# SIMULATION OF AGV SYSTEM – A MULTI AGENT APPROACH

KOMMA, V. R.; JAIN, P. K. & MEHTA, N. K.

***Abstract:*** *Multi agent system is a tool to model concurrent behaviour of complex systems. Agents are autonomous and computational entities that perceive their environment through sensors and act upon their environment through effectors. Application of agent technology in simulation of AGV system helps in modeling the concurrent behaving entities close to the real system. This chapter presents the development of an Agent-Based Shop Floor Simulator (ABSFSim) with focus on AGV systems. ABSFSim was developed on JADE platform. The steps involved were development of ontology for AGV system, hybrid communication protocols, agent synchronization mechanism and modeling of agents. In addition, a hybrid contract net based dispatching (HCNBD), a new approach for continuous dispatching of parts to AGVs, has been proposed and compred with AGV dispatching approaches Longest Waiting Entity and Nearest Entitiy which were implmented in ProModel environment.*

***Key words:*** *agent-based simulation, AGV system, Java Agent DEvelopment framework (JADE), contract net protocol, AGV dispatching*

**Authors´ data:** Dr. **Komma**, V[enkateswara] R[ao] *; Prof. **Jain**, P[ramod] K[umar]**; Prof. **Mehta**, N[arinder] K[umar]**, *Department of Mechnical Enginering, Motilal Nehru National Institute of Technology Allahabad, Allahabad – 211004, India, ** Department of Mechanical and Industrial Enginering, Indian Institute of Technology, Roorkee, Roorkee – 247667, India, vrkomma@gmail.com, pjainfme@iitr.ernet.in, mehtafme@iitr.ernet.in

**This Publication has to be referred as:** Komma, V[enkateswara] R[ao]; Jain, P[ramod] K[umar] & Mehta, N[arinder] K[umar] (2012). Simulation of AGV system – A Multi Agent Approach, Chapter 01 in DAAAM International Scientific Book 2012, pp. 001-018, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-901509-86-5, ISSN 1726-9687, Vienna, Austria
DOI: 10.2507/daaam.scibook.2012.01

## 1. Introduction

Today's manufacturing system is facing great challenges from customers in meeting the sudden changes in demand and increased product variety due to more personalized products. This market situation forces the manufacturing systems to be more flexible, more agile and more reliable. Discrete-Event Simulation (DES) is the commonly used tool in industries to study the performance of a manufacturing system. Most of the commercial software packages used in industries for simulation of manufacturing systems were implemented either with functional or object-oriented programming approaches (Law & Kelton, 1991).

Multi Agent System (MAS) is an important and relatively new specialization of Distributed Artificial Intelligence (DAI). Multi-agent technology has already been applied in many engineering applications, of which manufacturing is a leading one. A brief introduction to multi agent system and its relevance in meeting the challenges associated with the dynamic behaviour of the AGV system is given below for better insight of the problem.

### *1.1 Multi Agent Systems*

A Multi Agent System (MAS) is a collection of agents which interact to perform a task which is beyond the capability of individual agents. An agent is a computer system that is situated in some environment, and is capable of autonomous action in this environment in order to meet its design objectives (Wooldridge, 2001). Agents are also defined as autonomous, computational entities that can be viewed as perceiving their environment through sensors and acting upon their environment through effectors (Weiss, 1999). From the above definitions, it follows that the common attributes that an agent should posses are autonomy, cooperativeness, reactivity and proactive behaviour. Based on the characteristics of agent architectures, agents are classified as: i) logic based agents, ii) reactive agents, iii) Belief-Desire-Intention (BDI) agents, and iv) layered agents (Weiss, 1999). Java has been reported as a programming language for MAS development in most of the previously reported works due to its built-in support for multi threads and other useful features. Building agents from scratch is a time consuming and tedious task, therefore, researchers have developed several platforms for fast development of MASs, and most of them are freely available (Luck et al., 2003). However, these platforms were highly specific to the application and rarely of use to others. Moreover, agents developed on one software platform were difficult to interoperate with agents built on another software platform. This forced the agent community to move towards preparing standards for agent interoperability and MAS development. One such effort is specifications from Foundation for Intelligent Physical Agents (FIPA) (FIPA, 2012), an IEEE computer society standards that promotes agent-based technology and the interoperability of its standards with other technologies. Java Agent DEvelopment framework (JADE™) (JADE, 2012), a non-commercial software developed at Telecom Italia Lab (TILab), Italy, is one of the leading platforms that support FIPA specifications.

*1.2 AGV System*

A dynamic market with high product variety demands more agility in the manufacturing system for which computer controlled machines are being employed on the shop floor with flexible material handling system. Automated Guided Vehicle (AGV) is a widely used material transporting equipment in modern automated material handling systems due to its routing flexibility. AGVs which move along a guidepath are more common in industries. Control of AGV system involves dispatching and routing of multi-AGVs and avoiding collisions and deadlocks along the guidepaths. As AGVs behave in a highly concurrent and dynamic fashion, distributed or decentralized approach is suitable for control of AGV system. Thus, the problem of control of AGV system should be decomposed in such a way that the decision-making is performed by many simple, autonomous and cooperative entities rather than following a predetermined plan. The distributed intelligent control with multi-agent approach provides the advantages of adaptability, ease of upgradeability and maintenance and capable to handle dynamic behaviour of the system. On the other hand, the disadvantages of this approach are that global optima cannot be guaranteed and predictions of the system's behaviour can only be made at the aggregate level (Deen, 2003). Building a simulation model with agents gives an opportunity to develop the model that is more close to the real system. In addition, agent-based simulation provides a better opportunity for analyzing and decision making in the AGV system.

## 2. Review of Related Work

Wide-ranging details of agent-based technology and MASs were addressed in (Weiss, 1999; Wooldridge, 2001). The development of agent-based technology in different applications and its past, present and future developments were reported in the roadmap of agent technology (Luck et al., 2003). PROSA (Product-Resource-Order-Staff Architecture) is a well-known architecture for manufacturing control which originated from holonic manufacturing system (HMS) research project (Brussel et al., 1998). PROSA defines three types of essential (basic) agents: Product-Agents, Resource-Agents, and Order-Agents. Product-Agents take care of product and process related technological aspects, Resource-Agents take care of resource aspects such as maximizing its capacity and Order-Agents take care of logistical concerns about customer demands and due dates. A fourth type of agent is the Staff-Agent, an optional agent to assist the basic agents for performing their task optimally.

Literature on the AGV system is briefly presented here. Control of an AGV system is greatly influenced by the type of layout that is employed in the AGV system. It is easier to control an AGV system in a single loop without intersections than in a conventional bidirectional network. (Wallace, 2001) used AI to develop an AGV controller for large complex guidepaths. The article reported that the agents were used as traffic managers, which facilitated the AGVs to access points and segments of the guidepath. An important simulation tool called Manufacturing Agent Simulation Tool (MAST) was developed for simulation of material handling system by (Vrba, 2003). A multi-agent based architecture for AGV system was proposed by

(Farahvash & Boucher, 2004) in which communication was achieved by using a relational database (blackboard system). A complete overview of Agent-based manufacturing was presented by (Paolucci & Sacile, 2005). A detailed review of design and control of AGV system was reported by (Vis, 2006). A review of agent based systems for AGV system was reported by (Komma et al., 2005)

Survey and comparison of usage of ontologies in the area of manufacturing and the semantic web is made by (Obitko & Mařík, 2002). One of the main differences is that the ontologies in the manufacturing area tend to be standardized and fixed in order to achieve stable, guaranteed behaviour. On the other hand, in the area of the semantic web, it is expected that nobody can guarantee anything — ontologies can be changing and inconsistent, and agents that work with them are expected to be able to handle them, though they may not always get the required results. Despite the differences, these areas can take inspiration from the each other. (Obitko & Mařík, 2002) concluded that more attention should be paid to the problem of semantic interoperability even in a domain such as manufacturing environment where standardized solutions are usually anticipated.

Since research is being carried out aggressively in agent technology, a vast literature is available on the development of concepts, architectures, interaction techniques, general approaches to the analysis and specification of MAS. However, these contributions, which are sometimes formal but often informal, have been quite fragmented, without any clear way of "putting it all together" and thus completely inaccessible to practitioners (Bordini et al., 2005). Due to these facts, many of the published books are just a collection of several research articles.

In the existing literature, development details of agent-based simulation software for shop floor control focusing on AGV systems were not reported. The development details of agent-based simulation software, which have not been reported in the existing literature, are modelling of different manufacturing agents, communication protocols and AGV domain-specific ontology. Details of the agent-based software development are helpful to researchers for building agent-based simulators for different sections of manufacturing domain. Therefore, in this Chapter, development of an Agent-Based Shop Floor Simulator (ABSFSim), an agent-based shop floor simulator focusing on AGV system is presented. ABSFSim has been developed on JADE platform. Development of ABSFSim involves the following interdependent steps: i) development of semi-formal domain-specific ontology for AGV systems; ii) development of two hybrid communication protocols; iii) development of a novel synchronization mechanism for synchronizing agents with the simulation clock; and iv) modelling of different agent types in AGV system. A semi-formal ontology for an AGV system is developed on Protégé (Protégé, 2012) ontology editor. Two hybrid communication protocols, namely, Part-Machine-AGV (PMA) hybrid contract net protocol and AGV-Node-Segment (ANS) hybrid request protocols have been proposed for effective communication and decision making. A novel synchronization mechanism has been proposed and implemented in ABSFSim for synchronizing manufacturing agents with the simulation clock and subsequently enabling the discrete event simulation. Manufacturing agents are modelled on JADE reactive architecture to make the modelling simple and suitable for discrete event simulation.

ABSFSim is verified for its correct working with sample manufacturing systems. A new part-initiated continuous dispatching approach, HCNBD, is proposed and its effect is studied on AGV performance.

## 3. Ontology Development for AGV System

Before introducing the domain-specific ontology for AGV system, agent types identified in AGV system for ABSFSim, based on physical decomposition and functional requirements, are presented here. Agents in manufacturing domain are referred here as manufacturing agents. The list of manufacturing agents and their intended roles in ABSFSim is given in Table 1. Manufacturing agents in ABSFSim are modelled on JADE reactive architecture (Komma et al., 2011).

| Agent | Purpose |
|---|---|
| AGV-agent | Represents a typical AGV |
| Arrival-queue-agent | Represents a system arrival queue |
| Departure-agent | Represents the queue at the system departure place of parts |
| Machine-agent | Represents a typical machine with its input and output buffers on the shop floor |
| Node-agent | Represents a node or control point on the AGV guidepath |
| Part-agent | Represents a typical job or part on the shop floor |
| Part-generator-agent | Represents an agent that facilitates the arrival of part-agents at the shop floor |
| Segment-agent | Represents a segment on the AGV guidepath |

Tab. 1. Manufacturing agents and their purposes

During agent communication, agents exchange messages among the participants to achieve rational effect. A message on JADE platform is represented as per FIPA-ACL (FIPA-Agent Communication Language), which contains a number of slots for storing different information blocks of the message. Some of these slots are Communication-act or Performative, Sender, Receiver, Content, Content Language, Ontology, Interaction Protocol and Conversation ID. In each interaction protocol, a specific sequence of messages is exchanged between the agents to complete the conversation and achieve a rational effect. The content of a message is syntactically represented in the developed simulator as per FIPA-SL (Semantic Language). To understand the content of a message both sending and receiving agents should share the common ontology. A semi-formal ontology for an AGV system is developed on Protégé (Protégé, 2012). The developed ontology is translated into JADE/FIPA compatible Java classes with the help of ontology Beangenerator, a plug-in for Protégé. The ontology defined for the manufacturing agents facilitates in simulation of the shop floor and for effective exchange of information among manufacturing agents for decision making such as routing and dispatching of AGVs.

JADE compliant ontology consists of concepts, predicates (relations) and agent actions. Concepts are the 'objects' in the domain which are represented as classes in Protégé editor. An 'object' carries some attributes, which are represented as slots in the classes. Concepts are not used directly as the content of a message, but in combination with predicates. The actions that an agent can perform are also a special type of concepts, which are represented as agent-actions, for example Transport, Operation. Predicates represent the relations between the concepts in the domain, for example, 'PartOnAgv' represents "a part is loaded on AGV". The developed AGV system ontology covers most of the domain and task specific ontological terms. However, it is the application specific ontology and can be modified as per the specific requirement of the application. A subset of these ontological classes was used in the communication of the manufacturing agents for simulation of AGV system. For more details of semiformal domain-specific ontology for shop floor focusing on AGVS interested readers may refer to (Komma et al., 2012).

## 4. Development of Hybrid Communication Protocols

Messages are exchanged among agents in a sequence according to an Interaction Protocol (IP) to achieve a rational effect. Most of the interaction protocols, used in ABSFSim are the FIPA-Interaction Protocols which are basic protocols applicable to systems in general. These basic protocols are not sufficient for decision making in manufacturing system. Therefore, in this work, two hybrid communication protocols have been proposed and implemented in ABSFSim. The hybrid communication protocols, namely, Part-Machine-AGV (PMA) hybrid contract-net protocol and AGV-Node-Segment (ANS) hybrid request protocol were introduced by (Komma et al. 2007) and formal representation of these protocols is presented here.

### *4.1 Part-Machine-AGV (PMA) Hybrid Contract Net Protocol*

PMA hybrid contract-net protocol (CNP) is useful for a part to select simultaneously the best machine and AGV combinations for the next stage operation. Formal representation of the PMA hybrid contract-net protocol is shown as an UML Sequence Diagram in Fig. 1. The protocol is built by integrating two basic FIPA-CNPs. A part initiates the protocol by sending call for proposal (CFP) messages to a set of 'M' alternate machines based on their capability of performing the next stage operation. In response, the part receives 'P' proposals and 'R' refusals such that P + R = M, from the participating machines. For each received proposal, the part initiates a CNP by calling a CFP to 'n' potential AGVs for transporting the part from its current location to the proposed machine. In response to each CFP, the part receives either proposals or refusals from the AGVs. Based on the selection criteria, the part finalizes a specific machine and AGV combination for the required operation and accepts their proposals while rejecting all other proposals. The working of the implemented PMA hybrid CNP in ABSFSim was tested with parts having two alternate part processing routes and found satisfactory working. However, for the manufacturing systems considered in this work each part has one processing route.

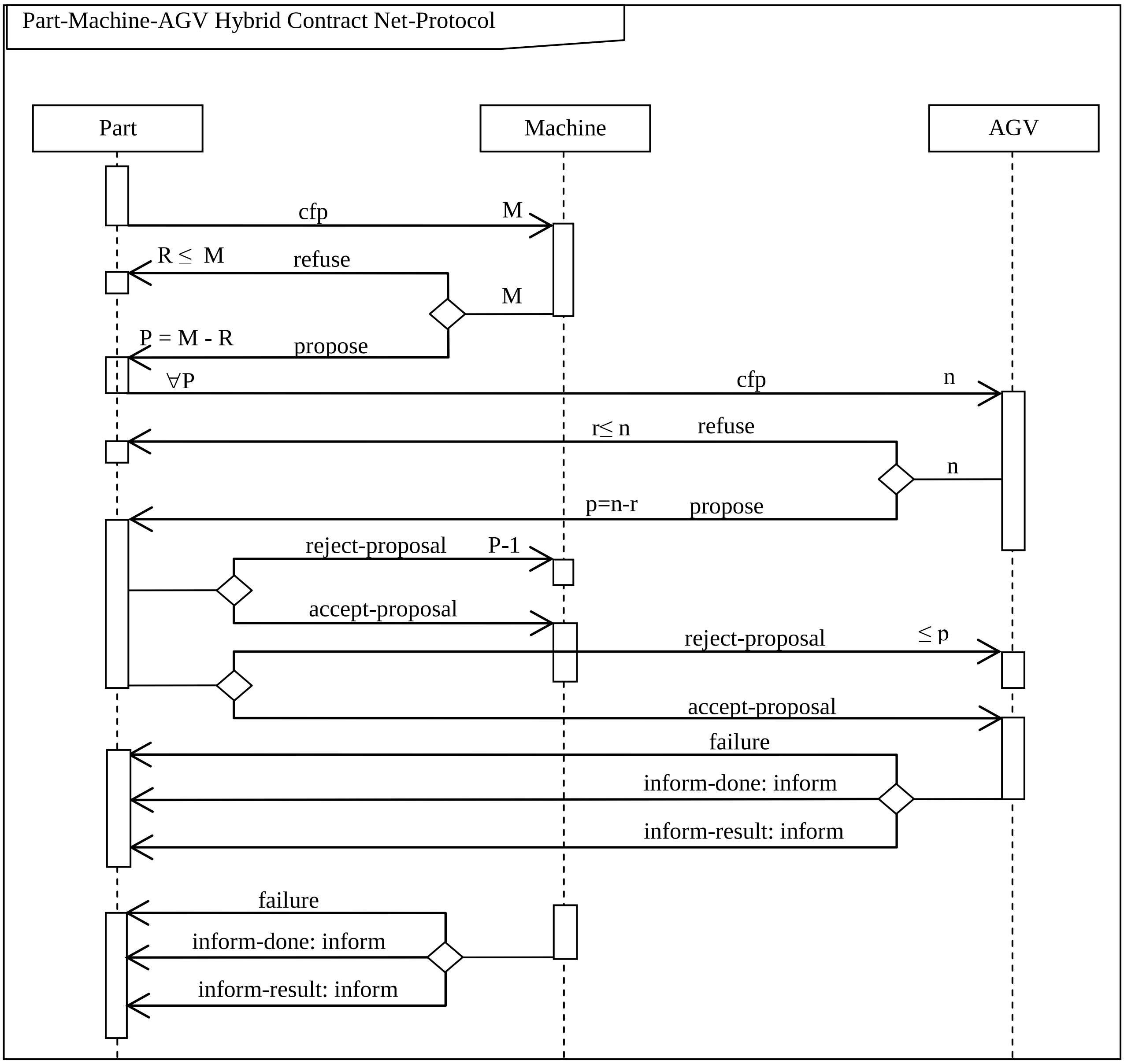


Fig. 1. UML sequence diagram of PMA hybrid contract net-protocol

*4.2 AGV- Node- Segment (ANS) Hybrid Request Protocol*

AGV agent initiates ANS hybrid request protocol to seek permission to move along the selected segment through a node (control point). The formal representation of the ANS hybrid request protocol is shown as a UML-Sequence Diagram in Fig. 2. The ANS hybrid protocol is built from the combination of two FIPA-Request protocols. When AGV is in siding buffer (vehicle buffer) of a segment or in parking at a node and it wants to travel to a node that is directly connected to the current node; in other words, the AGV wants passing through the current node to travel along an adjacent segment. The AGV agent requests the node agent at which it is currently standing for permission to move to the target node and the node agent forwards the request to the concerned segment agent. Based on the intended direction of the AGV and the status of the segment, the segment may accept or reject the forwarded request of the node. The same information will be communicated back to the AGV by the node. The protocol is simple and effective for a single-lane guidepath layout. However, the protocol is more effective when it is used in multi-lane paths where two or more parallel segments connect any two adjacent nodes. In this case, the participating node can identify and direct the AGV to the least congested segment.

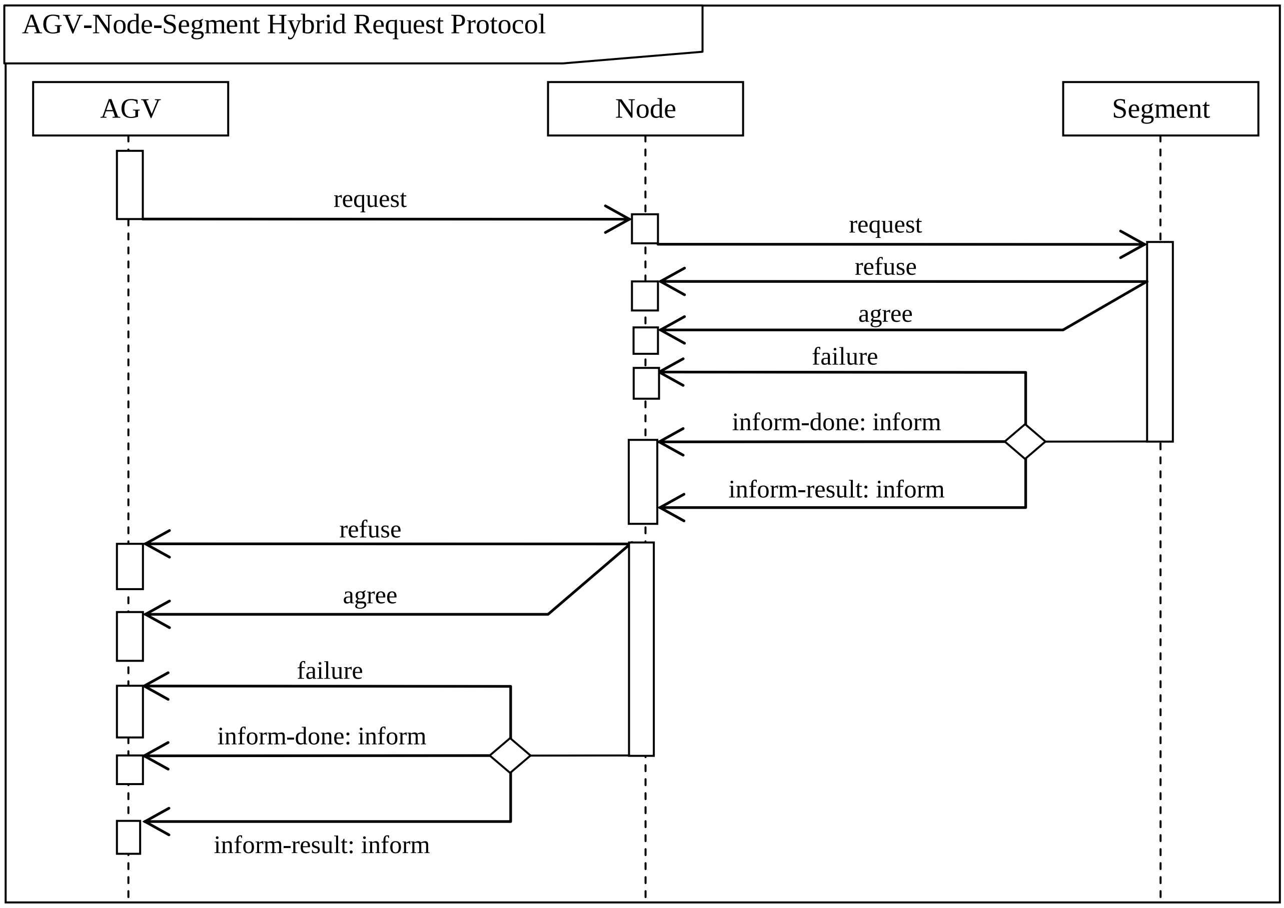

Fig. 2. UML sequence diagram of ANS hybrid request protocol

## 5. Modelling of Manufacturing Agents

Agents in ABSFSim are modelled on JADE reactive architecture (Komma et al. 2011). Proactive behavior for agents is realized from their decisions that enhance the performance of the agent. A JADE agent contains a set of '*Behaviour'* objects which can be dynamically added or removed from the agent's behaviour list in order to handle the messages received from the other agents. JADE '*FSMBehaviour'* (Finite State Machine Behaviour) and '*SimpleBehaviour'* (especially with multiple steps) classes play vital role in building the behaviour objects for most of the manufacturing agents. Due to the intricacy involved in modelling part and AGV agents compared to other manufacturing agents, details of these two agents are presented here.

### *5.1 Part Agent*

A part agent is responsible for the activities of part on the shop floor. Each part agent contains several behaviours and handles the messages received from other agents. One such behaviour is "*PartBehaviouralFSM*" which is represented as State Machine Diagram (SMD) of UML in Fig. 3, which shows the sequence of states through which the part agent passes during its life cycle. Many of these finite states are registered with multi-step '*SimpleBehaviour'* objects. Part agents are started by a part-generator agent by sending a suitable request message to AMS agent to simulate the part arrivals at different arrival times. When a part arrives at system input queue of the shop floor, the part agent requests the input queue agent with FIPA-Request interaction protocol to add it to the queue. Queue agent adds the part to queue and informs its position in queue. If the part is not in the first position of the queue, part subscribes the queue agent with FIPA-Subscription interaction protocol to receive notifications, whenever there is a change in the part position. Once the part reaches the first position in the queue, it cancels its subscription with the queue by sending a cancel message and reaches ‘wait for negotiation’ state. This state is registered with a multi-step ‘*SimpleBehaviour’*. In the first step of this behaviour, negotiation behaviour is added to the behaviour list that negotiates with machines and AGVs. Part agent negotiates with machines and AGVs with PMA hybrid CNP. As a result of negotiation, an AGV arrives at the queue for picking up the part. On receiving the information of AGV arrival at the queue, the part agent requests the queue agent to remove its identity from the queue. Subsequently, part requests the AGV to take it on-board (load). Once the part is loaded onto the AGV, the part agent enters in a state called ‘Travel on AGV’. When the AGV reaches the delivery node of the part, it informs the part. As a result, the part agent requests the AGV to unload it from the AGV and subsequently requests the machine agent at the delivery node with FIPA-Request interaction protocol to place it on the machine, or in input buffer (IB), if the machine is busy. FIPA-Subscribe interaction protocol is used by the part to get information regarding its updated status at the machine. When the required operation on the machine is completed, the machine informs the same to the part. Thus, the part agent updates the operation stage. If space is available in output buffer (OB), part is advanced to OB otherwise it will wait on the machine itself. Part behavioural state is then set back to “wait for negotiation” and the sequence of states is revisited by the part until it reaches the departure location.

*5.2 AGV Agent*

An AGV is initially at its home node and is in “Empty Idle” state. When a transportation request is received form a part, the AGV starts from its current node to move towards the pickup node, travelling along several segments and nodes. While travelling along the path, it passes through states of “Empty Wait”, while it waits for any traffic clearance in a siding before entering a segment and “Empty Move”, while it moves along a segment. To get permission to move along a segment from its current position, AGV agent uses ANS hybrid request protocol.

When the AGV reaches the pickup node, it enters into the parking of the node and informs about its arrival to the part. The AGV performs loading operation on receiving suitable request from the part agent. After completion of the loading, the loaded AGV starts moving along a selected path until it reaches the delivery node of the part. On reaching the delivery node, the AGV again enters into parking at the resource and informs the part. The AGV unloads the part on the latter’s request. This sequence of steps is repeated for different transportation assignments. An AGV behaviour that consists of finite states through which the AGV passes during its operation is represented as SMD of UML and is shown in Fig. 2. More details about the agent modelling were reported by (Komma et al. 2011).

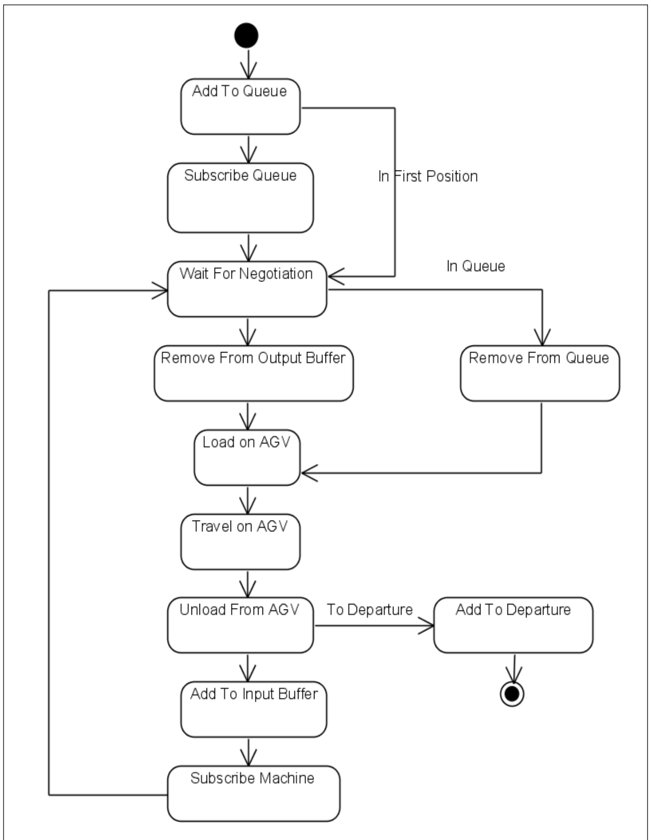


Fig. 3. State Machine Diagram of Part Behavioural FSM

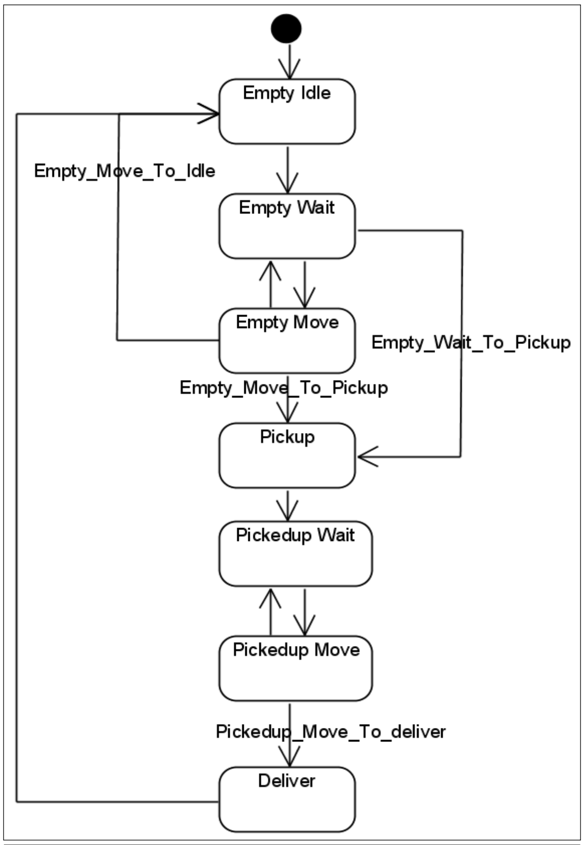


Fig. 4. State Machine Diagram of AGV Behaviour

## 6. Synchronization of Manufacturing Agents

As agents are autonomous, they run concurrently in an asynchronous manner. Since agents are used as entities in simulation model, it leads to Parallel Discrete Event Simulation (PDES). Synchronization is an important aspect that must be dealt in developing PDES (Paolucci and Sacile, 2005). Historically, two principal approaches were suggested for advancing the simulation clock in a conventional simulation, namely, next-event time advance and fixed-increment time advance (Law and Kelton, 1991). ABSFSim uses a novel synchronization mechanism for synchronizing the agents to enable agent-based discrete event simulation. In the implemented synchronization mechanism, a JADE agent is modeled that maintains shared variables of global virtual time (GVT) and a global event list. Since the agent shares its attributes which is against the basic property of an agent, it is referred here as “Timer-Thread”. Each agent, that contains a Java-thread, maintains a local virtual time (LVT) and a local event list. The purpose of the global event list is to provide a shared memory for agents to post their partial list of future events and assist them for synchronized execution of the events. The Timer-Thread advances the GVT by fixed and equal increment of time with large time steps, which is similar to fixed-increment time advance mechanism. Within the time step (time between the start of two consecutive time steps), the GVT is advanced as next-event time advance mechanism by the corresponding agents. Thus, both the approaches of simulation were combined in the present work for effective synchronization. The working of this mechanism is briefly described here. During initialization of an agent on JADE platform, the agent registers with the Timer-Thread. Whenever Timer-Thread advances GVT, it notifies all the registered agents by sending a message to all of them and the agents update their LVT to GVT. Subsequently, the registered agents post their individual events that must be executed before the upper limit of the time step into the global event list. While posting events by the agents, the events are automatically sorted with the help of a user-defined comparator that sorts the events in ascending order of their time-stamp. The events of equal time-stamp are resolved based on unique event numbers, which are allotted during the creation of the event objects. For the events of same time-stamp, higher priority is given to the events with the smaller event numbers. Once all the registered agents posted their events, they are allowed to execute the events in sequence. When an agent started executing an event, the agent is authorized to update the GVT. All other agents suitably update their LVT on their regular checking of GVT with its specialized behavior ("*SimulatorBehavior*"). In the current implementation of ABSFSim, the agent events with same time-stamp of different agents are allowed for parallel execution by the respective agents. Events of different time stamps are executed in sequence. The size of time-step increment of the GVT and the number of agents registered with the Timer-Thread are two critical parameters of the proposed synchronization mechanism. Reduced time-step size increases the number of messages being exchanged within a specific time period. If the time-step size is increased, the size of the global event list increases.

While an agent is executing an event, it may interact with other agents to complete the event logic. This implies that agents are not blocked from their execution process for the purpose of current event execution; they are free to participate in the communication and retain the nature of parallel simulation. The process of registering with the Timer-Thread and receiving information from it on GVT time-step increment resembles FIPA-Subscription-Protocol. In the following section, implementation of a sample manufacturing system in ABSFSim is discussed.

## 7. Implementation of a sample manufacturing system in ABSFSim

AGV system model implementation in ABSFSim needs, in addition to the agent modeling and communication, AGV path finding between any two nodes on the layout, AGV dispatching and routing. In the present work, a recursive breadth-first search algorithm is implemented that effectively determines all possible paths between a source and destination nodes of a typical layout consisting of loops (cyclic graphs). The algorithm works well for unidirectional, bidirectional or mixed uni/bidirectional layout. The dispatching of vehicles and loads is discussed here.

### *7.1 Dispatching of Vehicles and Loads*

In literature, numerous dispatching rules are available, which are vehicle initiated (or vehicle dispatching), i.e. deciding which load must be selected for the new task, and workstation initiated (or load dispatching), i.e. deciding which idle AGV must be selected for a load at the workstation. Most of the dispatching rules of vehicles are activated only when the AGV finishes its current task, such as longest waiting entity or nearest entity. In load dispatching, one of the idle vehicles will be selected such as nearest vehicle, random vehicle, if none of the vehicles are idle then the loads will be picked-up according to the vehicle dispatching rule in future.

In ABSFSim, a part or load initiated continuous dispatching, hybrid contract net based dispatching (HCNBD), is implemented as a part of PMA hybrid CNP. Based on the proposal from a machine, part sends a CFP with the details of pickup and delivery locations to all AGVs. Irrespective of the state of the AGV (need not be in idle state), all participating AGVs processes CFP and submit their proposals. For preparing the proposal by an AGV, it calculates the start node and task times for the task (start, pickup and delivery times), based on the list of pending tasks of the AGV. The pickup and delivery times are calculated while considering the shortest distance between the corresponding locations and the AGV speed. In the call for proposal, the part also specifies the preferable time for delivery based on the state of the destination machine. In the current implementation, the AGV that proposes the minimum pickup time is selected by the part. The proposed HCNBD not only helps the part to minimize its waiting time for the AGV by selecting the best possible AGV, but also balances the workload among the AGVs. Routing of AGV is the first shortest path (in the look up table) between any two nodes on the layout.

*7.2 Agent-Based Simulation of a Sample Manufacturing System*

Simulation model of a small sample manufacturing system was implemented in ABSFSim and almost equivalent simulation model was implemented in a conventional simulator, ProModel®. Output of the both the models were compared for the same total simulation time (2000 time units) and found that the outputs were in good agreement. Thus, verifying the proposer working of the ABSFSim. Details of verification and testing of ABSFSim were reported by (Komma et al 2011). Another sample working-level manufacturing system is considered here for agent-based simulation in ABSFSim and to study the effect of the proposed vehicle dispatching, HCNBD approach, over two commonly used AGV dispatching rules (viz. longest waiting entity (LWE) and nearest entity (NE)).

The considered sample working-level manufacturing is a slightly modified configuration of the manufacturing system considered by (Mahadevan & Narendran 1994). Graphical representation of the sample working-level manufacturing system is shown in Figure 6. The manufacturing system has eight machines, three AGVs and five part types. The guidepath layout has both unidirectional and bi-directional segments. The agent-based simulation model of the manufacturing system is referred here as HCNBD model. Input details of the HCNBD model is given in Table 2.

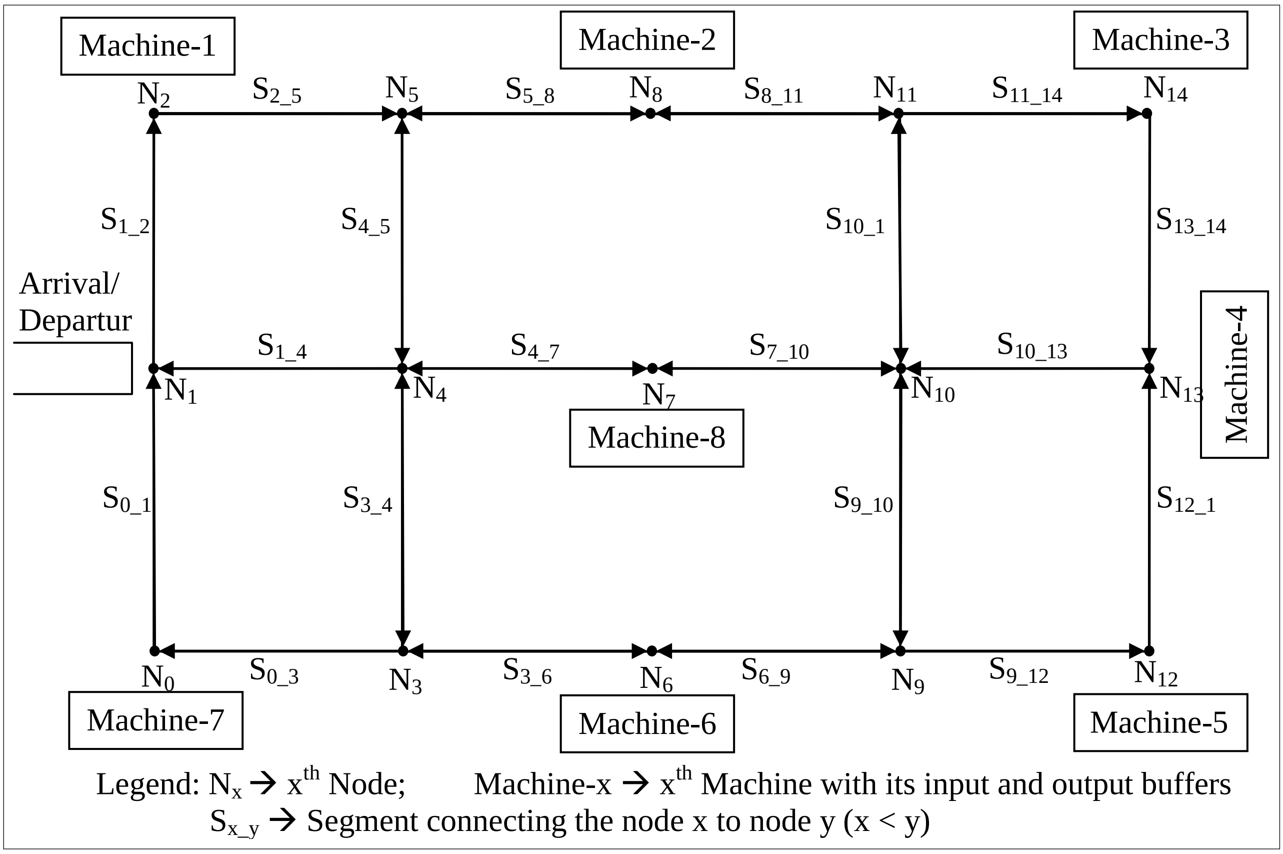


Fig. 6. A Sample working-level manufactruing system

| | |
|---|---|
| *1. Layout information* | |
| - Number of nodes = 15<br>- Number of bidirectional segments = 9 and unidirectional segments = 11<br>- Siding capacity (vehicle buffer) of each segment = 2 | |
| *2. AGVs information* | |
| - Number of AGVs = 3<br>- Time required to cross each segment is 1.5 minutes<br>- Pickup and delivery tasks each takes 0.5 minute<br>- After delivering a load, if AGV does not have any pending tasks, it waits at its location until it receives a new transportation request<br>- HCNBD approach is used for dispatching of loads to AGVs<br>- AGVs use the Fixed Shortest Path (the first shortest path in the lookup table) to travel between any two specific nodes<br>- All AGVs are available at $N_1$ at the start of the simulation | |
| *3. Workstations information* | |
| - Number of machines = 8; IB and OB capacity of each machine = 4<br>- Machines schedule the parts processing on FCFS basis | |
| *4. Parts information* | |
| Part type | Operation sequence - Machine (Operation time in minutes) |
| A | M1(34) - M3(32 ) - M5(26) - M4(22) - M6(26) |

| | |
|---|---|
| B | M2(24) - M8(31) - M4(30) - M7(31) - M3(26) - M5(18) |
| C | M1(23) - M8(29) - M3(25) - M5(28) - M6(30) - M4(21) - M7(24) |
| D | M2(23) - M8(24) - M5(17) - M3(26) - M6(29) - M7(21) |
| E | M1(28) - M2(21) - M4(24) - M6(27) - M5(30) |
| *5. Part Arrival information* | |
| - Parts arrive in a cyclic pattern (i.e. A-B-C-D-E-A-B-C…)<br>- Arrivals occur at a fixed interval of 27 minutes. | |
| *6. Simulation information* | |
| - Total length of simulation = 40,000 minutes<br>- Warm-up time = 2,500 minutes<br>*Note*: One time unit is considered as ½ minute. Thus, the total length of simulation and warm-up time will be 80000 and 5000 time units respectively. | |

Tab. 2. Input data for HCNBD model

On simulating the HCNBD model on ABSFSim with the considered input data, the following observational and time based outputs were recorded at the end of the simulation. i) total number of parts finished are 1389, (comprises part types A – 278, B – 277, C – 278, D – 278 and E – 278); ii) about 80 parts were finished during the warm-up period; iii) average make-span time of parts was 261.87 minutes; iv) average number of parts in the system (work in progress) is 9.7; and v) number of deliveries made by $AGV_1$, $AGV_2$ and $AGV_3$ are 3207, 3183 and 3061 respectively. The average utilization of AGVs was 72.36%, which includes the effective and ineffective operation times of AGVs. Based on less average waiting time of parts in arrival queue (i.e. 2.9 minutes in average make-span time of 261.87 minutes), medium utilizations of machines (in the range of 50 to 88 %) and AGVs (72.36 %), it can be stated that the system is not overloaded.

To test the proper working of the agent-based simulation model on ABSFSim for the extended period of simulation, a nearly equivalent simulation model, say LWE model (because AGV initiated dispatching uses longest-waiting entity), was built in ProModel® environment. Most of the input data and assumptions including the arrival pattern and inter-arrival times were same for both the models. The assumptions of LWE model, which are different from HCNBD model are: i) restriction on head-on collisions of AGVs was not imposed, therefore, waiting times of AGVs along their travel path are neglected, and ii) for the AGV initiated dispatching, the longest-waiting entity is selected for transportation, whereas for the workstation initiated dispatching, the nearest idle AGV is selected. As both HCNBD model and LWE model are not overloaded with parts, it is expected that the performance of AGVs will not affect the throughput of the system significantly. This is because AGVs have redundant capacity to meet all the transportation requests in time. Total number of finished parts in LWE was also 1389. Since the number and sequence of part arrivals are same in both HCNBD model (i.e. agent-based simulation model) and LWE model (conventional simulation model), the models are compared in terms of the percentages of time spent by the machines in their idle and operation states in Table 3. It is evident from the table that the percentage of times spent in idle and operation states by the machines in both the models are almost matching, which verifies HCNBD model and validates the proper working of ABSFSim for longer simulation run. However, the performance of AGVs in HCNBD and LWE shall differ due to the use of different AGV dispatching approaches.

| Machine | Machine states | | | |
|---|---|---|---|---|
| | Idle (HCNBD) | Idle (LWE) | Operation (HCNBD) | Operation (LWE) |
| Machine-1 | 37.02% | 37.04% | 62.98% | 62.96% |
| Machine-2 | 49.65% | 49.68% | 50.35% | 50.32% |
| Machine-3 | 19.22% | 19.34% | 80.78% | 80.66% |
| Machine-4 | 28.16% | 28.15% | 71.84% | 71.85% |
| Machine-5 | 11.86% | 11.92% | 88.14% | 88.08% |
| Machine-6 | 17.07% | 17.09% | 82.93% | 82.91% |
| Machine-7 | 43.72% | 43.71% | 56.28% | 56.29% |
| Machine-8 | 37.73% | 37.78% | 62.27% | 62.22% |

Tab. 3. Time spent in different states by machines in HCNBD and LWE models

### *7.3 Effect of the Proposed HCNBD Approach on AGV Performance*

In the HCNBD approach, part selects an AGV that proposes the minimum pickup time to reduce waiting time of the part. This also minimizes the empty travel time of AGV while meeting the schedules of the committed transportation tasks. In general, the nearest-entity (NE) selection for AGV dispatching is expected to have minimum empty travel time. Therefore, another model with NE as dispatching rule for AGVs was simulated in ProModel, say the model is termed as NE model. During the agent-based simulation of HCNBD model, the ABSFSim continuously records the time spent by AGVs in their different states, execution details of transportation tasks such as start time, pickup time and deliver time of the tasks. The effective operation time of an AGV is obtained as the sum of times spent in Pickup, Loaded-Move and Deliver states. Ineffective operation time of an AGV is the sum of times spent in Empty-Move, Empty-Wait and Loaded-Wait states; however, the Empty-Move time is predominant over the waiting times. For the LWE and NE models, ProModel® reported the percentages of time spent by the AGVs in In-Use state (equivalent to the sum of Pickup, Loaded-Move, Loaded-Wait and Deliver states), Travel-To-Use (equivalent to the sum of Empty-Move and Empty-Wait states) and Idle (equivalent to Empty-Idle) states. To understand the relative performance of HCNBD approach, the percentages of time spent by the AGVs in loaded travel (equivalent to In-Use state of ProModel), empty travel (equivalent to Travel-To-Use state of ProModel) and idle states during the simulation of HCNBD, LWE and NE models are compared in Table 4. Average percentages of time spent by AGVs in different states in HCNBD, LWE and NE approaches are compared in Fig. 4. For the same throughput of the parts, it can be observed form the figure that the average utilization of the AGVs in HCNBD model (i.e. 72.36%) is less than LWE model (i.e. 88.45%) and NE model (i.e. 83.62%). Furthermore, it is interesting to observe that the ratio of the average empty travel time to loaded travel time of AGVs is 0.51, 0.85 and 0.75 for HCNBD, LWE and NE models respectively. For the same throughput of the system, the HCNBD approach reduced the average ineffective operation time of AGVs by 40 % and 32 % of LWE and NE dispatching rules.

| Model – AGV State | AGV1 | AGV2 | AGV3 | Average |
|---|---|---|---|---|
| HCNBD – Loaded Travel | 49.1% | 47.7% | 46.8% | 47.8% |
| LWE – Loaded Travel | 48.4% | 47.6% | 46.9% | 47.7% |
| NE – Loaded Travel | 48.2% | 48.0% | 46.8% | 47.6% |
| HCNBD – Empty Travel | 24.7% | 25.2% | 23.7% | 24.5% |
| LWE – Empty Travel | 40.9% | 40.9% | 40.6% | 40.8% |
| NE – Empty Travel | 36.4% | 36.0% | 35.6% | 36.0% |
| HCNBD – Idle | 26.2% | 27.1% | 29.6% | 27.6% |
| LWE – Idle | 10.7% | 11.5% | 12.4% | 11.6% |
| NE – Idle | 15.5% | 16.0% | 17.6% | 16.4% |

Tab. 4. Percentages of time spent by the AGVs in different states

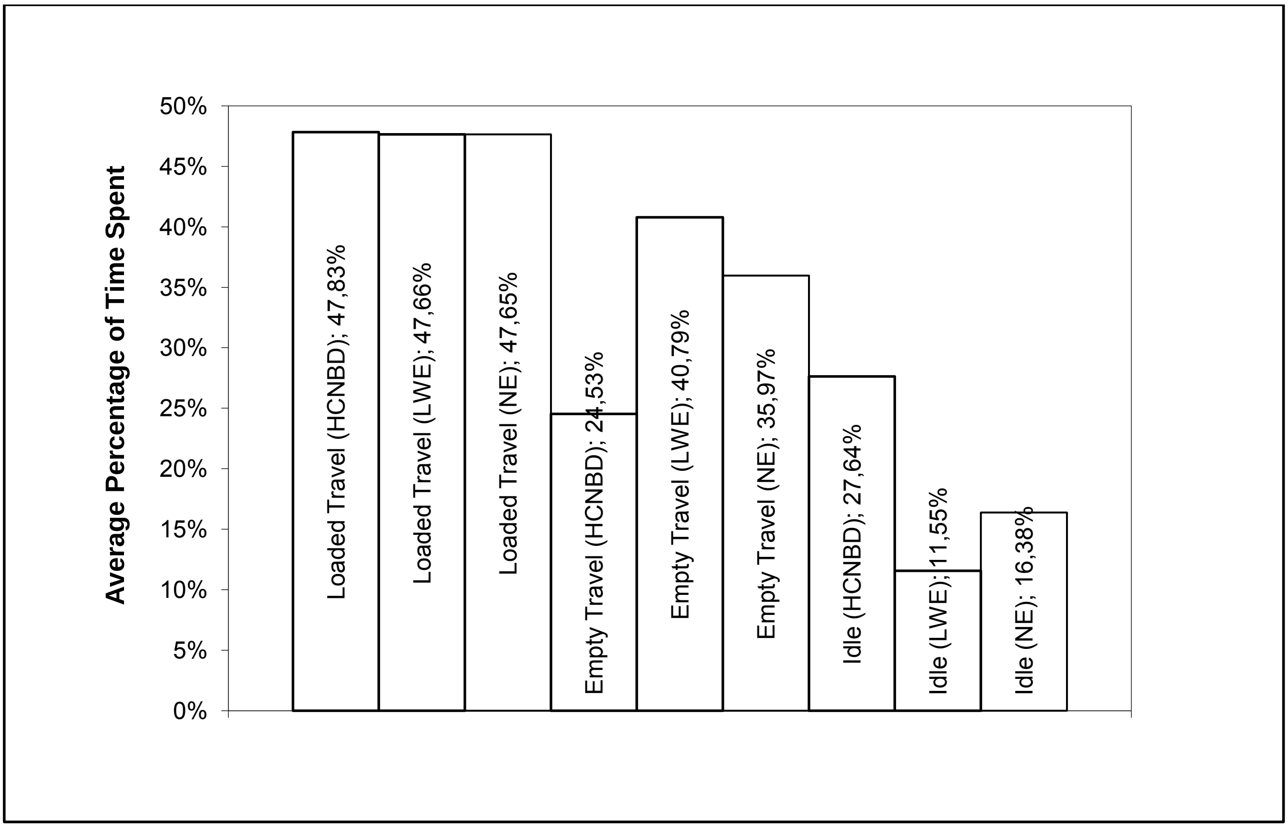


Fig. 7. Comparison of average percentage of time spent by AGVs in different states of HCNBD, LWE and NE models

## 8. Conclusions

In this chapter, ABSFSim, an agent-based shop floor simulator focusing on AGV system is introduced. ABSFSim is developed on JADE reactive architecture, which suits well for the agent-based discrete event simulation. The steps involved in the development of ABSFSim are development of a semi-formal domain-specific ontology for AGV system, development of agent communication protocols for effective information exchange, development of a novel synchronization mechanism for synchronizing agents with the simulation clock and modelling of different agent types. The developed ABSFSim was carefully debugged and verified its proper working with a sample manufacturing system by comparing ABSFSim output with an almost equivalent model developed in ProModel® environment. In ABSFSim, a part initiated continuous dispatching approach, HCNBD, has been implemented with the help of PMA hybrid CNP protocol. The performance of AGVs is compared for HCNBD, LWE and NE dispatching approaches. It has been observed that for the same throughput of the parts, average utilization of the AGVs in HCNBD model (i.e. 72.36%) is less than LWE model (i.e. 88.45%) and NE model (i.e. 83.62%). Furthermore, it is interesting to observe that the ratio of the average empty travel time to loaded travel time of AGVs is 0.51, 0.85 and 0.75 for HCNBD, LWE and NE models respectively. For the same throughput of the system, the HCNBD approach reduced the average ineffective operation time of AGVs by 40 % and 32 % of LWE and NE dispatching rules. Agent-based modelling and simulation of manufacturing system facilitates in developing simulation models that are close to the real system and provides higher flexibility for real-time decision making.

In future, the developed ABSFSim will be used for detailed analysis of AGV system performance on different dynamic AGV path selection strategies. ABSFSim will be extended for AGV systems with multiple-load AGVs and multi-lane paths, where dispatching and routing of AGVs is entirely different. On the other hand, ABSFSim shall be extended for integration of AGV and machine scheduling to improve the performance manufacturing system.

## 9. References

Bordini, R. H.; Dastani, M.; Dix, J. & Seghrouchni, A. E. F. (eds.). (2005). *Multi-Agent Programming Languages, Platforms and Applications*, Springer Science+Business Media, Inc., ISBN-10: 0-387-24568-5, New York, USA.

Brussel, H. V.; Wyns, J.; Valckenaers, P.; Bongaerts, L. & Peeters, P. (1998). Reference architecture for holonic manufacturing systems: PROSA, *Computers in Industry*, Vol. 37, No. 3, (November 1998), pp. 255–274, ISSN: 0166-3615

Deen, S. M. (Ed.). (2003). *Agent Based Manufacturing – Advances in the Holonic Approach*, Springer- Verlag, ISBN: 978-3-540-44069-7, Heidelberg, Germany.

Farahvash, P. & Boucher, T. O. (2004). A multi-agent architecture for control of AGV systems, *Robotics and Computer-Integrated Manufacturing*, Vol. 20, No. 6, (December 2004), pp. 473–483, ISSN: 0736-5845

FIPA. (2012). Foundation for Intelligent Physical Agents, *Available from:* http://www.fipa.org, *Accessed on*: 2012-06-15

JADE. (2012). Java Agent DEvelopment Framework, *Available from:* http://jade.tilab.com, *Accessed on*: 2012-06-15

Komma, V. R.; Jain, P. K. & Mehta, N. K. (2005). Agent Based Approach for Automated Guided Vehicle System- A Review. *Proceedings of 4th DAAAM International Conference on Advanced Technologies for Developing Countries (ATDC'05)*, Kljajin, et al. (Ed.), pp. 425-430, ISBN: 9536048299, Slavonski Brod, Croatia, September 21-24, Mechanical Engineering Faculty in Slavonski Brod, Slavonski Brod.

Komma, V. R.; Jain, P. K. & Mehta, N. K. (2007). Agent-based simulation of a shop floor controller using hybrid communication Protocols. *International Journal of Simulation Modelling*, Vol. 6, No. 4, pp. 206-217, ISSN: 1726-4529.

Komma, V. R.; Jain, P. K. & Mehta, N. K. (2011). An approach for agent modelling in manufacturing on JADE™ reactive architecture. *The International Journal of Advanced Manufacturing Technology*, Vol. 52, No. 9-12, pp. 1079–1090, ISSN: 0268-3768.

Komma, V. R.; Jain, P. K. & Mehta, N. K. (2012). Ontology development and agent communication in agent-based simulation of AGVS. *International Journal of Simulation Modelling*, ISSN 1726-4529 (Accepted)

Law, A. M. & Kelton, D. W. (1991). *Simulation Modelling and Analysis*, McGraw-Hill Inc., ISBN: 9780070366985, Singapore

Luck, M.; McBurney, P. & Preist, C. (2003). Agent technology: enabling next generation computing - a roadmap for agent based computing, *AgentLink II*, *Available from*: http://www.agentlink.org/admin/docs/2003/2003-48.pdf, *Accessed on:* 2012-06-15

Mahadevan, B. & Narendran, T. T. (1994). A hybrid modelling approach to the design of an AGV-based material handling system for an FMS. *International Journal of Production Research*, Vol. 32, No. 9, pp. 2015–2033, ISSN: 0020-7543

Obitko, M. & Mařík, V. (2002). Ontologies for multi-agent systems in manufacturing domain, *Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA'02)*, pp. 597- 602, ISBN: 0-7695-1668-8, Septmber 2002, IEEE Computer Society

Paolucci, M. & Sacile, R. (2005). *Agent-Based Manufacturing and Control Systems: New Agile Manufacturing Solutions for Achieving Peak Performance*, CRC Press, ISBN: 9781574443363, Florida, USA

Protégé. (2012). The Protégé ontology editor and knowledge acquisition system, *Available from*: http://protege.stanford.edu, *Accessed*: 2012-06-15

Vis, I. F. A. (2006). Survey of Research in the Design and Control of Automated Guided Vehicle Systems, *European Journal of Operational Research*, Vol. 170, No. 3, pp. 677-709

Vrba, P. (2003). MAST: Manufacturing Agent Simulation Tool, *Proceedings Emerging Technologies and Factory Automation (ETFA '03)*, IEEE Conference, pp. 282-287, September 2003, ISBN: 0-7803-7937-3

Wallace, A. (2001). Application of AI to AGV Control – Agent Control of AGVs, *International Journal of Production Research*, Vol. 39, No. 4, (March 2001), pp. 709-726, ISSN: 0020–7543

Weiss, G. (Ed.). (1999). *Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, ISBN: 0-262-23203-0, Massachusetts.

Wooldridge, M. (2001). *An Introduction to Multiagent Systems*. John Wiley & Sons Ltd., ISBN: 047149691X, London, England