NRC Publications Archive Archives des publications du CNRC

Implementing a hybrid simulation model for a Kanban-based material handling system

Hao, Q.; Shen, W.

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. / La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

For the publisher's version, please access the DOI link below. / Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

Publisher's version / Version de l'éditeur:

<https://doi.org/10.1016/j.rcim.2007.09.012>

Robotics and Computer Integrated Manufacturing, 24, 5, pp. 635-646, 2008-10-01

NRC Publications Record / Notice d'Archives des publications de CNRC:

<https://nrc-publications.canada.ca/eng/view/object/?id=4eb3c304-ae93-4971-b8ac-6aaffa5175c8>

<https://publications-cnrc.canada.ca/fra/voir/objet/?id=4eb3c304-ae93-4971-b8ac-6aaffa5175c8>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at

<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site

<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at

PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

National Research
Council Canada

Conseil national de
recherches Canada

Canada

Implementing a hybrid simulation model for a Kanban-based material handling system

IMTI-XP-591

Hao, Q.; Shen, W.

2008-05-01

A version of this document is published in / Une version de ce document se trouve dans:
Robotics and Computer Integrated Manufacturing, v. 24, no.5, 2008, pp. 635-646

The material in this document is covered by the provisions of the Copyright Act, by Canadian laws, policies, regulations and international agreements. Such provisions serve to identify the information source and, in specific instances, to prohibit reproduction of materials without written permission. For more information visit <http://laws.justice.gc.ca/en/showtdm/cs/C-42>

Les renseignements dans ce document sont protégés par la Loi sur le droit d'auteur, par les lois, les politiques et les règlements du Canada et des accords internationaux. Ces dispositions permettent d'identifier la source de l'information et, dans certains cas, d'interdire la copie de documents sans permission écrite. Pour obtenir de plus amples renseignements : <http://lois.justice.gc.ca/fr/showtdm/cs/C-42>

National Research
Council Canada

Conseil national
de recherches Canada

Canada

Implementing a Hybrid Simulation Model for a Kanban-Based Material Handling System

Qi Hao and Weiming Shen
National Research Council Canada
800 Collip Circle, London, Ontario N6G 4X8, Canada
[qi.hao; weiming.shen]@nrc.gc.ca

Abstract

Plant floor material handling is a loose loop in most assembly plants. Simulation offers a quick, controllable and tunable approach for prototyping complex material handling processes in manufacturing environments. This paper proposes a hybrid simulation approach, using both discrete event and agent-based technologies, to model complex material handling processes in an assembly line. A prototype system is implemented using a commercial multi-paradigm modeling tool. In this prototype, JIT principles are applied to both the production and the material handling processes. The system performance is evaluated and system optimization directions are suggested. The proposed hybrid modeling approach facilitates the implementation of a responsive and adaptive environment in that various “what-if” scenarios can be simulated under different simulation configurations and real-time situations.

Keywords: Agent-based systems, material handling, discrete event simulation, JIT, Kanban.

1. Introduction

Manufacturing enterprises are facing a substantially more complex situation than ever before because of the unpredictable market demands, growing product customizations, and fluctuating production environments. In order to stay competitive in the market, the decision making processes need to have the ability to adapt to the changing environment and to handle the system complexity. The ability to make use of the system flexibility and to retain flexibility itself is a primary requirement for the decision tools to be effective [1].

Material handling deals with material-flow related processes, including material requirement planning, scheduling and control of transportation resources (including AGVs, forklifts, and personnel), delivery of materials, buffering and optimization of materials. Material handling problem is easily neglected because most people think it is only an auxiliary process to production. From our observation, even in a well designed assembly line, when the whole line is optimized including its layout, processes, batching, scheduling, and operations, but material handling is still laid outside of the scope of control. In certain industrial sectors, material handling has been the major barrier that results in production breakdowns, low efficiency, and low performance of a production system.

As we all know that the optimization of work-in-process (WIP) is a critical objective for a production system since it represents saving of money in the product flow. In similar views, the flow of parts supply, resource allocation and buffer arrangement are also critical issues to be addressed before launching a production line. Kanban mechanism in the inventory management is applicable in this area since JIT principles in this scenario emphasize the delivery of right materials to the right place at the right time. Minimum and optimized material buffers are to be designed for individual assembly stations. Optimization of a production system, by itself, is a complex NP-Hard problem that is difficult to be solved using pure analytical/mathematical approaches. The targeted material handling system, which must be built on the basis of production, adds more unpredictable factors to the problem. Therefore, simulation has been by far the methodologies of choice in the majority of studies reported in the literature under the material handling topic [2].

Intelligent agent technology has recently been applied to material handling system simulations [3]. Since its emergence, agent technology has been widely recognized as a promising

paradigm for the next generation of design and manufacturing systems [3]. Agent technology makes a perfect fit for modeling dynamic and adaptive manufacturing systems. In the past eight years, we have developed agent-based technologies for manufacturing process planning and scheduling [4], shop floor control [5], collaborative product development [6], and inter-enterprise collaboration [7][8]. We have also completed an updated literature review on applications of agent technology to manufacturing in general [9].

In this paper, we propose a hybrid simulation approach to implement the material handling system simulation that combines both the discrete event (DES) and agent modeling approaches. To demonstrate the feasibility of applying agent technology to the simulation prototype, we constructed a discrete event model of a simplified pull production line and a Kanban based material handling system using AnyLogicTM [10]. Taking advantages of this multi-paradigm modeling tool, we build agent models for a number of components (either moving or stationary) in the discrete event model. As a result, the intelligence, controllability, and adaptiveness of the whole material handling system are greatly improved through agent-based technologies. For example, the vehicle agent hooking on its vehicle entity in the DES environment controls the routing, breakdown and repair activities, no matter whether the vehicle is moving around in the plant floor or it is idle in the parking lot. Moreover, the scheduling of transportation Kanbans is accomplished through the negotiation between the Kanban scheduling agent and a number of vehicle agents. With such capabilities, the system is able to simulate various dynamic situations and get more accurate information of transportation resources in general.

The rest of this paper is organized as follows: Section 2 provides a literature review on Kanban systems simulation and hybrid simulation technologies. Section 3 describes specifications of the material handling system and identifies the need of a hybrid modeling approach and a flexible modeling tool. Section 4 presents a prototype DES simulation environment created for a simplified pull assembly line and a Kanban based material handling system with an emphasis on the agent-based modeling implemented on top of the discrete event models. Finally, Section 5 draws conclusions and suggests some future research directions.

2. Literature review

From application point of view, in most manufacturing plants, material handling processes deal with material supplies in a supply chain or material inventories in an enterprise. The material

handling problem we intend to address in this paper refers to material and parts supply and delivery accompanying with the main production line. This section presents a brief research literature review on Kanban systems modeling and hybrid simulation technologies.

2.1 Kanban systems and simulation

There are two types of production control systems, namely *push* and *pull*. Material requirement planning (MRP) systems and Kanban control systems are the two most popular implementations of the push and pull strategies respectively. Pull is essentially a *replenishment* strategy that was initially designed for manufacturing environments producing repetitive products with high volumes [11].

According to Köchel et al. [12], Kanban research can be divided into two directions: *synthesis* and *analysis*. The synthesis approach aims at designing a new Kanban system that fulfills predefined conditions. Analytical mathematical models are commonly used to make such decisions [13][14]. Various analytical, mathematical or experimental models were proposed in the literature to address the Kanban allocation, planning and control issues. On the other hand, analysis approaches deal with performance analysis of Kanban systems under different structural deviations. Typical measurements employed in analysis are throughput, WIP, buffer size and average flow time.

Most searchable papers can be classified into the second category. Simulation has been by far the methodologies of choice in the majority of studies reported in the research literature of JIT systems [15]. Various simulations were done in Kanban-related topics targeting different purposes [16]. For example:

1. To study Kanban allocation problems, either for a fixed allocation of Kanbans [17][19] or a flexible Kanban allocation strategy [2][15].
2. To look at Kanban control issues in a Kanban-based system, such as Kanban sequencing [20][21], buffer capacity [22], control and maintenance policies [23][24].
3. To conduct performance analysis studies of Kanban-based systems [25][26].
4. To compare the Kanban control mechanism with others, i.e. traditional push systems [11][27].
5. To combine the simulation with optimization approaches, such as analytical models [2], neural networks [28][29] and evolutionary algorithms [12][30].

However, the models used in above Kanban simulation researches are all quite simple and far from reality. Different simulation experiments lead to different conclusions because of constrained experimental assumptions. From our investigation, we also found a few relatively complete developments towards realistic Kanban simulation systems. Early development includes a simulation generator for Kanban-controlled manufacturing systems [31]. This generator is itself written in C language but generates SIMAN code. It supports multiple manufacturing processes and queue selection rules. Its limitations include the lack of statistical tools, the lack of stochastic models, the lack of material handling devices, and the lack of experimental options. Köchel et al. [16] developed a Kanban simulator called KaSimIR for building simulation models of multi-stage Kanban-controlled systems. Work center, input buffer, transport unit, order queue, arrival store, and customer queue are basic classes that made up a realistic manufacturing environment. We noticed that the functionalities of transportation unit class in KaSimIR, not as the name suggested, only concern the transportation of Kanbans and Kanban containers. Neither the transportation of WIP between work cells nor the delivery of parts to work cells is considered.

The only Kanban system simulation tool that targets similar material handling problems as in our approach is KanbanSimTM [32] developed by PMC. Using WITNESSTM/Simul8TM discrete-event simulation language as simulation engine, KanbanSIMTM facilitates the optimization of Kanban numbers, Kanban container sizes, Kanban reorder points, material handling equipment (e.g., fork lift trucks, and Kanban train configurations), etc. It also captures variations in inventory levels at assembly stations, shows the dynamic traffic interactions, schedules material handling equipment, and predicts the impact of material handling on system throughput. Our approach presented in this paper is different from KanbanSimTM in that we combine an agent-based control mechanism on top of a pure discrete simulation system for Kanban-based material handling, which will be discussed in details in the following sections.

From application point of view, the researches of Kanban-based technologies can be classified into three categories: 1) production control; 2) inventory management; and 3) supply chain management. None of above literature has touched the topic of the material handling issues at the plant floor level, by which we refer to material and parts supply and delivery accompanying with the main production line. In most manufacturing plants, inventories are only virtually under the control of either an MRP II or a Kanban system at the enterprise level, but not physically at the real-time dynamic plant floors. As a result, material handling becomes the major barrier that results in low performance of a production system.

2.2 Hybrid simulation approaches

The use of simulation technologies is very popular in modeling complex manufacturing systems, because a simulation model can give early insights and estimates of behaviors for very complex systems. Four categories of modeling paradigms are identified by the modeling and simulation community [33]: *system dynamics*, *dynamic systems*, *discrete event*, and *agent-based*.

The construction and validation of simulation models is not an easy task. To focus more on the domain problem itself rather than the implementation issues, one often needs to seek a matured modeling and simulation product on the market to quickly build desired models. There are numerous commercial or research-based simulation tools which employ different modeling paradigms [34], for example, VenSim, PowerSim for system dynamics simulation; Arena, ProModel, FlexSim, Simul8, Delmia for discrete event simulation; Swarm, MASON, SimWalk, RePast for agent-based simulation; and MATLAB, Labview, VisSim, Simulink for dynamic systems simulation. A set of important attributes to be considered when selecting simulation software tools for manufacturing is given by Law and McComas [35]. In general, a simulation tool is designed to fit one specific modeling paradigm or one application domain. These traditional tools are facing challenges when simulating large-scale, complex, highly dynamic heterogeneous systems. AnyLogic™ [10] represents the newest trend and technology in modeling and simulation – the hybrid modeling technology. Up to now, it is the only hybrid/multi-paradigm simulator in the market that delivers complex modeling capacities in an all-in-one solution.

Only a few records can be found in the research community of hybrid simulation technologies in manufacturing. Venkateswaran and Son [36] decomposed the entire production planning problem into a two-level hierarchical system, using a system dynamics (SD) simulator for enterprise level planning (aggregate planning) and a discrete event simulation (DES) model for the shop-level scheduling (detail planning). Feedback control loops are employed at both levels to monitor the performance and update the corresponding control parameters. Another similar framework of discrete-continuous combined modeling (SD and DES) for a supply-chain system was proposed by Lee et al. [37]. They represented the continuous aspects of the supply chain using mathematical equations, which are then integrated with the discrete aspects of the supply chain such as transportation activities. Maione and Naso [38] proposed a kind of hybrid modeling approach combining multi-agent control and discrete event formalism. The authors modeled an agile manufacturing plant as a distributed multi-agent network, in which two major kinds of

agents, part agents and resource agents, interact and negotiate to control the part flow and the machine processing sequences. In this system, each agent is modeled as an atomic discrete event system so that it can make its own decisions depending on its states, decision rules, and the environment.

The initiative of hybrid agent/discrete event simulation approach in this paper, although the same terms are used, is quite different from Maione and Naso's approach [38]. We implement a discrete event simulation environment and add agents upon it instead of authorizing the controllability of slave objects (especially those slave objects generated and routed by the simulator). This kind of control flexibility (or "intelligence") is not possible by using any pure discrete event modeling simulator. To the best of our knowledge, this work is the first to successfully interface discrete event simulation systems and agent-based systems for the purpose of manufacturing material handling problems.

3. System specification

2.1 The simulation scenario and assumptions

Simulation of material handling in a pull production setting is the major purpose of this research. A Kanban-based material handling system is investigated to make it in line with the pull production line. For the convenience of a common understanding, a general assembly line scenario is chosen as the simulation prototype, as shown in Figure 1.

(Insert Figure 1 here.)

The following assumptions are to be clarified:

- Production type: mass production of mixed product models.
- Production organization: U-shaped production/processing lines, which integrate the manufacturing processes into an even, balanced and continuous material flow.
- Modular/station arrangement: the flow-of-products-oriented layout of processes asks for careful separation of the whole process to production stations according to the optimized process rate of the whole line and the granularity of material control.
- Pull production control: a pull control mechanism is applied for the control of production. Products are carefully sequenced before going on the line. There is no WIP

buffer arrangement on the main assembly line. Production rate is a constant number that is optimized by operation research practices.

- Material handling process: materials/parts are accumulately assembled to the product sequentially at different stations. Material handling process refers to the transportation of the required materials/parts to the right station from the central inventories when they are requested.
- There are many transportation vehicles (illustrated by the icon of a forklift) moving around the whole plant to finish the delivery of materials/parts. The vehicles are mounted with wireless communication capacities and simple transaction systems for material handling.
- There are centralized inventories supplying materials/parts to the assembly line. The capacity of these inventories is assumed to be unlimited, in other words, the amount of materials/parts in stock is infinite.

2.2 Specifications of the material handling simulation – Kanban and agent aspects

The Kanban-based material handling approach proposed in this paper borrows similar principles from JIT-based production control and JIT-based inventory management in that: the right material is delivered from its inventory to the right production site, at the right time and in the right amount. Here, material transportation in the plant floor is considered as individual tasks. A task requires a Material Kanban to be delivered. For the convenience of description, material Kanban is simply referred as Kanban in the rest of the paper. In Figure 1, a material request signal is firstly generated by an assembly station running out of a part supply. After occupying a Kanban, the request is then ready to be allocated to a forklift truck and delivered to the destination station. In our view, a material handling system based on Kanban concept is not merely a pure event based system. In a pure event system, an event calls for a system response immediately; while in a Kanban-based system, a generated event gets processed only after obtaining a physical object – Kanban. In other words, the processing (transportation) of a material requirement event holds until the system releases a Kanban and the event is qualified to occupy this free Kanban among all other events. Based on a Kanban control mechanism, we believe that the material handling system is able to reach a natural balance of material requirements and transportation activities through delicate arrangement and management of Kanbans.

Hundreds of dispatching rules, such as SIRO (service in random order), FCFS (first come first serve), EDD (earliest due date), SDTF (shortest delivery time first), or MINCOST (minimum cost) can be applied to solve the scheduling problem (or sequencing problem) of Kanbans. However, in order to demonstrate agent characteristics in the simulation, we envisage another agent-based Kanban scheduling mechanism, in which a material request embedded in a Kanban is broadcasted to a number of vehicles moving in the scope of a wireless network; then through negotiation between the Kanban agent and multiple vehicle agents, the task is confirmed by a vehicle and being delivered finally to the right station.

Another idea to be proved is individual control of vehicle behaviors through its representative vehicle agent. The basic requirement for vehicle control is to identify the ID of a vehicle. Therefore, it is possible for the system to trace and control each vehicle in its data, schedule, delivery route, and other actions, i.e. failure, repair, rest, recovery, etc. Each transportation vehicle has a separate control agent to make its own decisions, including task sequence, task schedule, movement control, loading and unloading operations, or even collision resolution decisions if necessary.

2.3 Choice of simulation package - AnyLogicTM

Generally speaking, in a discrete simulation environment, it is very hard to implement dynamic controls over simulation components; especially those passive objects generated and routed by the simulation engine, such as products, vehicles in this simulation. We implement the aforementioned material handling simulation based on AnyLogicTM because of its distinctive multi-paradigm modeling facilities. The term “passive” object is used in contrast with the term “active” object used by AnyLogicTM. According to AnyLogicTM, an active object is a very flexible component that can fulfill all kinds of requirements of multi-paradigm modeling, for example, an active object may further consists of encapsulated active objects, and it can have variables, parameters, ports, equations, state charts, timers, animations, etc. On the other hand, a passive object is generally controlled by the simulator and is named as “Entity Message” in AnyLogicTM.

A distinctive feature of the designated material handling simulation is that it is a hybrid approach combining discrete event simulation and intelligent agents. The proposed simulation system possesses functions that surpass traditional simulations: 1) ability to facilitate run-time reconfiguration, for example, the arrangement of assembly tasks to manufacturing stations can be adjusted during the execution of a simulation, so that the bottleneck (of the line) and system

responses can be constantly changing; 2) another example is that each component is manageable not only in its configuration parameter, but also controllable in its behaviors. In contrast, other simulation systems read a batch file for configuring a simulation system. It is difficult for people to analyze dynamic system behaviors in separate simulation launches.

Another reason for choosing AnyLogic™ is that it is a real object-oriented modeling tool and is 100% JAVA coding. The main building blocks, active objects, can be used to model very diverse objects of the real world: processing stations, resources, people, hardware, physical objects, etc. Customized JAVA coding can be added to any of its encapsulated modeling entities, such as properties, equations, state charts, startup codes, and user-defined functions. Moreover, user can choose to build a project into a standalone Java application (.jar) or into a Java Applet that is executable without AnyLogic™ environment. It is also much easier to integrate with other applications, tools, and physical devices. All these features make AnyLogic™ a perfect choice for our material handling simulation.

4. Implementation of the hybrid simulation of a Kanban-based material handling system

The implementation of the material handling simulation is based on the Enterprise Library provide by AnyLogic™ for discrete event modeling. The implemented components of the JIT material handling system are shown as in Figure 2. The functionalities of the Active Object classes, Message classes and data source are briefly described below:

(Insert Figure 2 here.)

- **Root:** The **Root** class is the root class of the system. The system is executed from this class.
- **PullSource:** The **PullSource** class generates sources of products in the assembly line in both push and pull mode.
- **PullStation:** The **PullStation** class represents left-to-right workstations in the system. In its animation, a product moves from left to right.

- **InvertPullStation:** The **InvertPullStation** class represents right-to-left workstations in the system. In its animation, a product moves from right to left.
- **InterNode:** The **InterNode** class represents a node in the route path of vehicles. It wraps a **Node** Active Object and is able to handle failed vehicle.
- **Link:** The **Link** class wraps a **Segment** Active Object to provide connectivity for routes in the map of the plant floor.
- **Inventory:** The **Inventory** class is responsible for vehicle handling, Kanban handling and material requests handling.
- **KanbanQueue:** The **KanbanQueue** class is responsible for storing **Kanban** and combining Kanbans with material requests.
- **KanbanScheduleAgent:** The **KanbanScheduleAgent** class is responsible for scheduling a **Kanban** to a **vehicle** randomly or through a bidding process.
- **VehicleAgent:** The **VehicleAgent** class is the agent representative of a **vehicle** Entity Message. This class is used to communicate with a **KanbanScheduleAgent** to schedule Kanbans to vehicles.
- **Kanban:** A **Kanban** message represents a Kanban.
- **KanbanMsg:** A **KanbanMsg** message is used to control **KanbanQueue**.
- **Pallet:** A **Pallet** message represents a product.
- **Part:** A **Part** message represents a part.
- **ReleaseSig:** The **ReleaseSig** message is used to let all stations release current holding products.
- **Request:** A **Request** message represents a material handling request.
- **StateMsg:** The **StateMsg** message is used to send the station failure information.
- **StockMsg:** The **StockMsg** message is used to send the station stock information.
- **Task:** A **Task** message represents a task and its associated stock information.
- **VCMsg:** The **VCMsg** is used for messages among agents.
- **Vehicle:** A **Vehicle** message represents a material handling vehicle

- **Config:** an EXCEL spreadsheet or an ODBC data source that provides system configuration information and stores statistical results of the system.

A sample structure diagram of PullStation is shown in Figure 3. The structures of other active objects are modeled similarly using enterprise libraries. An animation of plant layout as specified in Figure 1 is generated at run-time by fragmented animations of active objects, such as pull stations, links, nodes, and segments. Sample screenshots of animations can be found in the rest of this paper.

(Insert Figure 3 here.)

The following sections describe major implementation issues in this simulation prototype in more details including production simulation, Kanban-based material handling simulation and agent control aspects. Above objects listed in Figure 2 will also be clarified further from different perspectives.

4.1. Production Simulation

Both the push and pull production modes can be simulated in the implemented prototype.

In the push production mode, the PullSource keeps releasing new products to the line in a fixed production rate in regardless of different processing times of PullStations. In the same manner, the PullStations keep dumping products to their succeeding stations thus causing lining up situation on conveyors.

The realization of a pull control mechanism needs much more effort. Each station is doing some operations to the product on hand, say, P2, P3 and P4. Assume that the time required for the operations at a station is not a fixed number due to: 1) complexity of operations on different products; 2) experience and tiredness of human operators; 3) availability of tools, etc. In this occasion, a flexible production rate is required so that when and only when all stations finish their duties, the production is “pulled” forward by one step. In the next step, station 1 will grab a new product P1, and P2, P3, and P4 will switch to succeeding stations, respectively.

(Insert Figure 4 here.)

In Figure 4, once a station finishes its operations on the holding product, it sends a Work Done signal to a counter and then blocks its status. Until the count of signals reaches the total number of stations (3 in Figure 4), a Release Signal is generated which then unlocks all stations from blocking status. In this occasion, all stations “pull” a product from their previous stations when a production signal unblocks their status from “blocking” to “normal”. Figure 5 shows one of the state charts implemented in PullStation that controls the switch between the pull and push modes, as well as status transitions of the simple pull production mode described in Figure 4.

(Insert Figure 5 here.)

Similarly, accompanying with the pull mechanism in PullStation, a pull mechanism is also required to implement in the production source (PullSource) that controls the release of products to the production line. The “one-entity-per-time” structure shown in Figure 6, which provides the functionality to release a single entity every time when a signal is given, is implemented by a Hold object in the PullSource to hold a new product until the Release Signal unblocks the Hold object. This structure is also used in the Inventory Active Object and the KanbanScheduleAgent Active Object to release one vehicle and one Kanban each time.

(Insert Figure 6 here.)

In Figure 6, the default status of the Queue is unblocked and the default status of the Hold is blocked. Every time after an entity enters the Queue, the Queue will be blocked. When a Release Signal comes, the Hold object will be unblocked, thus the entity in the queue can flow through the Hold. When another entity enters Hold, the Hold will be blocked again, while the Queue will be unblocked at the same time.

4.2. Material Handling Simulation

Material transportation and delivery is the primary purpose of this simulation system. Material handling simulation can only be achieved based on the availability of production simulation, since the source of material transportation tasks is the material requirements accompanied with the production process.

Each instance of PullStation Active Object maintains a set of storage levels of its materials/parts' supplies. Within each production step, a station consumes certain units of materials/parts from its storage. Therefore, a material handling event is generated when a station runs out of any of its material/part supplies. There are two levels of material handling events to represent two situations happened at stations: Material Requirement (MR) and Material Query (MQ). A material requirement is an event signaling a situation when the stock level of the referred material/part is low at a station and a full package of this material needs to be delivered to this station. A material query is an event designed to signal a higher level of material shortage. After a MR signal, the station continues to consume materials/parts at each production step. At a certain level, a MQ event is generated and it indicates that the material/part requested in the previous MR event has not been delivered to the station yet. The current balance reaches such an emergent level that it needs an emergent care from the supporting staff. The system automatically traces the queried material information on behalf of the users and reports the status of material delivery. When the storage level at any station drops to zero, the source station will generate a failure signal which then forces the whole line to a stop. Figure 7 shows a line stop situation caused by the storage of Part D and H at Station #3 and #7, respectively. The sources of failure (the 3rd and 7th station) are highlighted using a different light signaling. The corresponding delayed material requests are also highlighted in the MR List using red color.

(Insert Figure 7 here.)

In a pure event system, the generated MR or MQ are put into separate queues, and if possible, being processed immediately once the events are generated.

In this system, we designed a Kanban control mechanism on top of the events so that they are able to be handled tactically. Kanban Queue is a kind of data storage for all Kanbans currently circulating in the simulation environment. It has several “containers” (logical structure) to maintain Kanbans with different properties. In this material handling simulation, we assume two logical Kanban containers with adjustable capacity to separately control the material flows of the two aisles in the plant floor. By adjusting the number of Kanbans in both aisles and the total number of vehicles moving around, the performance of the material handling system is able to be balanced against some statistic measures, i.e., average Kanban utilization, average vehicle

utilization, overall throughput, average MR wait time, average MR satisfaction time, average/maximum vehicles simultaneously serving in an aisle, production failure count, etc.

4.3. Implementation of vehicle agents and Kanban scheduling process

Based on the discrete event simulation of a pull production line and a Kanban-based material handling system, we add intelligent agents to this simulation prototype to demonstrate our agent-based hybrid simulation technology. The ability of intelligent agents to interface with physical devices and to warp legacy functional modules has been proved by research projects or industry applications [9]. In this paper, we emphasize the potential of agents in representative of simulation entities in a pure simulation environment.

With an agent stays in the middle, the simulated object in the simulator can have the opportunities to be connected to a physical device at lower level, and to a control module at a higher decision making level. On one hand, a virtual simulation environment is possible to receive feedbacks from the physical world so that it can perform adaptively and dynamically just like a real world operational system; on the other hand, the behavior of a virtual simulation environment is possible to be controlled by higher level software components to make the simulation more targeted, smart and interesting.

To demonstrate the hybrid simulation approach, we implemented two kinds of agents, vehicle agent and Kanban scheduling agent, in the discrete event simulation of the material handling system. The following subsections introduce our implementation of these agents and some experiments performed to prove the hybrid modeling approach.

4.3.1 Vehicle agent and Kanban scheduling agent

A vehicle needs to be modeled as an agent to participate in the negotiation-based Kanban scheduling process and to facilitate intelligent control over individual vehicles. In AnyLogicTM, the discrete event Enterprise Library models a vehicle as a Message entity which does not have the full capabilities of an Active Object. In general, Message class is designed by AnyLogicTM to model moving components that pass useful information between active objects. Apart from the regular “message” that carries information, command or signal, messages can model an entity flow in a manufacturing environment, where messages represent entities (items) that are being served, produced, or otherwise acted on by manufacturing processes, such as parts, products, AGVs and pallets. In this simulation, vehicle entities belong to the “message” class in the discrete event simulation to model moving forklift trucks that move around the plant floor. As its

definition, unlike an active object, a vehicle entity is a passive object that is served by other active objects. It cannot actively change its status, schedule, or route plan.

To implement individual control over vehicles, we implement an active object class which we call `VehicleAgent` to represent each individual vehicle. The following steps can bind a vehicle agent with a vehicle Entity Message:

- (1) Create a `VehicleAgent` active object which is responsible for performing agent related actions of a vehicle, including a Kanban bidding protocol and some actions for vehicle control.

- (2) Add a `VehicleAgent` type property, called “myAgent”, to the vehicle Entity Message Type.

- (3) When a vehicle Entity Message is constructed, it will also construct its “myAgent” property through calling myAgent’s `onStartup()` method. This method will call the `VehicleAgent`’s `create()` and `startup()` methods to create and initialize this agent. Upon initialization, a vehicle agent also sends a message to a specific port of the Root to register it so that the system will know the amount of hooked-up vehicles at any time.

The control of vehicles is a multi-facet mission including: 1) motion control – moving, loading/unloading, stop; 2) status control – idle, active, repair, pause, delivery, return; 3) route control – route plan, route choice, return route, collision resolution; 4) bidding process control – giving a bid, responding to an awarded Kanban; 5) schedule control – local schedule, drop tasks, rearrange tasks. With the help of its vehicle agent, a vehicle Entity Message can fulfill all these five missions.

The `KanbanScheduleAgent` (KSA) is built as an active object that is in charge of a bidding based scheduling process for the allocation of Kanbans to vehicles. The next section will introduce the interactions between vehicles agents and the KSA for a bidding process.

4.3.2 Bidding-based Kanban scheduling

A number of dispatching rules applied in manufacturing, such as SIRO (service in random order), FCFS (first come first serve) and EDD (earliest due date), can be applied to solve the scheduling problem of material Kanbans. In this paper, as our major purpose of research is on agent-based hybrid simulation technology, an agent-based Kanban scheduling mechanism is implemented.

There is a state chart designed in both the `VehicleAgent` (VA) class and the `KanbanScheduleAgent` (KSA) class to handle the message interactions during a negotiation process. A simple contract net-based protocol is implemented for this purpose. Four rounds of

messages are sent back and forth: 1) KSA->VA, call for proposal; 2) VA->SCA, proposal; 3) SCA-> VA, award (to the selected vehicle); 4) VA-SCA, inform (with material delivery information). After the KSA receives all proposals from vehicle agents that are supposed to provide bids, it simply always selects the vehicle with the highest utility number.

4.3.3 Vehicle control

With the help of vehicle agents binding to vehicle entities in the AnyLogic™ simulator, the following scenario can be expected to be realized in a discrete event material handling simulation environment:

- Each vehicle has a unique ID and is traceable in the system;
- A vehicle has its own plan, schedule; it has full control over its decisions, such as which task to deliver next and which route should choose for a delivery;
- The driver of a vehicle can deactivate a vehicle from the system to take personal activities (rest, lunch break, etc.) or when the vehicle malfunctions and needs a repair. Similarly, a vehicle can be hooked back to the system at any time once it is ready to provide services.
- A vehicle can, at any time, have a failure and block any segment of the route. The tasks that it carries in its schedule can be taken over by other vehicles.
- Vehicles can coordinate to solve the route blocking and route conflict problems.
- The system can get accurate statistical information of individual vehicles and individual drivers, such as utilization, service time, failure rate, delivery delay, re-allocation rate, etc.

For a simple demonstration of vehicle agents over vehicle entity messages, we implemented a vehicle failure situation. When a specific vehicle fails and needs repair, the simulator needs to send it to the maintenance area of the inventory immediately. However, neither the Segment Active Object or the Node Active Object is able to operate on the trespassing vehicle Entity Message directly. In order to distribute a vehicle failure command to the required vehicle and re-route it to a place other than its original destination, we wrapped a Node Active Object as an InterNode Active Object as shown in Figure 8.

(Insert Figure 8 here.)

After a vehicle enters an InterNode Active Object, it will be sent to a SelectOutput to check if this vehicle is set to failure by its agent. If it is the case, its destination will be set to the maintenance area and the vehicle then exits this Node. After the vehicle is grabbed again by the enterVehicle object (of the Inventory), it will be directed to the maintenance area by the nodeParking object automatically (Figure 9). Otherwise, the vehicle will be sent to the outputTrue port which has the same functionality of an original Node Active Object's output port.

When a Kanban is scheduled to a specific idle vehicle, that vehicle needs to move to the nodeLoading Node to pickup the Kanban. Normally, according the Enterprise Library, an idle entity (vehicle in this occasion) will stay in a parking node and the entity's destination property is set to null. However, by default, idle entities can only be picked up randomly in the AnyLogicTM simulator. This does not fulfill the requirement of Kanban scheduling. In order to implement the functionality of enforcing a specific vehicle to move from the parking Node to the loading Node, we added an Enter Active Object and a Queue Active Object in the Inventory as shown in Figure 9.

(Insert Figure 9 here.)

In this situation, all incoming idle vehicles will be lined up in the nodeParkingQueue. When a specific vehicle wins a Kanban bid, this vehicle will be removed from the nodeParkingQueue using remove() and its destination property will be set to the nodeLoading node. Then this vehicle will re-enter the network again and be taken by the enterVehicle and then by the nodeParking. Since the destination property is set to the target node (which is a loading node), this vehicle will be automatically routed to its new destination.

5. Conclusions and future work

A simulation model is a dynamic or an operating model of a system or problem entity that mimic the operating behavior of the system or problem entity. For fast construction of complex manufacturing systems, we need to find an effective simulation tool to perform important decision variations before launching a production line or re-engineering an operating line. This paper proposed a hybrid simulation modeling approach which combines the discrete event modeling and

agent-based modeling technologies. A discrete event model of the Kanban-based material handling prototype is built based on a simplified assembly line using a multi-paradigm modeling tool called AnyLogicTM. In this simulation, an agent is binding with an Entity Message (vehicle message) every time when the simulator generates an Entity Message instance. As proved by the developed prototype, the intelligence, controllability, and adaptiveness of the whole material handling system are greatly improved through agent-based technologies. The developed material handling simulation is smart in terms of: 1) using a bidding protocol between the Kanban scheduling agent and vehicle agents for its Kanban scheduling process; and 2) individual control of individual transportation vehicles in certain behaviors, such as routing, breakdown and repair activities. With such capacities, the system is able to simulate various dynamic situations and get more accurate information of transportation resources in general. From a comprehensive literature review on topics of Kanban systems and simulations, hybrid simulation technologies, and agent-based material handling systems, this research is the first attempt to combine agent-based modeling and discrete event modeling in a simulation environment.

JIT principles are applied to both the production and material handling processes. The primary objectives of this research is to develop a prototype software system to simulate a flexible pull production line, to demonstrate a dynamic material handling environment, to optimize a material transportation flow, and to make accurate statistics and predictions. Based on the implemented prototype at this stage, we can continue carrying out some interesting topics:

- Optimizing the material handling environment by doing comparative studies over some simulation parameters, such as allocation problems of Kanbans, Kanban sequencing technologies, and the combination effect of Kanban and vehicle allocations.
- The ultimate goal of this simulation is to optimize the arrangement of buffers and their storage levels at stations so that the whole production line can reach a balance in terms of material requests and material handling. As we all know that, a real balanced material handling system can only be achieved through balanced material requirement events from stations. This optimization can be done using analytical modeling approaches, evolutionary based approaches, or hybrid simulation approaches.
- Building accurate prediction models based on system performance measures, such as vehicle utilization rate, Kanban utilization rate, average service time for Kanban, failure rate, average waiting time for request events, average waiting time for Kanban at each vehicle, and production shutdown rate.

- Implementing advanced functionalities for agent-based vehicle control in route conflicts resolution, route blocking, and task takeover.
- Simulating and investigating human factors in material handling processes.

Finally, from system integration point, research and development work can be explored in two directions: the integration of the material handling simulation environment with an agent platform (such as JADE) so that agents representing simulation entities can reside in an agent community in order to make more advanced and independent decisions; and the integration of the material handling simulation environment with real-time physical devices in the shop floor so that the system can perform both the monitoring and simulation functions. Only when the integration goals are achieved, the real value of this research can be revealed.

References

- [1] Lin GY, Solberg JJ. An agent-based flexible routing manufacturing control simulation system. Proc. Winter Simulation Conference, 1994; Lake Buena Vista, FL.
- [2] Martins K, Lewandrowski U. Inventory safety stocks of Kanban control systems. Production Planning and Control, 1999; 10(6): 520-9.
- [3] Shen W, Norrie DH. Agent-based systems for intelligent manufacturing: a state-of-the-art survey. Knowledge and Information Systems, 1999; 1(2): 129-56.
- [4] Shen W, Lang S, Wang L. iShopFloor: an Internet-enabled agent-based intelligent shop floor. IEEE Trans. on Systems, Man and Cybernetics-Part C, 2005; 35(3): 371-81.
- [5] Wang L, Wong B, Shen W, Lang S. Java3D enabled cyber workspace. Communications of the ACM, 2002; 45(11): 45-9.
- [6] Hao Q, Shen W, Zhang Z, Park SW, Lee JK. Agent-based collaborative product development engineering: An industrial case study. Computer in Industry, 2006; 57(1): 26-38.
- [7] Hao Q, Shen W, Wang S. Agent-oriented manufacturing scheduling services for enterprise collaboration. SAE World Congress, 2005; Detroit, MI.
- [8] Shen W, Li Y, Hao Q, Wang S, Ghenniwa H. A Service Oriented Integration Framework for Collaborative Intelligent Manufacturing. Robotics and Computer-Integrated Manufacturing, 2007; 23(3): 315-325.
- [9] Shen W, Hao Q, Yoon HJ, Norrie, DH. Applications of agent systems in intelligent manufacturing: An updated review. Int. J. of Advance Engineering Informatics, 2006; 20(4): 415-31.

- [10] XJ Techonologies. AnyLogic™, <http://www.xjtek.com/anylogic/>
- [11] Krishnamurthy A, Suri R, Vernon M. Re-examining the performance of MRP and Kanban material control strategies for multi-product flexible manufacturing systems. *Int. J. Flexible Manufacturing Systems*, 2004; 16(2): 123-50.
- [12] Köchel P, Nieländer U. Kanban optimization by simulation and evolution. *Production Planning and Control*, 2002; 13(8): 725-34.
- [13] Gunasekaran A, Goyal SK, Martinkainen T, Yli-Olli P. Modeling and analysis of Just-In-Time manufacturing systems. *Int. J. Production Economics*, 1993; 32: 23-37.
- [14] Berkley BJ. A review of Kanban production control research literature. *Production and Operations Management*, 1992; 1(4): 393-411.
- [15] Gupta SM, Al-Turki YAY. The effect of sudden material handling system breakdown on the performance of a JIT system. *Int. J. Production Research*, 1998; 36(7): 1935-60.
- [16] Köchel P, Nieländer U, Sturm M.: KASIMIR – Object-oriented KAnban SIMulation Imaging Reality. Research Report, Chemnitzer Informatik-Berichte CSR-01-03, 2001; Chemnitz University of Technology, Chemnitz, Germany.
- [17] Andijani AA. A multi-criterion approach for Kanban allocations. *Omega: International Journal on Management Science*, 1998 ; 26: 483-93.
- [18] Askin, H, Mitwasi MG, Goldberg JB. Determining the number of Kanbans in multi-item Just-in-Time systems. *IIE Transactions: Design and Manufacturing*, 1993; 25: 89-98.
- [19] Bard JF, Golany B. Determining the number of Kanbans in a multiproduct, multistage production system. *Int. J. Production Research*, 1991; 29: 881-95.
- [20] Berkley BJ, Kiran AS. A simulation study of sequencing rules in a Kanban-controlled flow shop. *Decision Sciences*, 1991; 22: 559-82.
- [21] Hum SH, Lee CK. JIT scheduling rules: A simulation evaluation. *Omega: International Journal on Management Science*, 1998; 26: 381-95.
- [22] Berkley, BJ. Effect of buffer capacity and sequencing rules on single-card Kanban system performance. *Int. J. Production Research*, 1993; 31: 2875-93.
- [23] Savsar M. Effects of Kanban withdrawal policies and other factors on the performance of JIT systems: A simulation study. *Int. J. Production Research*, 1996; 34: 2879-99.
- [24] Savsar M. Simulation analysis of maintenance policies in Just-in-Time production systems. *Int. J. Operations and Production Management*, 1997; 17: 256-66.

- [25] Karaesmen F., Dallery Y. A performance comparison of pull type control mechanisms for multi-stage production control. *Int. J. Production Economics*, 2000; 68: 59-71.
- [26] Yavuz IH, Satir A. A Kanban-based simulation study of a mixed model Just-in-Time manufacturing line. *Int. J. Production Research*, 1995; 33: 1027-48.
- [27] Bonney MC, Zhang Z, Head MA, Tien CC, Barson RJ. Are push and pull systems really so different? *Int. J. Production Economics*, 1999; 59: 53-64.
- [28] Savsar M, Choueiki H. A neural network procedure for Kanban allocation in JIT production control systems. *Int. J. Production Research*, 2000; 38: 3247-65.
- [29] Viharos, ZJ, Monostori L. Optimization of process chains and production plants by using a hybrid-, AI-, and simulation-based approach. *Lecture Notes in Artificial Intelligence (Proc. IEA/AIE 2001)*; 2070: 827-35.
- [30] Nieländer U. Simulation optimization of Kanban systems using a non-standard genetic algorithm. *Proc. 4th ISIR Research Summer School on Inventory Modeling, School of Mathematics Sciences*; p.137-46.
- [31] Autug H, Dogan CA; A framework and a simulation generator for Kanban-controlled manufacturing systems. *Computers and Industrial Engineering*, 1998; 34: 337-50.
- [32] KanbanSIMTM, PMC. <http://www.pmc corp.com/kanbanSIM.shtm>
- [33] Modeling approaches, XJ Technologies. <http://www.xjtek.com/anylogic/modeling-approaches/>
- [34] Rizzoli AE. A collection of modeling and simulation resources on the Internet. Dec 2005 (last update). <http://www.idsia.ch/~andrea/simtools.html>
- [35] Law AM, McComas MG. How to select simulation software for manufacturing applications. *Industrial Engineering*; 7(7): 29-33.
- [36] Venkateswaran J, Son YJ. Hybrid system dynamic-discrete event simulation-based architecture for hierarchical production planning. *Int. J. Production Research*, 2005; 43(20): 4397-429.
- [37] Lee YH, Cho MK, Kim YB. A discrete-continuous combined modeling approach for supply chain simulation. *Simulation*, 2002; 78: 321-9.
- [38] Maione G, Naso D. Modelling adaptive multi-agent manufacturing control with discrete event system formalism. *Int. J. Systems Science*, 2004; 35(10): 591-614.

List of Figures

Figure 1: A scenario of material handling simulation in an assembly line

Figure 2: System components

Figure 3: Structure diagram of PullStation

Figure 4: A simple pull production mechanism

Figure 5: A state chart of PullStation

Figure 6: One-entity-per-time structure

Figure 7: Production stop caused by a station running out of materials

Figure 8: Structure of the InterNode Active Object

Figure 9: Node with an Enter and a Queue

Figure 1: A scenario of material handling simulation in an assembly line

Figure 2: System components

Figure 3: Structure diagram of PullStation

Figure 4: A simple pull production mechanism

Figure 5: A state chart of PullStation

Figure 6: One-entity-per-time structure

Figure 7: Production stop caused by a station running out of materials

Figure 8: Structure of the InterNode Active Object

Figure 9: Node with an Enter and a Queue