Int j simul model 11 (2012) 4, 173-184
ISSN 1726-4529 Original scientific paper

ONTOLOGY DEVELOPMENT AND AGENT COMMUNICATION IN AGENT-BASED SIMULATION OF AGVS

Komma, V. R.*; Jain, P. K.** & Mehta, N. K.**

* Department of Mechanical Engineering, Motilal Nehru National Institute of Technology,
Allahabad – 211004, Uttar Pradesh, India

**Department of Mechanical and Industrial Engineering,
Indian Institute of Technology Roorkee, Roorkee – 247667, Uttarakhand, India

E-Mail: vrkomma@gmail.com, pjainfme@iitr.ernet.in, mehtafme@iitr.ernet.in

Abstract
For effective information exchange in agent-based systems, domain-specific ontology is strongly needed, which is the vocabulary representing facts and relations of objects in the domain. In this paper, a semi-formal domain-specific ontology for shop floor focusing on Automated Guided Vehicle Systems (AGVS) is presented. The domain-specific ontology is developed on Protégé, a leading frame-based ontology editor, and is further converted into JADE™ (Java Agent DEvelopment Framework) compliant java classes. The developed ontology has been used in agent-communication of a simulator known as agent-based shop floor simulator (ABSFSim). Agent communication used in the ABSFSim is discussed. Agent-based simulation model of a sample manufacturing system is built with ABSFSim and the working of the model is verified and validated by comparing the output with an equivalent model developed in ProModel. The ontology development process presented in this paper is useful for developing agent-based models in different sections of manufacturing and also for developing high level agent-based software platforms for manufacturing systems.

(Received in February 2011, accepted in June 2012. This paper was with the authors 2 months for 2 revisions.)

Key Words: AGVS, ABSFSim, Agent-Based Simulation, Protégé Ontology Editor, JADE™

1. INTRODUCTION

Multi Agent System (MAS) is a relatively new specialization of Distributed Artificial Intelligence (DAI) that helps in developing complex systems with concurrent behaviour. Due to the complex nature of manufacturing domain, it has been a leading application of MAS. Automated Guided Vehicle System (AGVS) is an important modern material handling system due to its routing flexibility. Application of multi-agent concept in simulation leads to agent-based simulation. Agent-based simulation of AGVS helps in better understanding the working of dynamic and concurrent behaviour of entities. In agent-based systems, agent communication plays vital role in information exchange and decision making. For effective information exchange among the agents, there is a strong requirement of domain-specific ontology. Ontology of a domain is the vocabulary representing facts and relations of objects in the domain.

In this paper, a semi-formal domain-specific ontology for shop floor focusing on AGVS is presented. The domain-specific ontology is developed on Protégé [1], a leading ontology editor and knowledge-base framework. The developed Protégé Ontology is converted into JADE™ (Java Agent DEvelopment Framework) [2] compliant java classes with the help of ontology Beangenerator, a plug-in for Protégé. Ontology development is a preliminary step in agent-based simulation. Further, the developed ontology is successfully used in communication of agents in AGVS during the agent-based simulation on a simulator known

DOI:10.2507/IJSIMM11(4)1.193

as agent-based shop floor simulator (ABSFSim) [3], which is developed by the authors. A brief introduction to the agent-based technology is given below.

1.1 Agent-Based Technology

Agents are defined as autonomous, computational entities that can be viewed as perceiving their environment through sensors and acting upon their environment through effectors. A few common properties that can be assigned to an agent are autonomy, co-operativity, reactivity and pro-activeness. In addition to these properties, learning is a desirable property in applications like machine learning, computer-human interaction and social simulation [4]. Based on the characteristics of concrete agent architectures, agents can be classified as four classes, viz. logic-based, reactive, Belief-Desire-Intention (BDI) and layered agents. Suitability of agent architecture depends on the type of application. Layered architecture of agents is the most commonly used architecture due to its simplicity and robustness [5]. Most of the agent-based systems reported in literature were developed in Java™ programming language due to its built-in support for multi-threaded programming. As developing agents from scratch is time consuming and tedious task, researchers have developed several platforms for quick launching of the MAS [6]. Most of these platforms were specific to the application. To promote agent-based technology and the interoperability of its standards with other technologies, Foundation for Intelligent Physical Agents (FIPA) [7]; a non-commercial Swiss based organization, proposed specifications for agent-based systems. JADE [2], a non-commercial software developed at TILab, Italy, is one of the leading platforms that helps in developing FIPA compliant agent-based systems. ABSFSim, an agent-based simulator focusing on AGVS, has been developed on the top of JADE framework. ABSFSim is used for agent-based simulation in this work.

2. RELATED WORK

Wide-ranging details of agent-based technology and MASs were addressed in [5, 8]. The development of agent-based technology in different applications and its past, present and future developments were reported in the roadmap of agent technology [6]. In the past one and half decade, researchers have applied agent technology to manufacturing enterprise integration and supply chain management, manufacturing planning, scheduling and execution control, material handling and inventory management, and in developing new types of manufacturing systems such as Holonic Manufacturing Systems (HMS) [9]. PROSA (Product-Resource-Order-Staff Architecture) is a well-known architecture for manufacturing control which originated from HMS research project [10]. 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, which is optional and it may assist the basic agents in performing their task more optimally.

As far as AGVS are concerned, control of AGVS is greatly influenced by the type of layout that is employed on the shop floor. It is easier to control an AGVS in a single loop without intersections than in a large bidirectional network. In [11], AI was used 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 in [12]. A multi-agent based architecture for AGVS was proposed in [13], in which communication was achieved by using a relational database (blackboard system). A complete overview of agent-based manufacturing was presented in [9, 14, 15]. A detailed review of design and control of AGVS was reported in [16, 17].

A survey and comparison of usage of ontologies in the area of manufacturing and the semantic web was made in [18]. The ontologies in 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 each other. They 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. A manufacturing system engineering (MSE) ontology for semantic interoperability across extended project teams was reported in [19]. The MSE ontology enables the operation of an extended enterprise MSE Moderator to provide common understanding of manufacturing-related terms and therefore to enhance the semantic interoperability and reuse of knowledge resources within globally extended manufacturing teams.

In the above literature, ontology and agent communication required for the manufacturing domain have not been completely specified. Therefore, in this paper, details of development of domain-specific ontology for shop floor control focusing on AGVS are presented.

3. ONTOLOGY DEVELOPMENT FOR AGVS

Ontology development for AGVS was a part of ABSFSim development. Therefore, in this section, the agents used in ABSFSim are briefly introduced followed by the details of the domain-specific ontology development for AGVS.

3.1 Agents in AGVS

In the development of ABSFSim, different agent types are defined for a shop floor with AGVS such as part-agent, machine-agent, AGV-agent and so on. Agents in manufacturing domain are referred here as manufacturing agents. Table I shows the list of manufacturing agents and their intended role in ABSFSim. In this work, manufacturing agents are modelled on JADE reactive architecture, details of agent modelling in ABSFSim was reported in [20].

Table I: Manufacturing agents and their purposes.

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

To achieve the required functionality for different manufacturing agents, several JADE behaviour classes such as "*FSMBehaviour*", "*ParallelBehavior*" and "*SimpleBehaviour*" (with multi-steps) were customized. Agent's behaviour objects can be dynamically added or removed from its behaviour list in order to respond suitably to the messages received from other agents in its environment. Required behaviours in a particular finite state of an agent were determined by critically analysing the operation of the agent.

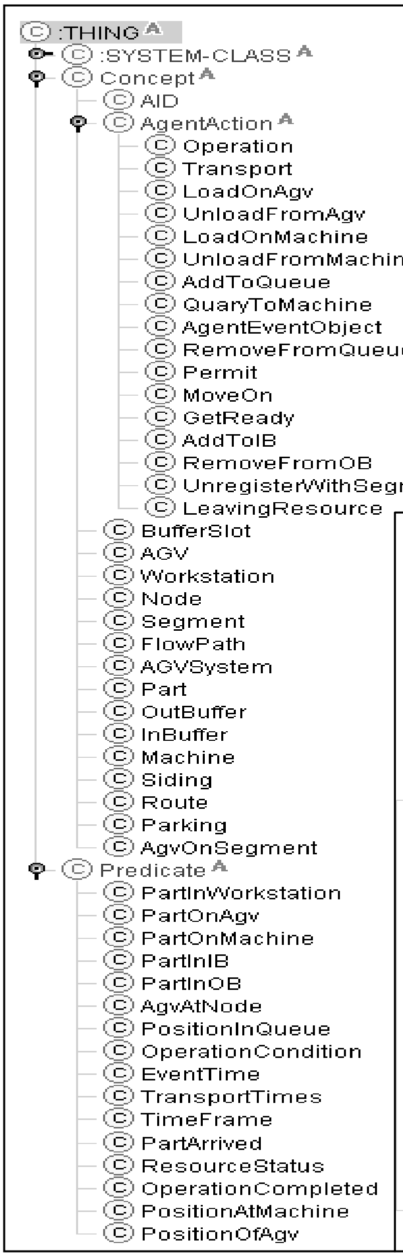
Agents in the manufacturing system can be broadly classified as two categories: agents that live for longer time period (resource and functional agents) and agents that live for shorter time period (part agents). Development of domain-specific ontology for AGVS and use of the ontological terms in the preparation of content expressions according to the selected Interaction Protocols (IPs) was part of the development of ABSFSim. Effective agent communication plays important role in information exchange and decision-making.

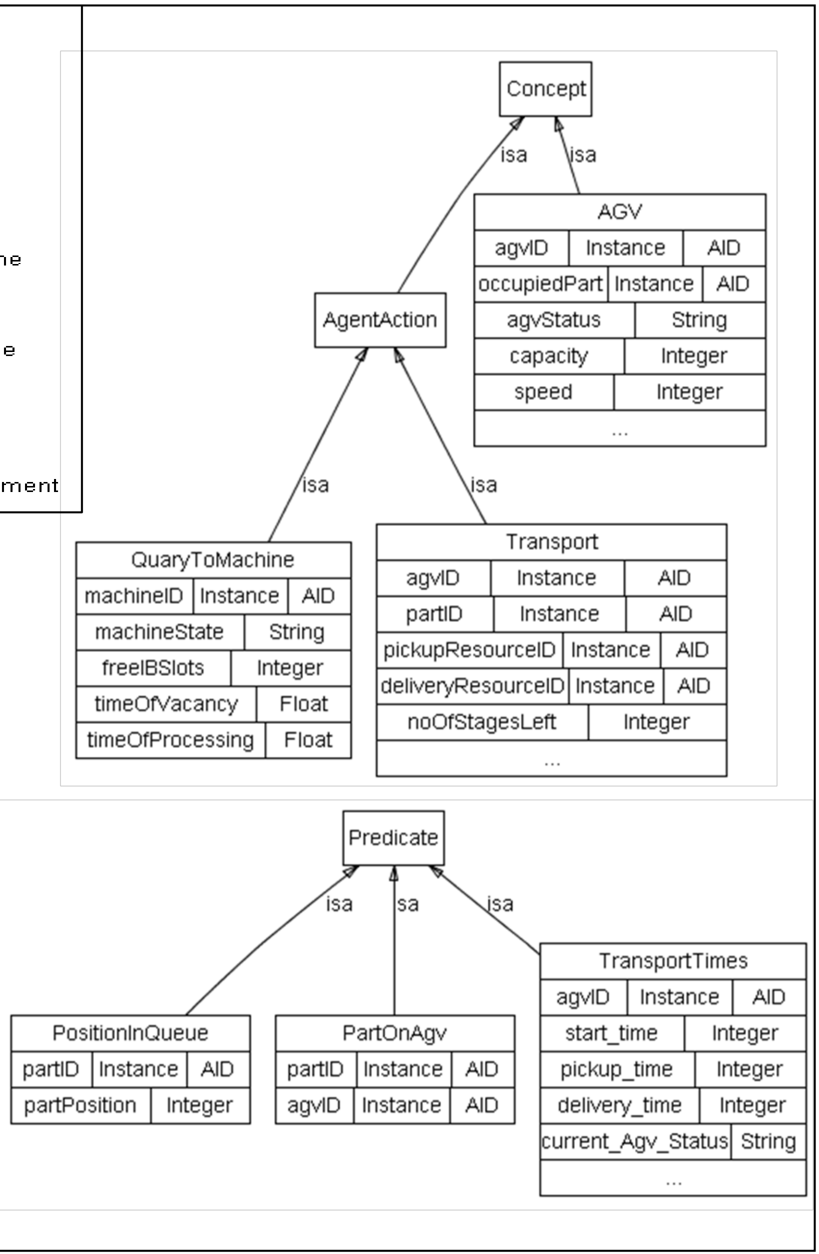
3.2 JADE compliant ontology development for AGVS

In agent-communication, agents exchange messages in a specific sequence to complete the conversation and achieve rational effect. A message is represented as per FIPA-ACL (FIPA-Agent Communication Language) that contains a number of slots for carrying 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. Communicative act indicates the purpose of the message being sent. The interaction protocol controls the sequence of messages to be exchanged. Content is an important slot of the message that carries information. The content is syntactically represented as per the grammar of the selected content language. In ABSFSim, content is expressed according to FIPA-SL (Semantic Language). To understand the content of a message both the sending and receiving agents should share the common ontology.

Ontology for a manufacturing enterprise can be developed by extending or reusing the ontologies of sub-domains of manufacturing such as material handling, inventory control, shop floor, factory, enterprise control and supply chain management. In this work, a semi-formal ontology for shop floor focusing on AGVS is developed on Protégé, a leading frame-based ontology and knowledge editor, which is available from Stanford University, USA [18]. The developed Protégé ontology was converted into JADE complaint Java classes with the help of the Ontology Beangenerator, a plug-in for Protégé. Automatic generation of the Java classes from the Protégé project allows not only developing large size ontology in less time but also provides flexibility for the development.

When Ontology Beangenerator plug-in is added to a Protégé environment, "*SimpleJADEAbstractOntology*" project is loaded that implements the "*BasicOntology*", i.e., the ontology containing schemas for primitive types and SL0 operators, which are required for minimal agent interaction on JADE platform. Ontology contains concepts, predicates and agent-actions. Concepts are the facts of the domain and in AGVS; facts are objects in the system such as part, machine and AGV. These can be correlated with 'Noun' in English grammar. The names of these objects are represented as classes and each class has several attributes. For example, Machine is a class with several attributes such as capacity of the machine, type of the machine etc. In the Protégé frame-based knowledge, the classes are represented as frames and attributes as slots. A similar slot or attribute can exist in more than one class. For example, AID (Agent-Identifier), a name slot to identify an agent, is a common attribute among several objects, such as Machine, AGV, Part etc. Concepts cannot be used directly as content of a message; however, they can be used in combination with predicates. This is obvious from the fact that nouns alone cannot make a meaningful sentence. Agent-actions are also a special type of concepts, which represent the actions that an agent can perform. Thus, an agent-action can be correlated with ‘verb’ such as Transport, Operation etc. Predicates represent the relations between concepts in the domain and are synonymous with ‘preposition’ in the grammar. For example, “*PartOnAgv*” represents a predicate that relates a Part and an AGV with the statement: “Part is loaded on AGV”. The expected result for the predicate is either true or false indicating the belief is right or wrong. The predicates can be used while querying or while conveying information to other agents. Predicates can be directly used as a content expression in some of the messages. The hierarchical tree structure of the developed semi-formal AGVS Ontology that consists of concepts, agent-actions and predicates is shown in Fig. 1. Concepts are: *“BufferSlot”*, *“AGV”*, *“WorkStation”*, *“Node”*, *“Segment”*, *“FlowPath*” etc., agent-actions are *“Operation”*, *“Transport”*, *“LoadOnAGV”*, *“UnloadFromAGV”* etc., and predicates are *“PartInWorkstation, “PartOnAGV”*, *“PartOnMachine”* etc.





AGV		
agvID	Instance	AID
occupiedPart	Instance	AID
agvStatus	String	
capacity	Integer	
speed	Integer	
...		

QuaryToMachine		
machineID	Instance	AID
machineState	String	
freeIBSlots	Integer	
timeOfVacancy	Float	
timeOfProcessing	Float	

Transport		
agvID	Instance	AID
partID	Instance	AID
pickupResourceID	Instance	AID
deliveryResourceID	Instance	AID
noOfStagesLeft	Integer	
...		

PositionInQueue		
partID	Instance	AID
partPosition	Integer	

PartOnAgv		
partID	Instance	AID
agvID	Instance	AID

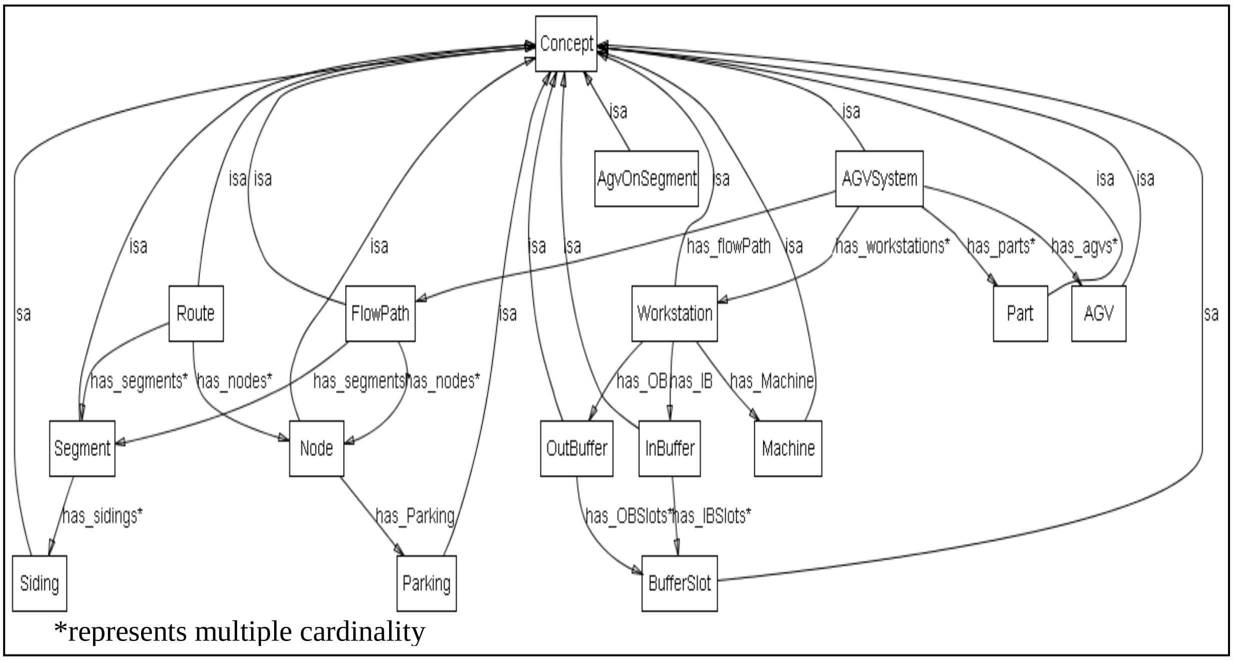
TransportTimes		
agvID	Instance	AID
start_time	Integer	
pickup_time	Integer	
delivery_time	Integer	
current_Agv_Status	String	
...		

Figure 1: Hierarchical tree structure of the semi-formal AGVS ontology.

Figure 2: Ontological slots of different elements of the AGVS ontology.

In the preparation of a content expression of message, instances of the suitable ontological classes were created and their attributes were properly set. For illustration, some of the ontological classes with their attributes that are subclasses of concepts, agent actions and predicates are shown in Fig. 2. While developing the AGVS ontology, Ontoviz and TGviz plug-ins of Protégé are used in visualizing and analysing the ontology. The structure of some of the concepts in AGVS ontology, which is generated with Ontoviz under specific configuration, is shown in Fig. 3. This structure shows that AGVS consists of AGVs, workstations, flowpath and parts. Further, each workstation consists of an input buffer, a machine and an output buffer. Flowpath consists of a set of nodes and segments. Each segment has a siding (vehicle buffer) at either end of the segment or both the ends. When a node acts as a pickup or delivery point for AGVs, then a parking location exists at the node.

The developed AGVS ontology facilitates in working and coordination of multi-agents on the shop floor while dispatching and routing the AGVs. The ontology covers most of the domain and task specific ontological terms. ABSFSim uses a set of these ontological classes for communication among the agents on shop floor. Nevertheless, as it is the application specific ontology, it can be easily modified or extended for any specific application on the shop floor, such as extending for other material handling equipment or machine scheduling etc.



*represents multiple cardinality

Figure 3: The structure of some of the concepts in AGVS ontology.

4. AGENT-COMMUNICATION DURING SIMULATION

Most of the interaction protocols, used for agent communication in the ABSFSim, are the FIPA-Interaction Protocols, which are essentially basic protocols applicable to systems in general. For manufacturing systems, these basic protocols must be suitably modified by integration/hybridization for better communication among the various manufacturing agents.

In the developed ABSFSim, two hybrid communication protocols, namely, Part-Machine-AGV (PMA) hybrid contract-net protocol and AGV-Node-Segment (ANS) hybrid request protocol proposed in [21] have been implemented for communication among the manufacturing agents. The PMA hybrid contract-net protocol is useful for a part-agent to select simultaneously the best possible machine and AGV combination for performing the next stage operation. The ANS hybrid request protocol is an AGV initiated protocol to seek permission to move along the selected segment through a node (control point).

In ABSFSim, different messages were exchanged during the implementation of PMA and ANS hybrid protocols. Content is the core of a message and the information contained in it helps in decision making by the receiving agent. Therefore, content of messages shall be modified and filled according to the goal of a specific agent. Two sample contents, represented in FIPA-SL format, are given in Table II, which were used in ABSFSim. The sample contents help the agent developers in preparing informative and meaningful contents for ACL messages.

Table II. Content representation of some of the ACL messages.

1. Content of CFP (Call for Proposal) message sent by Part_0 to all AGVs

```
 ((action
    (agent-identifier :name AGV@vrkomma:1099/JADE)
    (Transport
      :partID
        (agent-identifier :name Part_0@vrkomma:1099/JADE
          :addresses (sequence http://vrkomma:7778/acc))
      :partType 2  :ineffectiveTime 17  :noOfStagesLeft 5  :arrivalTime 86  :partPriority 5
      :pickupResourceID
        (agent-identifier :name Machine_8@vrkomma:1099/JADE
          :addresses (sequence http://vrkomma:7778/acc))
      :effectiveTime 340
      :deliveryResourceID
        (agent-identifier :name Machine_3@vrkomma:1099/JADE
          :addresses (sequence http://vrkomma:7778/acc))
      :dueTime 0)
      :agvID
        (agent-identifier :name AGV@vrkomma:1099/JADE))
  (OperationCondition
    (agent-identifier :name Machine_3@vrkomma:1099/JADE
      :addresses (sequence http://vrkomma:7778/acc)) 0 260 260 Stage2_Part_0
    (agent-identifier :name Part_0@vrkomma:1099/JADE
      :addresses (sequence http://vrkomma:7778/acc)) 0))
```

Brief description: The content has two agent-actions i.e. "*Transport*" and "*OperationCondition*". "*Transport*" agent-action contains details of part such as part identity (AID), part type, total operation time (effective time) and transportation time (ineffective time), part priority, identities of pickup and delivery resources, due time (if any). Here, AGV@vrkomma:1099/JADE indicates any potential AGV. "*OperationCondition*" contains constraints imposed by the machine such as availability of empty slot at the machine.

In response to the CFP, each AGV-agent submits a proposal that includes the details of current state of the AGV, possible pickup and delivery times for completing the task. The part-agent evaluates all the received proposal and suitable decision will be made according to the goal of the agent.

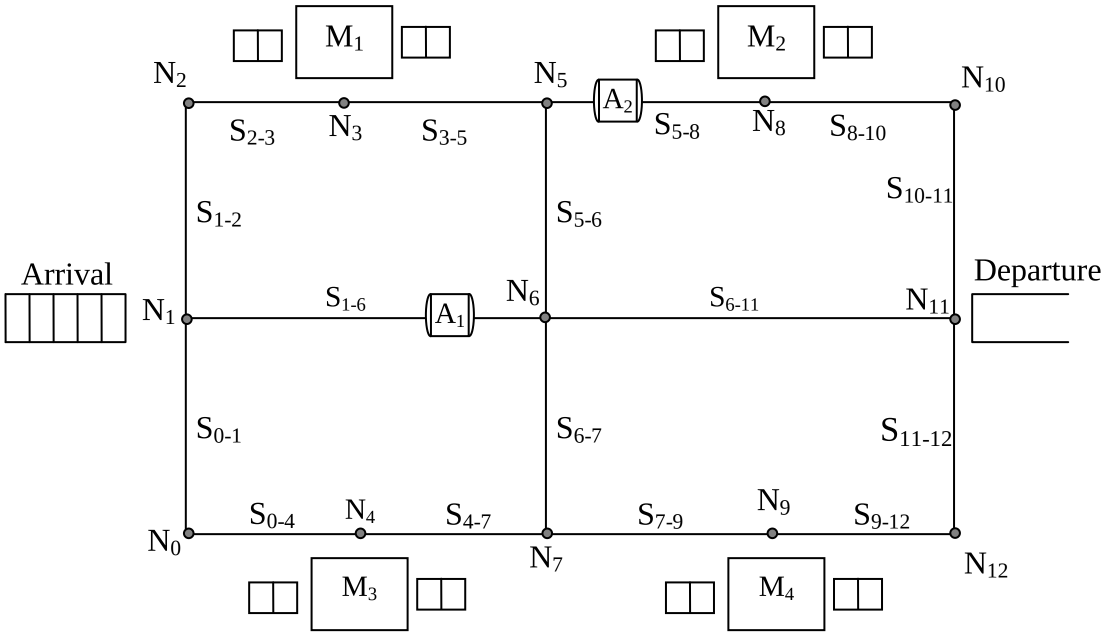
2. Content of request message from AGV_1 to Node_3 to permit to move on to Segment_0_3

```
((action
    (agent-identifier :name Node_3@vrkomma:1099/JADE)
    (Permit
      :next_node 0   :n2n_node 1  :event_time 372
      :agv_Details
        (AgvOnSegment
          :agvID
           (agent-identifier  :name AGV_1@vrkomma:1099/JADE
             :addresses (sequence http://vrkomma:7778/acc)
          :speed 1  :direction 2
          :in_Resource Siding_3_Segment_3_6
          :fromNode 3          :expected_Exit_Time 0
          :enter_Time 372    :is_PorD_Node false)
          :next_Move_To Segment_0_3
          :toNode 0  :exit_Time 0))))
```

Brief description: The content has an agent-action "*Permit*" that contains the details of next node, next to next node (n2n_node) and AGV details (AGV AID, current position etc.). Based on the response from Node_3, the AGV_1 either move along the Segment_0_3 or wait in the current Siding (Siding 3 on Segment_3_6).

5. MODEL DEVELOPMENT ON ABSFSim

The sample manufacturing system considered in [20, 21] is considered here for agent-based simulation on ABSFSim, which is shown in Fig. 4. The sample manufacturing system has *four* machines, *two* AGVs, *one* arrival and *one* departure location. The layout has 13 nodes (graphical position of all nodes is given in Table III) and 16 segments. Parts arrive at part arrival queue in unit quantity at discrete points of time with fixed inter-arrival time of 15 time units. Three part types were considered and the sequence of arrival of part types are considered to takes place in a cyclic manner (i.e. part-types 0-1-2-0-1-2 and so on). For each part type, operation sequence and corresponding operation time is given in Table IV. Speed of each AGV was considered as 1 length unit/time unit.



Legend: N_x → x^{th} Node
M_x → x^{th} Machine with its input and output buffers
A_x → x^{th} AGV
S_{x_y} → Segment connected node x to node y (x < y)

Figure 4: Sample model of a manufacturing system for agent-based simulation on ABSFSim.

Table III: Nodes and their corresponding coordinates (in length units) of the layout.

Node	N_0	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}	N_{12}
x	0	0	0	2	2	4	4	4	6	6	8	8	8
y	0	2	4	4	0	4	2	0	4	0	4	2	0

Table IV: Details of part types produced in the sample manufacturing system.

Part type, j	Volume mix	No. of stages, s_j	Operation sequence (Operation Time in time units)
0	33.33 %	3	M_1 (10) – M_2 (8) – M_3 (4)
1	33.33 %	4	M_2 (8) – M_1 (3) – M_4 (5) – M_3 (6)
2	33.33 %	3	M_1 (4) – M_2 (7) – M_4 (3)

Simulation results were collected after simulating the system for a period of 2000 time units. Total number of parts arrived during the simulation were 134 as the first arrival of a part is scheduled at 0 time units (which is same as 1 + 2000/15 = 134).

Some of the simulation output measures of ABSFSim are used for estimation of other output measures based on their interrelation. For instance, utilization of machines and average number of deliveries made by the AGVs are estimated from the number of finished parts. The cross-verification of output measures is meant for partial verification of the developed model on the ABSFSim. Details of the cross verification of the model on ABSFSim is given below.

Based on the number finished parts at the end of the simulation, one can calculate the total operation times of machines (Busy time) with the help of the relation:

$$TO_i = \sum_{j=1}^{Np} n_j \times O_{ij} \tag{1}$$

In (1), N_p represents the number of part types, n_j is the number of finished parts of part type j, and O_{ij} represents the operation time on machine i for part type j. If a job visits the machine more than once then O_{ij} is the sum of the operation times of part type j on machine i. In the sample system considered, incidentally parts visit a machine only once in its sequence. Similarly, average loaded travel time of AGV ($\overline{T_l}$) can be obtained as:

$$\overline{T_l} = \frac{\sum_{j=1}^{Np} n_j \times Tn_j}{Nagv} \tag{2}$$

In (2), Tn_j is the total travel time required for part type j and $Nagv$ is the number of AGVs in the system.

Theoretical make-span time of each part type, which is the sum of total operation and travel times, is given in Table V. Total operation time of each machine (with the help of (1)) and the corresponding machine utilizations is calculated and presented in Table VI. It can be observed from the Table VI that the utilizations of machines obtained from the ABSFSim are slightly higher than the calculated values based on number of finished parts as the former includes the operations carried out on the semi-finished parts, which are still on the shop floor at the end of the simulation. Similarly, the calculated average loaded travel time of AGV (calculated from (2)) is 1203, whereas from ABSFSim it is obtained as 1244. The reason for lower calculated value over the value obtained from ABSFSim is that the former does not consider the travel times of semi-finished parts, which are still on the shop floor at the end of simulation.

Table V: Theoretical make-span times of different part types.

Part type, j	Total operation time $\sum_{i=1}^{s_j} O_{ij}$	Total travel time, Tn_j	Make-span time
0	22	32	54
1	22	42	64
2	14	28	42

Table VI: Determination of machine utilization based on number of finished parts.

Part type, j	Number of finished parts, n_j	$n_j \times O_{1j}$	$n_j \times O_{2j}$	$n_j \times O_{3j}$	$n_j \times O_{4j}$
0	24	24 × 10	24 × 8	24 × 4	24 × 0
1	23	23 × 3	23 × 8	23 × 6	23 × 5
2	24	24 × 4	24 × 7	24 × 0	24 ×3
Total operation time of Machine i, TO_i		405	544	234	187
Calculated utilization		20.25 %	27.20 %	11.70 %	9.35 %
Actual utilization obtained from ABSFSim		21.25 %	28.40 %	12.10 %	9.60 %

For obtaining higher level of confidence on the model developed in ABSFSim, the output is compared with almost similar model developed in ProModel®, a conventional simulation environment for manufacturing. The outputs obtained from both the models are in good agreement; it is evident from the comparison of the percentages of time spent in different finite states of machines and AGVs in Table VII, and VIII. The slight difference in the results of ABSFSim and ProModel is due to the difference in dispatching of AGVs in the two models. In ABSFSim, the dispatching of AGVs is the hybrid contract-net based dispatching (HCNBD) which is based on PMA protocol [21]; whereas in ProModel dispatching of AGVs is based on longest waiting entity. The difference between the models increases with the increase in length of simulation run. For the considered simulation run also, the difference is clear for comparison of number of finished parts and average throughput time of parts (Table IX). It is difficult to implement HCNBD on ProModel or any other conventional simulation platforms. Therefore, agent-based modelling, such as ABSFSim, provides greater flexibility for decision making over the conventional simulation platforms in dynamic environment with the help of agent communication. However, as agent-based technology and its programming is in the process of standardization, more effort is required to implement or use any agent-based simulation models in industries. The ontology development process presented in this paper is useful for developing agent-based models in different sections of manufacturing. This also helps in bringing commercial agent-based platforms for manufacturing simulation.

Table VII: Comparison of percentage of time spent in finite states by machines.

Resource	ABSFSim			ProModel		
	Idle	Operation	Blocked	Idle	Operation	Blocked
M1	78.75 %	21.25 %	0.00 %	79.10 %	20.90 %	0.00 %
M2	71.20 %	28.40 %	0.40 %	72.00 %	28.00 %	0.00 %
M3	87.90 %	12.10 %	0.00 %	88.30 %	11.70 %	0.00 %
M4	90.40 %	9.60 %	0.00 %	90.60 %	9.40 %	0.00 %

Table VIII. Comparison of percentage of time spent in finite states by AGVs.

Resource	ABSFSim			ProModel		
	Idle	Empty travel	Loaded travel	Idle	Empty travel	Loaded travel
AGV_1	0.35 %	41.80 %	57.85 %	0.51 %	40.59 %	58.90 %
AGV_2	1.45 %	32.00 %	66.55 %	0.86 %	36.79 %	62.35 %
AGV_{avg}	0.90 %	36.90 %	62.20 %	0.69 %	38.69 %	60.62 %

Table IX: Comparison of number of finished parts and average throughput time.

Part type, *j*	Number of finished parts and average throughput time from ABSFSim	Number of finished parts and average throughput time from ProModel
0	24 (503.9)	19 (632.3)
1	23 (528.7)	26 (534.8)
2	24 (515.6)	22 (591.8)

6. CONCLUSIONS

Domain specific ontology and agent communication plays vital role in decision making and implementing different operational strategies in shop floor control. Therefore, in this paper a semi-formal ontology for shop floor focusing on AGVS is presented. The developed AGVS ontology is successfully used in the development of ABSFSim. The AGVS ontology is first developed on Protégé ontology editor and is converted into JADE compliant java classes with the help of Ontology Beangenerator. This is the fastest way of creating and modifying the ontology for the manufacturing system. In agent communication in ABSFSim, contents of messages were represented with the AGVS ontological elements according to FIPA-SL grammar. A simulation model of a sample manufacturing system was implemented in ABSFSim. The output of the model has been analysed to verify the working of ABSFSim. Further, the output has been compared with an almost equivalent model developed on ProModel. The results are in good agreement indicating the correct working of the ABSFSim and its communication. The scope for future work by the authors is to use ABSFSim to implement and test the performance of different decision making on conventional AGV layout such as HCNBD. Further, the ABSFSim will be extended to model AGV systems with multilane flowpath and multiple load AGVs.

Agent-based modelling of AGVS facilitates in developing models that are closer to real systems. Dynamic operational strategies, such as real time dispatching of AGVs, are easier to implement in agent-based simulation models than conventional simulation models with the help of agent communication. However, as agent-based technology is still in the process of standardization, more effort is required to implement agent-based simulation in industries. The ontology development process presented in this paper is useful for developing agent-based models in different sections of manufacturing and also for developing high level agent-based software platforms for manufacturing systems.

REFERENCES

[1] Protégé. (2009). The Protégé ontology editor and knowledge acquisition system, from *http://protege.stanford.edu*, accessed on 22-04-2012

[2] JADE. (2009). Java Agent DEvelopment Framework, from *http://jade.tilab.com*, accessed on 22-04-2012

[3] Komma, V. R. (2009). *Agent-Based Simulation of Distributed Automated Guided Vehicle Systems*, Ph.D. Thesis, Department of Mechanical and Industrial Engineering, Indian Institute of Technology Roorkee, Roorkee, India

[4] Barnett, T.; Ehlers, E. (2010). Cloud computing for Synergized Emotional Model Evolution in Multi-Agent Learning Systems**,** *Strojniški vestnik - Journal of Mechanical Engineering*, Vol. 56, No. 11, 718-727

[5] Weiss, G. (Ed.). (1999). *Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, Massachusetts, USA

[6] Luck, M.; McBurney, P.; Preist, C. (2003). *Agent technology: enabling next generation computing – a roadmap for agent based computing*. AgentLink, from *http://www.agentlink.org/admin/docs/2003/2003-48.pdf*, accessed on 22-04-2012

[7] FIPA. (2009). Foundation for Intelligent Physical Agents, from *http://www.fipa.org*, accessed on 11-08-2009

[8] Wooldridge, M. (2001). *An Introduction to Multiagent Systems*, John Wiley & Sons, London

[9] Shen, W.; Hao, Q.; Yoon, H. J.; Norrie, D. H. (2006). Applications of agent-based systems in intelligent manufacturing: An updated review, *Advanced Engineering Informatics*, Vol. 20, No. 4, 415-431, doi:10.1016/j.aei.2006.05.004

[10] Brussel, H. V.; Wyns, J.; Valckenaers, P.; Bongaerts, L.; Peeters, P. (1998). Reference architecture for holonic manufacturing systems: PROSA, *Computers in Industry*, Vol. 37, 255-274, doi:10.1016/S0166-3615(98)00102-X

[11] Wallace, A. (2001). Application of AI to AGV control – agent control of AGVs, *International Journal of Production Research*, Vol. 39, No. 4, 709-726, doi:10.1080/00207540010007310

[12] Vrba, P. (2003). MAST: Manufacturing Agent Simulation Tool, *Proceedings of Emerging Technologies and Factory Automation (ETFA '03): IEEE Conference*, 282-287, doi:10.1109/ETFA.2003.1247718

[13] Farahvash, P.; Boucher, T. O. (2004). A multi-agent architecture for control of AGV systems, *Robotics and Computer-Integrated Manufacturing*, Vol. 20, 473-483, doi:10.1016/j.rcim.2004.07.005

[14] Deen, S. M. (Ed.). (2003). *Agent Based Manufacturing – Advances in the Holonic Approach*, Springer- Verlag, Heidelberg

[15] Paolucci, M.; Sacile, R. (2005). *Agent-Based Manufacturing and Control Systems: New Agile Manufacturing Solutions for Achieving Peak Performance*, CRC Press LLC, Boca Raton, Florida

[16] 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, 677-709, doi:10.1016/j.ejor.2004.09.020

[17] Le-Anh, T.; De Koster, M. B. M. (2006). A review of design and control of automated guided vehicle systems, *European Journal of Operational Research*, Vol. 171, No. 1, 1-23, doi:10.1016/j.ejor.2005.01.036

[18] 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)*, from *http://ieeexplore.ieee.org/iel5/8104/22410/01045963.pdf*, accessed on 11-08-2009

[19] Lin, H. K.; Harding, J. A.; Shahbaz, M. (2004). Manufacturing system engineering ontology for semantic interoperability across extended project teams, *International Journal of Production Research*, Vol. 42, No. 24, 5099-5118, doi:10.1080/00207540412331281999

[20] Komma, V. R.; Jain, P. K.; Mehta, N. K. (2011), An approach for agent modeling in manufacturing on JADE™ reactive architecture, *The International Journal of Advanced Manufacturing Technology*, Vol. 52, No. 9-12, 1079-1090, doi:10.1007/s00170-010-2784-2

[21] 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, 206-217, doi:10.2507/IJSIMM06(4)2.096