logical time of other federates (i.e., may send time stamp ordered messages).

Time constrained or logical time passive. Federate is constrained by the logical time of other federates (i.e., may receive time stamp ordered messages).

Logical time synchronized. Federate not only affects some federates, but also is controlled by other federates (i.e., may send and receive time stamp ordered messages).

A. Time management of Agent-Oriented Simulation

Time management of federate is as discussed above, the time management of federate agent has the same classification. In other words, time regulating agent and logical time synchronized agent can send receive order (RO) message and time stamp order (TSO) message to time constrained agent and logical time synchronized agent, and they only send the RO message to regulating agent. Time constrained agent only sends RO message. $LBTS_i$ of Federate agent can be defined as follows:

$$LBTS_i = (Min(T_j + lookahead_j) \cdot \alpha_i) + M \cdot (1 - \alpha_i) \tag{1}$$

When the agent is time restrained agent, α_i=1, the agent only send the TSO messages; when the agent is non-time restrained agent, α_i=0, M is infinite.

To the core agent, LBTS can be defined as follows:

$$LBTS = Min(LBTS_i) \tag{2}$$

B. Time Advancing Mechanism for Agent-Oriented Simulation

In this section, we first present a framework (see Figure 2) of time advancing, then explain the proposed time advancing mechanism according to the framework in detail.

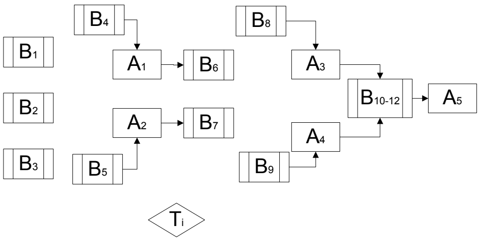


Figure 3. Experimental model.

The procedure can be summarized as follows[1] : (1) core agent receives the message of the completion of federate agents, then confirms federation LBTS, and sends federation time to the agents. After that, the simulation enters a new synchronization, (2) the second procedure of time advancing starts to discuss from different cases:

- Regulating agent and logical time synchronized agent receive federation LBTS, then compares their LBTS to federate LBTS, if LBTS is larger than federation LBTS, reply the core agent through their LBTS, the synchronization is completed. Otherwise, the regulating agent and logical time synchronized agent send the messages to other agent, after the completion of all tasks, they send TSO messages to the related agent (processing task included).
- Time restrained agent (including the constrained and logical time synchronized agent) receives federation LBTS, then compares his own LBTS, if LBTS is larger than federation LBTS, the core agent will receive the reply of time restrained agent, the synchronization is completed, and the time restrained agent will wait for the next federation LBTS. Otherwise, the time restrained agent completes all the tasks, and check the last circle Lookahead of related regulating agent LBTS (logic time is less than federate LBTS) whether having new Lookahead or not, the time restrained agent waits for the updating, after updating, the restrained agent gets his own LBTS (see equation (1)), then sends the messages to core agent.

And (3) core agent receives all the reply of time constrained agent and logical time synchronized agent, then computes the new federation LBTS (see equation (2)), and sends the new LBTS to all agents. After that, the simulation enters a new circle.

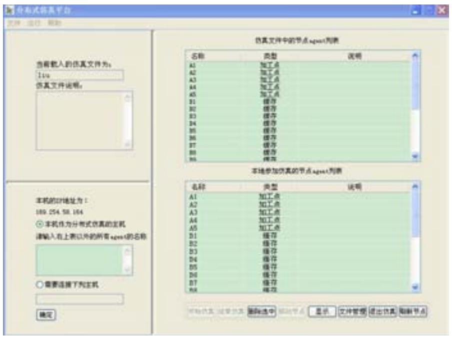
V. EXPERIMENTAL RESULTS

To evaluate the correctness and feasibleness of our proposed time advancing mechanism, we employ JADE in Eclipse platform to build a system to simulate the supply chain based on our novel time advancing mechanism. In order to illustrate the mechanism effectively, we establish an experimental model of supply chain (see Figure 3).

In Figure 3, there are five machining workshops (A_1, A_2, A_3, A_4, A_5), twelve warehouses or buffers (B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8, B_9, B_{10}, B_{11}, B_{12}), and five

---

[1] We illustrate the advancing mechanism according to ordinal numeration in Figure 2.



名称	类型	说明
A1	加工点	
A2	加工点	
A3	加工点	
A4	加工点	
A5	加工点	
B1	缓存	
B2	缓存	
B3	缓存	
B4	缓存	
B5	缓存	
B6	缓存	
B7	缓存	
B8	缓存	

名称	类型	说明
A1	加工点	
A2	加工点	
A3	加工点	
A4	加工点	
A5	加工点	
B1	缓存	
B2	缓存	
B3	缓存	
B4	缓存	
B5	缓存	
B6	缓存	
B7	缓存	

Figure 4. Experimental model in simulation system.

TABLE I.
PARTIAL SIMULATION RESULTS(OPERATION)

Lookahead	Operation State	Logic Time
...	...	...
130	waiting	120
140	waiting	130
140	waiting	133
140	waiting	135
155	exchanging	140
...	...	...
455	Producing	400
455	Producing	403
455	Producing	405
455	Producing	410
455	Producing	420
...	...	...

transportation trucks (T_1, T_2, T_3, T_4, T_5). We first load this model into the simulation system (see Figure 4), then check the simulation results of this model (partial simulation results are shown in Table I, II and Figure 5, 6[2]. Figure 5 and 6 show the bottleneck of this production process completely.

We can see that logic time advancing is complete consistency when the statuses of all nodes change, and satisfies the requirements of production process in supply chain. So the proposed time advancing mechanism is feasible and correct.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have introduced a novel time advancing mechanism that integrates HLA with multi-agent distributed simulation to meet time management in supply chain simulation context. The experimental results show our approach is feasible and scientific.

The main novel points and merits of the proposed method are in threefold: First, the proposed time advancing mechanism can apply in supply chain simulation,and help to solve the bottleneck of supply chain. Second, the proposed time advancing mechanism can apply in distribution simulation, and benefit for enabling sound simulation models in support of SCM (across enterprises). Third, the proposed time advancing mechanism can apply in agent-oriented simulation, and assist agent-based simulation in a heterogeneous environment.

TABLE II.
PARTIAL SIMULATION RESULTS(BUFFER AND TRANSPORTATION)

Buffer		Transportation	
Volume	Logic Time	Location	Logic Time
...	...	...	...
100000	120	B_2	120
100000	130	B_2	130
100000	133	B_4	133
100000	135	B_4	135
100000	140	B_4	140
...	...	...	...
99810	400	B_2	400
99810	403	B_5	403
99660	405	B_5	405
99660	410	B_5	410
99660	420	B_5	420
...	...	...	...

Future work needs to consider the efficiency of our proposed time advancing mechanism. Furthermore additional work is necessary to verify our novel time advancing mechanism in more complicated system.

ACKNOWLEDGMENT

Authors thank Chenghong Gao for his assistance with this work, and the reviewers for their detailed comments.

REFERENCES

[1] S. Terzi and S. Cavalieri, "Simulation in the Supply Chain Context: A Survery," *Computers in Industry*, vol. 53, no. 1, pp. 3–16, January 2004.

[2] R. Hieber, *Supply Chain Management: A Collaborative Performance Measurement Approach*, Zurich, VDF, 2002.

[3] H. C. W. Lau and W. B. Lee, "On a Responsive Supply Chain Information System," *International Journal of Physical Distribution and Logistics Management*, vol. 30, no. 7, pp. 598–610, July 2000.

[4] D. Schunk, "Using Simulation to Analyze Supply Chain," in *Proceedings of the 32nd Winter Simulation Conference (WSC)*, pp. 1095–1100, December 2000.

[5] H. Stadtler and C. Kilger, *Supply Chain Management and Advanced Planning*, Springer-Verlag, Berlin, 2000.

[6] J. Kosturiak and M. Gregor, "Simulation in Production System Life Cycle," *Computers in Industry*, vol. 38, no. 7, pp. 159–172, March 1999.

[7] M. Wooldridge and N. R. Jennings, "Intelligent Agents: Theory and Practice," *Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, June 1995.

[8] R. M. Fujimoto, "Parallel and Distributed Simulation," in *Proceedings of the 27th Winter Simulation Conference (WSC)*, pp. 118–125, December 1995.

[9] "High Level Architecture (HLA) Overview," Defense Modeling and Simulation Office, 2002.

[10] "RTI Programmer's Guide," Defense Modeling and Simulation Office, 1999.

[11] G. Archibald, N. Karabakal and P. Karlsson, "Supply Chain vs. Supply Chain: Using Simulation to Compete Beyond the Four Walls," in *Proceedings of the 31st Winter Simulation Conference (WSC)*, pp. 1207–1214, December 1999.

---

[2] Here, we set an example for B_4, B_6, B_8, B_{12}, A_1 and A_3 to show the simulation results (the warehouses or buffers and machining workshops above existing correlation in production process). The horizontal axis denotes logic time, and the vertical axis denotes the volume of material.

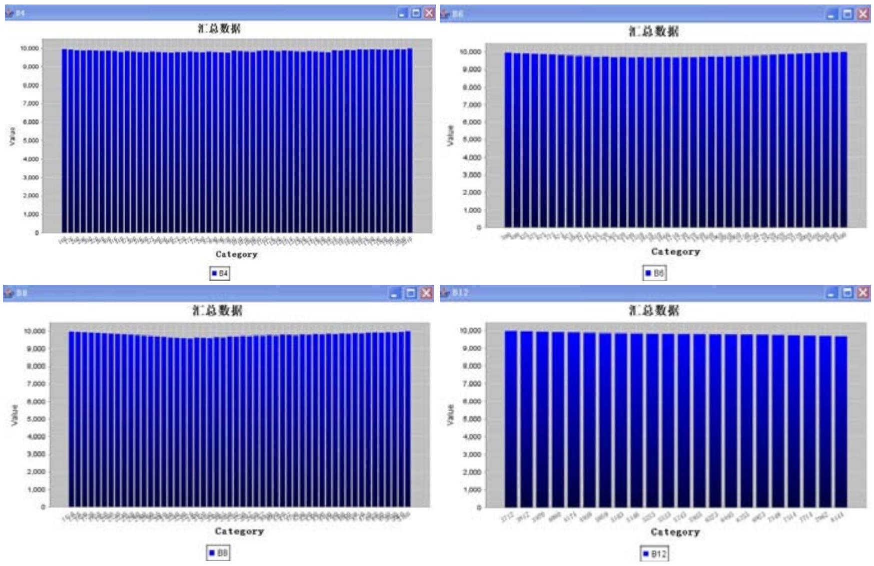


Figure 5. Partial simulation results: B_4-B_6 and B_8-B_{12}.

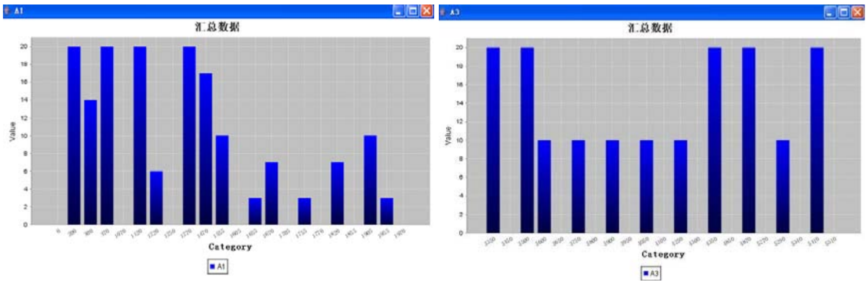


Figure 6. Partial simulation results: A_1 and A_3.

[12] C. Chandra and N. Chilov, "Simulation Modeling for Information Management in a Supply Chain," in *Proceedings of the 12th Annual Conference of the Production and Operations Management Society (POMS)*, March 2001.

[13] C. Mclean and F. Riddick, "The IMS Mission Architecture for Distributed Manufacturing Simulation," in *Proceedings of the 32nd Winter Simulation Conference (WSC)*, pp. 1539–1548, December 2000.

[14] R. M. Weatherly, A. L. Wilson, B. S. Canova, E. H. Page, A. A. Zabek and M. C. Fischer, " Advanced Distributed Simulation through the Aggregate Level Simulation Protocol," in *Proceedings of the 29th Hawaii International Conference on Systems Science (HICSS)*, pp. 407–415, January 1996.

[15] E. H. Page and R. Smith, "Introduction to Military Training Simulation: A Guide for Discrete Event Simulationists," in *Proceedings of the 30th Winter Simulation Conference (WSC)*, pp. 53–60, December 1998.

[16] R. Lannone, S. Miranda and S. Riemma, "Supply Chain Distributed Simulaiton: An efficient Architecture for Multi-Model Synchronization," *Simulation Modeling Practice and Theory*, vol. 15, no. 1, pp. 221–236, January 2007.

[17] S. Strassburger," On the HLA-Based Coupling of Simulation Tools, in *Proceedings of the 13th European Simulation Multiconference (ESM),*" pp. 45–51, June 1999.

[18] A. Borshchev, Y. Karpov and V. Kharitonov, "Distributed Simulation of Hybrid Systems with AnyLogic and HLA," *Future Generation Computer Systems*, vol. 18, no. 6, pp. 829–839, May 2002.

[19] M. Schumann, E. Bluemel, T. Schulze, S. Strassburger and K. Ritter, "Using HLA for Factory Simulation," in *Proceedings of the 1998 Fall Simulation Interoperability Workshop (SIW)*, September 1998.

[20] S. J. Turner, W. Cai, and B. P. Gan," Adapting a Supply Chain Simulation for HLA," in *Proceedings of 4th IEEE International Workshop on Distributed Simulation and Real-Time Applications(DS-RT)*, pp. 71–78, August 2000.

[21] R. Fujimoto, "Parallel Simulation: Parallel and Distributed Simulation Systems," in *Proceedings of the 33rd Winter Simulation Conference (WSC)*, pp. 147–157, December 2001.

[22] J. Misra, "Distributed Discrete-Event Simulation," *ACM Computing Surveys* ," vol. 18, no. 1, pp. 39–65, March 1986.

[23] D. Weyens and T. Holvoet, "A Formal Model for Situated Multi-Agent Systems," *Fundamenta Informaticae*, vol. 63, no. 2, pp. 125–158, May 2004.

[24] M. D'inverno, M. Luck and U. Ukmas, "Practical and Theoretical Innovation in Multi-agent Systems Research," *The Knowledge Engineering Review*, vol. 17, no. 3, pp. 295–301, September 2002.

[25] J. Ferber and J. P. Muller, "Influences and Reaction: A Model for Situated Multi-Agent Systems," in *Proceedings of the 2nd International Conference on Multiagent Systems (ICMAS)*, pp. 72–79, December 1996.

[26] A. Park, R. M. Fujimoto and K. S. Perumalla, "Conservative Synchronization of Large-Scaled Network Simulations," in *Proceedings of the 18th Workshop on Parallel and Distributed Simulation (PADS)*, pp. 153–161, May 2004.

[27] D. M. Nicol, "The Cost of Conservative Synchronization in Parallel Discrete Event Simulations," *Journal of the ACM*, vol. 40, no. 2, pp. 304–333, April 1993.

[28] X. G. Wang, S. J. Turner, M. Y. H. Low and B. P. Gan,

"Optimistic Synchronization in HLA-Based Distribution Simulation," *Simulation*, vol. 81, no. 4, pp. 279–291, April 2005.

[29] S. Xu and L. F. Mcginnis, "Optimistic-Conservative Synchronization in Distributed Factory Simulation," in *Proceedings of the 38th Winter Simulation Conference (WSC)*, pp. 1069–1074, December 2006.

Xu Xu received the B.E. degree and the M.E. degree in Computer Science in 2005 and 2008, respectively. He is currently a Ph.D. student at the Department of Management Science and Engineering of Tongji University. He is also a Research Assistant in Laboratory of Intelligence Decision and Decision Support.

His current research interests are data mining applied to customer relationship management and supply chain context, modeling and simulation of manufacturing systems and supply chain, fuzzy systems and knowledge discovery.

He has been participating to several research projects. He has published about 20 papers on national and international journals and conference proceedings, such as International Journal of Applied Mathematics and FUZZ-IEEE. He is a reviewer for Quarterly Cornell Hotel and Restaurant Administration. He is a member of the IEEE and a member of the IAENG.

Jie Lin is currently professor at the Department of Management Science and Engineering of Tongji University, and Director of Laboratory of Intelligence Decision and Decision Support. He received his Ph.D. degree in 1999.

His main fields of interest are decision support systems, modeling and simulation of manufacturing systems, application of multi-agent systems and soft-computing techniques (ant colony algorithm, group decision theory) for operations and supply chain management.

He has presided and finished several research projects. He has published about 50 papers on national and international journals and conference proceedings, such as International Journal of Production Research, Journal of Management Science in China and AMIGE. He is a member of the CNAIS.