An Intelligent Simulation Environment for Manufacturing Systems

Nancy Ruiz[a], Adriana Giret[a*], Vicente Botti[a], and Victor Feria[b],
[a] Departmento de Sistemas informaticos y Computacion,
[b] Departamento de Ingenieria de Proyectos,
Universidad Politecnica de Valencia.
{nruiz,agiret,vbotti}@dsic.upv.es, vferiap@doctor.upv.es

Abstract

The manufacturing field is an area where the application of simulation is an essential tool for validating methods and architectures before applying them on the factory floor. Despite the fact that there are a great number of simulation tools, most of them do not take into account the specific requirements of the "new manufacturing era" such as distributed organization, interoperability, cooperation, scalability, fault tolerance and agility. On the other hand, Multiagent System technology has demonstrated its utility in manufacturing system modeling and implementation. Agenthood features such as proactivity, reactivity, and sociability may also be useful for associating them with the specific simulation needs of the new changing requirements for manufacturing systems. In this paper, an Agent-supported Simulation Environment for Intelligent Manufacturing Systems is presented. The different roles that are played by the agents of the simulation environment are defined taking into account the specific dynamic features in manufacturing simulation and the requirements of the new manufacturing era. Moreover, the interaction and cooperation scenarios among these agents are specified to facilitate manufacturing simulation in an appropriate and flexible way. A detailed evaluation study, with regards to the new manufacturing era requirements, demonstrates the advantages of the proposed approach over current state-of-the-art proposals.

Keywords: Agent-based Simulation, Multiagent Systems, Simulation Tool, Intelligent Manufacturing System.

1. Introduction

According to Heim (Heim, 1997), simulation models have the potential to provide most of the information that is required and reduce the risk linked to the design, analysis, and operation of complex systems. Thus, simulation technology has been widely applied to design manufacturing systems (Yu and Popplewell, 1994).

In a previous study (Ruiz et al., 2006) it was possible to identify specific features that have not been included in simulation tools until now. Almost all of the tools include a sub-group of the requirements of the new manufacturing era, but none of them offers features such as autonomy or proactivity which add flexibility to the manufacturing system simulation. Most of the Simulation Tools are focused on solving problems related to manufacturing scheduling. Some state-of-the-art simulation tools also offer an optimization function to improve static models before executing the simulation process, and other tools allow the user to modify the model "on-the-fly". Current simulation tools allows the definition of static models, but it is not possible to have dynamic reconfiguration, for example, in Supply Chain management execution or temporal configurations. However, since manufacturing systems' requirements (specification) change dynamically, according to customer needs, the simulation tools must also adapt to those changing needs and, at the same time, provide support for allowing run-time modifications to the manufacturing model under simulation. In order to cope with this problem, our goal is to propose a model that allows to define the available resources and their capabilities at a shop floor, and, at the same time, manufacturing routes which do not a-priori link processes to specific resources. In this way it could be possible to assign resources during the simulation execution according to the availability and capability of these resources when the production orders arrive. To the best of our knowledge, there are currently no simulation tools that provide this kind of flexibility.

Multiagent System technology has demonstrated its utility in manufacturing system modeling. Some examples are CIIMPLEX, (Peng et al., 1998); HOLOS Architecture (Rabelo and Camarinha-Matos, 1994); Metha-Morph I (Maturana and Norrie, 1996); DEDEMAS (Tonshoff and et. al., 2000); MASCADA, (Valckenaers and et. al., 1999); and ANEMONA (Botti and Giret., 2008). Based on the features of agents (proactivity, reactivity, and sociability), it has been possible to apply this technology to Enterprise Integration and Supply Chain management. Thus, agent-oriented simulation approaches for manufacturing systems are suggested as an emergent solution for today's manufacturing requirements (Shen and Norrie, 1999a). MAS technology offers advantages when the low level behavior is known but it is not possible to know and/or predict the whole behavior (high level) of the system. In today's manufacturing systems this is the most common scenario, i.e. the machine/tools in the factory floor are known but the specific details of the manufacturing products constantly change and as such the manufacturing routes, the factory sequence processes, then resources have to adapt to that changes by providing time to execute specific processes. MAS Modeling allows the definition of reconfigurable models that takes into account concepts such as: negotiation, trust and reputation in dynamic organizations, among others.

Therefore, the Agent-supported Simulation Environment, proposed in this paper, focuses on the improvement of the modeling and simulation processes of a Shop Floor, solving problems related to flexibility, control and knowledge distribution, complex behavior simulation, automatic creation and/or elimination of elements, and process execution according to the current system state. The main goal is to provide a flexible simulation environment that can be adapted to solve the new manufacturing requirements. This paper presents SimIShop, which is a prototype used to validate the proposed agent-supported architecture for the simulation of Intelligent Manufacturing Systems.

The rest of this paper is organized as follows. Section 2 presents the state-of-the-art review of simulation tools. Section 3 presents a brief description of the Simulation Architecture of SimIshop. Section 5 presents a brief description of the framework of the agent-supported simulator implementation that will be used to simulate a model of an intelligent manufacturing system. Section 5.1 describes the modeling process that allows the representation of an intelligent manufacturing system. Section 5.2 presents the agent interaction during the simulation process. The simulation process includes tasks such as product creation, and launch of production order. Section 5.3 presents the metrics for the evaluation of the created models. Section 5.4 presents the metrics for the evaluation of shop floor configurations and the production data that are used in Model Simulation. Section 6 presents the evaluation of the proposal with regards to the requirements of the new manufacturing era and the improvements as a result of applying the agent paradigm. Finally, the conclusions and future work are presented.

2. State of the art review and comprehensive analysis

In this section the state of the art on simulation tools for manufacturing systems is analyzed. An in depth study is detailed with respect to the key issues (robustness and agility) for modeling and simulating flexible manufacturing systems that react to changing requirements. In this study, a focus is made on how the agent-based features could help to cope with the specified feature. In order to organize this analysis two phases are considered: the simulation model creation and/or definition, and the simulation itself.

The Evaluation of the Model Creation Phase is based on the Eldabi and Paul proposal (Eldabi and Paul, 2001). This methodology is based on Business Process Modeling (BPM). The evaluated features are selected from a previous evaluation of Simulation Tools and a detailed list of collected features specified by Nikoukaran in (Nikoukaran and Paul, 1999). The evaluation detail includes features linked to each simulation phase proposed by Nikoukaran (Nikoukaran and Paul, 1999) and Banks et al. (Banks et al., 2001). Also, the results include the advantages of using the agent paradigm. Some additional features were not taken into account because they are outside of the simulation process (i.e. capacitation, technical support, version updates, etc.).

Table 1 and 2 present the features taken into account in Model Creation Phase and their evaluation. For the Model Simulation Phase: a) Table 3 presents the features and the evaluation related to the Execution Environment, b) Table 4 presents the Animation during Simulation evaluation, and c) Table 5 presents the evaluation of the Experimentation and Results Analysis.

From the evaluation results presented in Tables 1 to 5 it is clear that current state of the art simulation tools do not properly deal with manufacturing systems that require to adapt to changing requirements. It is also clear that agent-based simulation can improve this situation. In the following section it is presented the proposed agent-based model for simulation of manufacturing systems.

No.	Feature	Description	Tools-Evaluation	Agent Paradigm Added Value
1	Modeling world	It includes: Process Interaction, Event Perspectives, Continuous Modeling.	The following state-of-the-art tools cope with this feature: Arena (Arena and http://www.arenasimulation.com, 2005), Extend (Extend, 2005), Microsaint (MicroSaint and http://www.maad.com, 2005), Quest (Quest and http://www.delmia.com, 2005), Witness (Witness and http://www.lanner.com, 2005), FlexSim (FlexSim, 2005) allow the definition of material flow and manufacturing systems for their analysis.	MAS approaches provide more flexibility by the distributing control and management of process in an individual level.
2	Capability of Input Data Analysis	It is implemented by using empiric estimated distributions or statistical data.	Factor/Aim (FactorAim and http://www.wintek.com, 2005), MicroSaint, DE3 (DE3, 2014) allow integration by file transfer. Arena, Extend, Promodel (Promodel and http://www.promodel.com, 2005), Witness, Flexsim among others use OLE, ODBC, and/or Active X to implement it.	The data associated to each entity of the MAS model are used during negotiation processes for resources allocation among Production Orders and the resources (workers, machines, tools) where the individual attributes are used to obtain more precise information. Thus,the information is distributed and can be managed by different models and technologies whenever it is required.
3	Construction of graphic model	Including: flow of processes, Flowchart, or net approaches.	Tools such as: Arena, Extend, MicroSaint, AutoMod (Automod, 2014), DE3, Grasp 2000 (Grasp2000, 2014), Flexsim, Quest offer menu bars, icon panel, dialog boxes.	
4	Conditional Routes	The route Entities are based on predefined conditions or attributes.	Arena, Extend, MicroSaint, Quest, Witness allow to define material flow and production routes. DE3, Grasp 2000 allow conveyor design. DE3 includes warehousing and distribution taking into account factory capacity.	The flexibility of the MAS APPROACH to apply different strategies to the resource allocation and the association of process to resources, allows to include a wide variety of physical routes and the use optimization of the available resources when production orders are processed.

Table 1: Evaluation of Features related to the Model Creation - Part 1

No.	Feature	Description	Tools/Evaluation	Agent Paradigm Added Value
5	Programming of Simulation	It requires to add procedural cognition in a high level.	AutoMod allows to simulate continuous flow and automated material handling, Enterprise Dynamics allows to define scheduling, process control. Most of tools (Arena, FlexSim, ProModel, Factor Aim)predefine the route including conditional rules (such as the use of ”if”) to change sequences.	It is possible to include complex behavior in individual level according to current system state. With MAS technology it is possible to build complicated functions that can add autonomy, and flexibility to the Simulation programming.
6	Syntaxis	Use of solid language, ambiguous or similar to English.	Arena offers animation language (SIMAN), Factor Aim offers process flow language, MicroSaint uses a ”parser” to converts mathematical and/or logical expressions into computer code. Grasp200 uses native robot languages.	Agent paradigm allows to use ”common” English language to identity agent actions both programming and interfaces.
7	Flexibility of input data	Accept data from external files, databases, spreadsheet and others.	Accepts data from external files. There has been predefined some easy templates to import data from real systems (resources, inventory, processes)	It is possible to accept data from external files such as formatted files (.csv), and spreadsheets.
8	Customized Objects build by User	Reusable Objects, templates and submodels.	Grasp2000, DE3, Arena, Witness, ProModel offer library of materials handling, and real process data.	It is possible to provide templates of agent models according to a type (workers, machines, tools, products, routes, bill of material). These templates include the specification of attributes that allows the definition of their capabilities (type, process that can made, capacity per hour, capability grade(beginner, medium, expert) and calendar).

Table 2: Evaluation of Features related to the Model Creation - Part 2

No.	Feature	Description	Evaluation	Agent Paradigm Added Value
1	Execution Speed	A wide number of executions require speed management due to a wide number of replications, impact of results and experiments.	Arena, Promodel, FactorAim, Quest, Witness, allow to define speed for simulation. Extend allows to run continuous simulations for DES input values. DE3 can execute real-time interactive simulations.	The execution speed also affects the processing time of agent tasks related to process production orders, and their additional asynchronous tasks. The asynchronous tasks include verification process of environment changes (current state of each agent) to update animation, or create or remove agent instances that are not required.
2	Model Size, number of variables and attributes	This feature must not be limited.	Grasp2000 support multiple robots. Tools do not specify the maximum number of elements that can simulate. The limits are associated to hardware requirements (memory) to process if number of elements increase.	The agent paradigm allows to add or remove instances without limit from the model. Then, the limits are associated to hardware requirements (memory) to process when number of elements increase (there were made experiments with 2000 agents using 2 MB at RAM memory).
3	Interactive Depurator	Monitor simulation progress, stop and execute until a specific condition is true, show current state of elements, attributes and variables.	Extend allows to change on the fly its parameters and model logic. Grasp2000 requires that each robot reaches validation and configuration checks. DE3 can create a realtime interactive simulation using DE3's accurated geometry, and real process data.	It is possible to have distributed control, stop, pause and continue the execution of any element (agent) without affecting the global behavior of the system.
4	Model State and Statistics	Accessible at any time during Simulation Model State.	AutoMod, Extend, Arena, Factor/aim, ProModel, MicroSaint, Quest and Witness generate automatic standard output statistics.	It is possible to get data of any element (agent) at any time.
5	Authorization during Simulation	Capability to change parameters and to execute models without changing the logic.	All the evaluated tools except Extend and DE3, allows to change its parameters on the fly.	The speed change affects all model elements (agents), however, it is possible to modify attributes at an individual level like the configuration of the animation depending to its state.

Table 3: Evaluation of Features related to the Execution Environment for Model Simulation

No.	Feature	Description	Evaluation	Agent Paradigm Added Value
1	Animation Type	Easy to scale model or use icons.	AutoMod, Grasp2000, ProModel, FlexSim, DE3, MicroSaint, Extend, Arena offers predefined icons and changes of color to identify current state of elements.	Each agent has a predefined icon, and its configuration can be modified at individual level at any time.
2	Dimension	2-D, 3-D.	Arena, Extend, FactorAim, MicroSaint, ProModel, Witness, offer 2-D animation. Quest, AutoMod, Grasp2000, FlexSim offer 3-D animation.	
3	Movement	Movement of entities or state indicators.	Most of the tools (Arena, AutoMod, EmPlant, FlexSim, Quest, Witness, among others) allows the movement of objects or blinking to show inputs and output at work centers.	
4	Library of common objects.	Predefined Graphics.	Most of the tools (Arena, MicroSaint, Quest, ProModel, Extend, among others) offers a library with predefined icons to represent machines, materials, and equipment.	It offers a set of predefined graphs (icons) for each type of element(worker, staff, machines, tools, production orders, semiproducts, products.
5	Show Progress	Control of animation speed.	All tools allows to define simulation speed at the beginning, but no change it during simulation.	The Simulator Agent coordinates animation speed, it also updates data and graphical animation, it can also update data and graphical animation, and send data to all agents modifying their execution speed.
6	Object Selection	It requires to show the dynamic status and statistics of an object by selecting it.	Arena, Promodel, Quest, Witness, FlexSim, allow to observe current state of elements during simulation.	When an element is selected its attributes and the active behavior of the agent can be shown.
7	Hardware Requirements	Standard or special video device, RAM requirements.	Grasp2000, DE3, Enterprise Dynamics Dynamics (2014), Flexsim, require a specialized hardware due to they can control robots, conveyors, GVS.	In order to scale-up a simulation model thanks to the MAS approach it is possible to increase the capacity adding new computer resources to the network which is being used for the simulation in an easy way. In order to connect simulation environment with physical equipment it is required to add and connect the specialized hardware in the same as with conventional tools.

Table 4: Evaluation of Features related to the Animation and Distribution of Model Objects

No.	Feature	Description	Evaluation	Agent Paradigm Added Value
1	Scenario Manager	Create scenarios for simulation.	Factor/aim allows to define manufacturing decision support scenarios. DE3 allows to include warehousing, distribution scenarios. Flexsim, AutoMod, Extend, MicroSaint, ProModel allow to include material handling systems. Grasp2000 allows to define robotics systems.	It is possible to include an Agent that manages the creation/modification of scenarios that allows to create models with a wide range of variables.
2	Execution Manager	Generate simulations and store results for subsequent analysis.	All evaluated tools store simulation results for their analysis.	An Agent that controls and manages simulation and animation can be included in order to monitor the agents tasks. The results generated by the agent behaviors are graphically presented and stored during simulation and used to generate reports and graphs.
3	Optimization	Use of genetic algorithms, among others techniques.	FlexSim offers an experimenter tool to simulate 'what-if' scenarios. Witness offers an optimizer to improve models, it has a process mapping tool from knowledge. AutoMod provides Autostat that allows to design experiments.	The negotiation techniques allow to improve the resource usage by applying complex algorithms in the same simulation.
4	Standard Reports	Final reports.	Arena, Extend, Factor/Aim, MicroSaint, ProModel, AutoMod offer automatic standard output statistics. Enterprise Dynamics offer report customization.	
5	Statistical Analysis	Trust Intervals, designed experiments.	Arena provides an input /output analyzer with enhanced statistics. Witness, AutoMod and Quest provides enhanced statistics.	It is possible to get more detailed statistical analysis in order to compare the results.
6	Business Graphs	Bar Graphs, Time Lines.	Promodel provides graphical analysis. Quest provides generation of CAD drawings. FlexSim provides charts, graphs and histograms. AutoMod provides gantt charts.	
7	Cost Module	Cost based on activities.	All evaluated tools offer reports about cost of processes. Quest and Factor/Aim provides a cost modelling and simulation ABC.	
8	File Exportation	Use of spreadsheets or Databases. It is used for customized analysis.	AutoMod uses IGES cimtechnologies to share data (factory-flow, Plan and CAD). Grasp2000 allows to use VRML, .avi. EnterpriseDynamics, Flexsim uses ODBC, CAD, Active X, and ODBC objects. DE3, Factor/Aim, MicroSaint, uses data file transfer. Arena, Extend and Promodel uses ODBC, OLE among others.	

Table 5: Evaluation of Features related to the Experimentation and Model Analysis

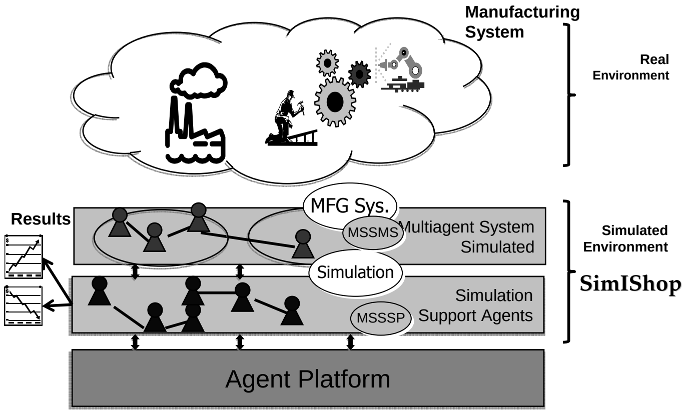


Figure 1: Simulation Architecture

3. Agent-based Simulator Architecture

In this section, the abstract architecture of SimIShop related to the basic functions of the agent-based simulation tool is illustrated.

The goal of the Simulation Architecture is the representation of a Real Environment of a Manufacturing System into a Simulated Environment that is supported by SimIShop (Figure 1). The SimIShop's architecture is composed by two layers. The upper layer represents the Manufacturing System that is simulated by a Multiagent System (MSSMS). The lower layer represents the simulation support agents organized into a Multiagent System (MSSSP). Then, these Multiagent Systems use an Agent Platform to execute their actions. Any state of the art Agent Platform[1] can be used.

The Simulation Process is based on the set of steps in a simulation study and the principles of Simulation Modeling proposed by Banks et al. (Banks, 1998). SimIShop provides interfaces for the Simulation Process, and is divided into two phases: *I) Model Creation, and II) Model Simulation* (Fig.2).

[1] Some available agent platforms include: JADE (JADE, 2005), AMELI (Esteva et al., 2004), Agent Development Kit (BV and http://www.tryllian.com/, 2005), April (Dale, J. and Knottenbelt J. and http://sf.us.agentcities.net/aap/index.html, 2005), Comtec (Information-Technology Promotion Agency and Communication Technologies, 2005a), Grasshopper (Grasshopper and http://www.grasshopper.de/, 2005), JACK (Information-Technology Promotion Agency and Communication Technologies, 2005b), JAS (Fujitsu et al., 2005), LEAP (LEAP and http://leap.crm-paris.com/, 2005), and ZEUS (Labs and http://193.113.209.147/projects/agents/zeus/, 2005)

In Phase I, a model is created by the User. This model is then used as input to Phase II, where the model is simulated by animating the behavior of its elements.

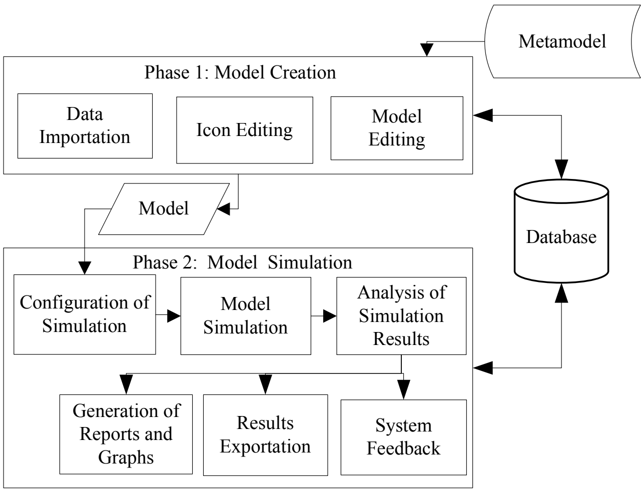


Figure 2: Main scenarios that appear at Phases of Simulation Process.

In order to implement SimIShop as an Agent-supported Simulation Tool, there is a Multiagent System (MAS) that supports the Simulation Process (MSSSP, lower layer in Fig.1). This MAS includes agents for each simulation feature as shown in Figure 3. The model of the MSSMS is created by the support of this MSSSP during the model definition (Phase 1). In Phase 2 the MSSMS is instantiated into an new MAS which represents the manufacturing system that is being simulated. After the instantiation, agents of MSSSP such as Event Generator, Speed controller, interact with the agents of MSSMS to control and manage the simulation.

The features of SimIShop were selected from a detailed list of collected features specified by Nikoukaran in (Nikoukaran and Paul, 1999) in order to provide an efficient simulation of a manufacturing scenario. The selected features are directly related to operational simulation functions. These features were assigned to the agents that implement SimIShop. The detailed descriptions of the actions that are incorporated as functions of each one of the agents are described as follows:

The Import/Export Analyzer allows to import the information related to equipment at external databases, and translate information into an appropri-

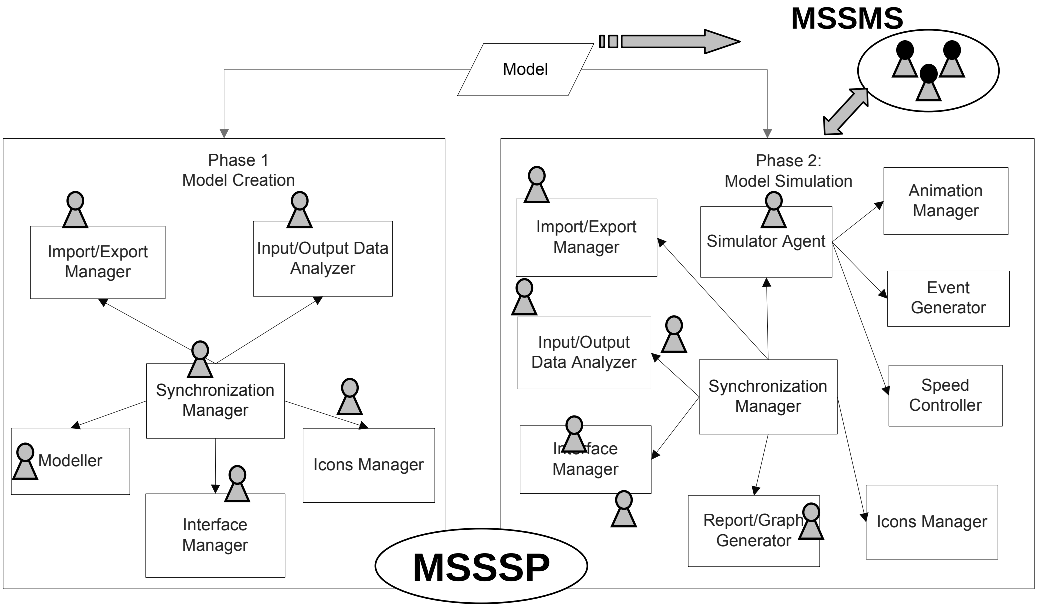


Figure 3: Agents that interact at Simulation Phases at the MSSSP.

ate structure. It also input translated information into the system according to the simulation agents and the resources model. Finally, it exports data information into a generic structured file to be used to another software. The Input/Output data Analyzer is in charge of analyzing imported information and classify and integrate this analyzed information as part of the agents of the systems. It analyze the results obtained after a simulation so that they can be reused by the system agents. The Modeler is responsible of building the model of the simulation by using database icons (libraries). It offers basic functional schemas to define the model and creates agents by defining their ontology, behavior, communication protocols. Also, it allows to edit a model, and verify whether the created model is free of mistakes (model checker). To do this, there are a predefined ontology that specifies communication rules, general behaviours that can be chosen, and protocols that can be defined according to the nature of the element/simulation agent (Product, Order, Resource, or Staff) that is being defined. The Icon Manager allows to add icons from an external file. It can create a new icon and suppress icons using a tool bar. Moreover it also edit predefined icons by associating features to the icon such as: behavior, ontology, and communication protocols taking into account the nature of the element/simulation agent (Product, Order, Resource, or Staff) that is being defined. The Simulator Manager is in charge of controlling traces and steps of simulation based on the time controller. At the end of a simulation it can manage and display output information of the simulation. The Simulator Agent can be executed in three different modes (agent role):

At the Event Generator Role the agent can produce additional information to the initial state of the system. This information is used to input dynamic data during simulation such as: noise (temperature change), add new equipment or human resources. Also it can activate/deactivate agent instances or roles according to the current state of the system (specific time interval or conditions -progress of production order-, the received messages from Manufacturing System Agents). Thus, the human User interacts with this role agent during simulation.

At the Animation Manager Role the agent is in charge of animating the Simulation Model according to the generated events by the Event Generator Role. During simulation it controls traces and steps of simulation based on the time controller. Thus, the human User can modify the initial configuration to animate the Manufacturing System Agents.

The Speed Controller Role allows to define the rhythm of the simulation, it can accelerate or slow down simulation time to observe results. Moreover, it also configures the global clock. Thus, the human User specifies the execution speed. The Report Generator agent is responsible of creating templates of business reports and graphics. This agent can edit report and graphic templates, and generates reports and graphics based on the produced simulation output. The Synchronization Manager is in charge of monitoring the current system state. This agent requests the execution of specific agents, kill them, and activate/deactivate them to guarantee the system stability. Finally, the Interface Manager offers modules to create or modify current interfaces. This feature allows to scale up and to adapt an interface to new requirements in an easy way without disrupting the current structure.

The MSSMS model created in Phase I is based on a manufacturing system metamodel which is defined in next section.

4. Metamodel to Simulate an Intelligent Manufacturing System

SimIShop uses a metamodel in order to define a MSSMS model. This metamodel is based on the PROSA Reference Architecture (Brussel et al., 1998) and ANEMONA (Botti and Giret., 2008). The metamodel includes specific simulation features related to animation, internal and external interaction, negotiation strategies, capabilities, norms, shop floor configurations, instance type (capability grade/quality), distributed planning. There are four types of elements (see Fig. 4): orders (production orders), products, resources and staff. The Orders manage the production of Products and use Resources. The Resources have four specializations: Workers, Machines, Tools, and Raw Material. The Staff assists to Orders and Resources during production of orders. Thus, the metamodel elements represent agents in a MSSMS model, except raw material.

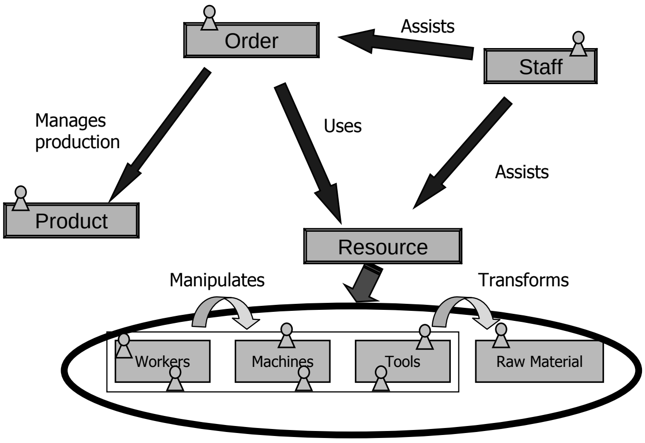


Figure 4: Agents and their interaction in the Metamodel of an Intelligent Manufacturing System

The metamodel includes those elements that are identified as key attributes in the specialized literature related to Manufacturing Systems (MCLean and Leong, 2001). The work on Product Model (Magnusson and Holm, 1996) is used in order to define the structure of a product, the production processes, and the resources. The works on Product State Model (PSM) (Larsen et al., 1997, 1999) are used to define the input data and the production process inputs by monitoring the predefined product attributes. These attributes are linked to products and/or subparts during the production process. Also, the work from (Larsen et al., 2001) is used in order to define the objects for triggering simulation events. Moreover, other specific attributes that are required for simulation (management of icons, speed, animation) have also been included.

Some important features of this metamodel are: The scope is limited to

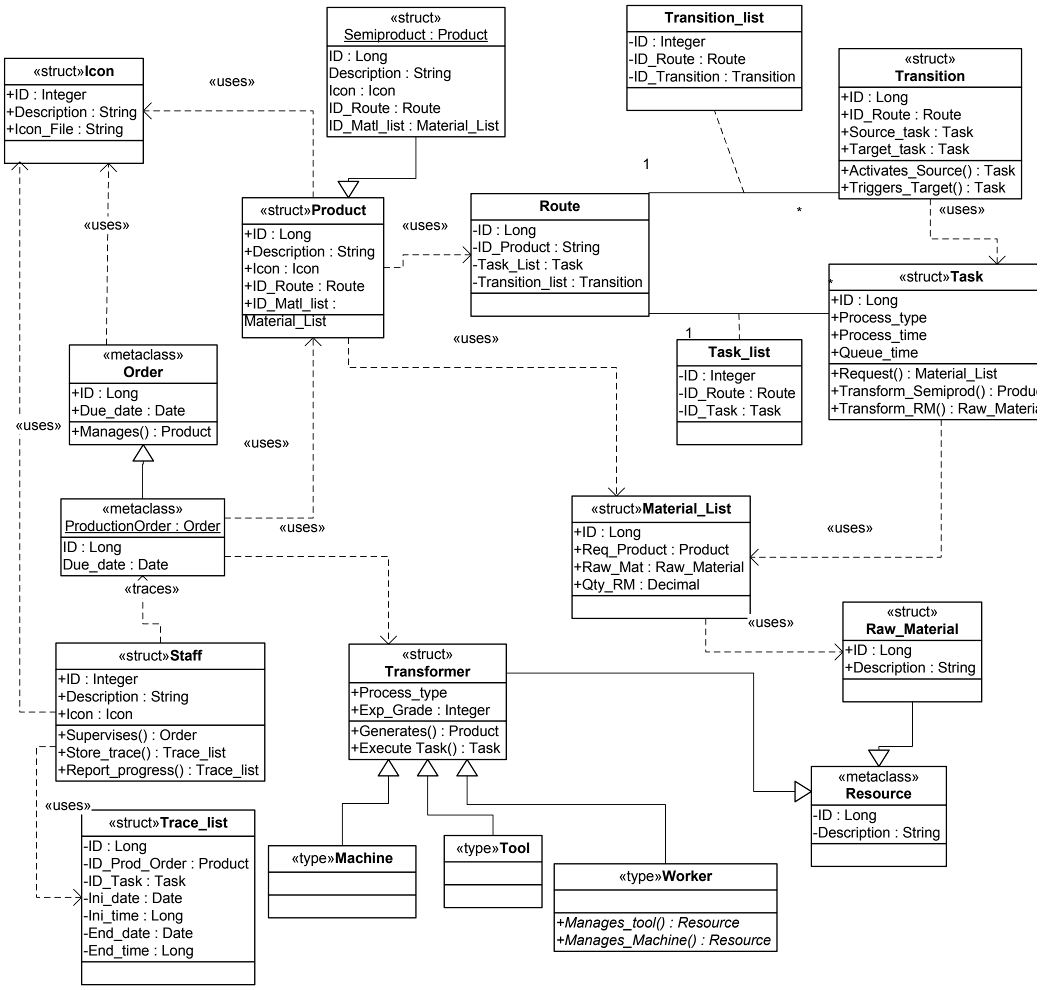


Figure 5: High level of proposed Metamodel for the Simulation of Intelligent Manufacturing System.

the production phase. The features linked to the material state, product entity and worker are included as part of the properties and objects of the agent specializations (artifact). The metamodel considers each one of the artifact instances according to its route during the evolution of the semi-product until it becomes a product (event).

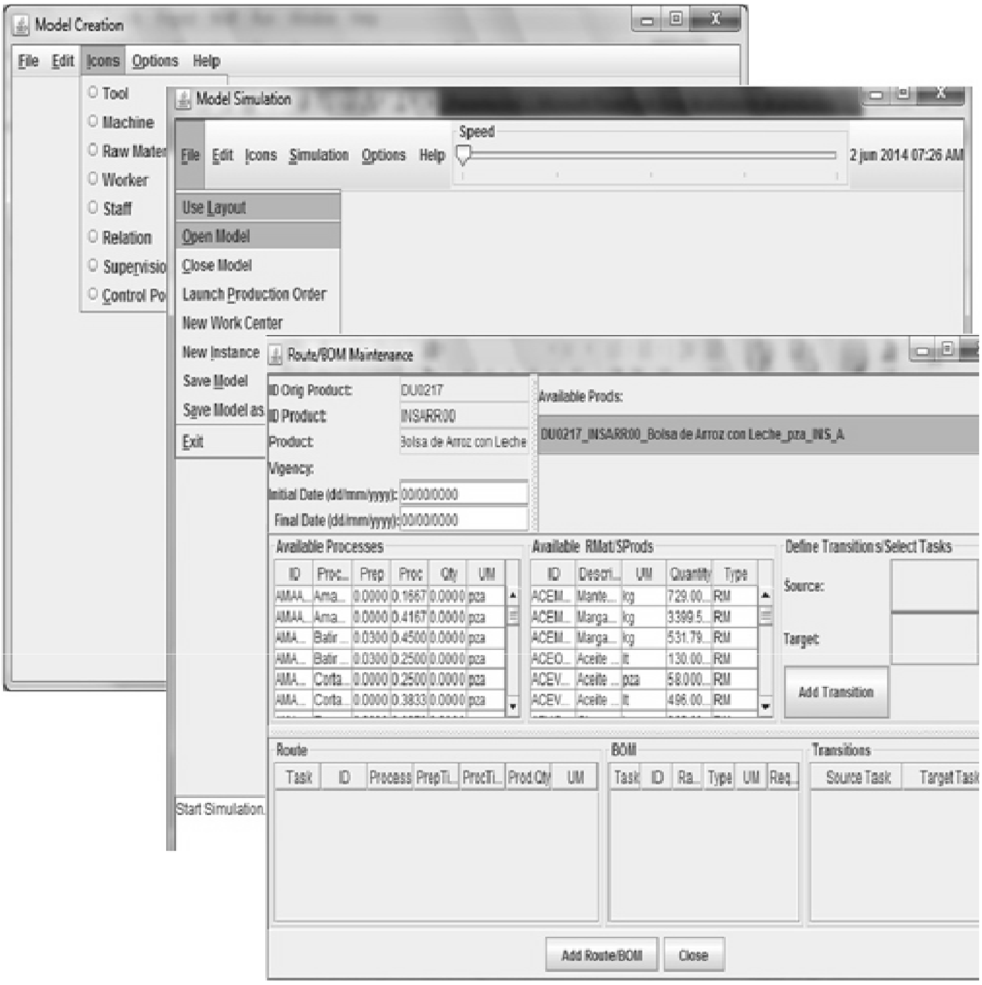
Figure 5 shows the relationships that are allowed among the different modeling entities. A Product uses Resources (Worker, Machine, and Tools) to make transformation processes over Raw Materials and SemiProducts and produce a new Product according to its Route and its Bill of Materials. Resources (Machine, Tools) can operate by themselves or can be manipulated by a Worker. A Production Order controls the production of a Product according to its Route. A Maintenance Order controls the maintenance services made to Resources (Machines, Tools). Thus, the Staff supervises the actions made by the Resources, Products and Orders. The attributes of the entities are shown in Fig. 5 by means of objects aggregated to them. See for example the attributes of a Product which includes: an ID,a Description, a Route (which in turn is a complex object described by an aggregation) of other objects.

5. SimIShop: The simulation tool

SimIShop is the simulation tool that implements the agent-based architecture of Section 3 . In this section the details of how every think work in the tool is described. The description is made in terms of the agent-based algorithms and negotiation protocols that are executed in order to implement the different functions of the tools in the two simulation phases: Model Creation (Phase I) and Model Simulation (Phase II). Figure 6 shows the User interface of SimIShop.

The technology used to implement the SimIShop includes: a) JAVA (Ver. 6.x), a standard programming language, b) JADE (Ver. 3.4), an agent development framework based on JAVA, which provides basic schemas to define agents and communication protocols, c) ECLIPSE (Ver. 3.2.2), a development framework that manages JADE, d) PROTEGÉ-OWL (Ver. 4.x), an ontology editor, and e) MySQL (Ver. 5.0), which provides a multi-threaded, multiuser and robust SQL database server (Structured Query Language).

Some representative features of the prototype are the flexibility and proactivity to interact and use the model resources and the information generated



ID Orig Product: DU0217

ID Product: INSARR00

Product: Bolsa de Arroz con Leche

Vigency:

Initial Date (dd/mm/yyyy): 00/00/0000

Final Date (dd/mm/yyyy): 00/00/0000

Available Prods:

DU0217_INSARR00_Bolsa de Arroz con Leche_pza_INS_A

Available Processes

ID	Proc...	Prep	Proc	Qty	UM
AMAA...	Ama...	0.0000	0.1667	0.0000	pza
AMAA...	Ama...	0.0000	0.4167	0.0000	pza
AMA...	Batir ...	0.0300	0.4500	0.0000	pza
AMA...	Batir ...	0.0300	0.2500	0.0000	pza
AMA...	Corta...	0.0000	0.2500	0.0000	pza
AMA...	Corta...	0.0000	0.3833	0.0000	pza

Available RMat/SProds

ID	Descri...	UM	Quantity	Type
ACEM...	Mante...	kg	729.00...	RM
ACEM...	Marga...	kg	3399.5...	RM
ACEM...	Marga...	kg	531.79...	RM
ACEO...	Aceite ...	lt	130.00...	RM
ACEV...	Aceite ...	pza	58.000...	RM
ACEV...	Aceite ...	lt	496.00...	RM

Define Transitions/Select Tasks

Source:

Target:

Route

Task	ID	Process	PrepTi...	ProcTi...	Prod.Qty	UM

BOM

Task	ID	Ra...	Type	UM	Req...

Transitions

Source Task	Target Task

Figure 6: Snapshot of the User Interfaces of SimIShop.

during the simulation process with easy-to-use interfaces. Another significant feature included in the simulation tool is the scalability. This feature is provided by the Interface Agent which allows the User to scale up the tool according to its needs.

The following sections details the implementation of the simulation tool in terms of agent-based algorithms and cooperation scenarios to support the two simulation phases.

5.1. Phase I: Simulation Model Specification

This section presents the detailed activities that the Agents execute during the Model Creation (Phase I). The tool allows a Model of an Intelligent Manufacturing System (Fig. 2) to be defined, based on the metamodel of Section 4. This section also presents the main features of the modeling supported by the agent interactions such as: how the metamodel is used during the model creation; the interaction between the metamodel and the Agent-supported Simulator during the Simulation Process; the initialization process for the database, the criteria to help the User to create a model; how the scenarios are defined in a model, and; how specific resource instances are linked to the model icons and the rules related to the organization modeling that are applied.

5.1.1. Agent interaction during Model Creation

In Phase I, the agents that interacts in the Simulation Process are (Fig. 2): Import/Export Manager (IEM), Input/Output Data Analyzer (IODA), Event Generator (EVEG), Modeler (MOD), Icon Manager (ICOM), Synchronization Manager (SYNM), and the Interface Manager (INTM). Figure 7 shows the main interactions between agents during Model Creation (phase 1).

In this phase, the User creates a Model and the Synchronization Manager supervises the model creation. The Synchronization Manager determines when it is necessary to create an instance of each type of role according to the current state of the system. Then, the Import/Export Manager translates the information from the real world that the User provides. The Input/Output Data Analyzer analyzes this information and formats it for internal usage. Then, the Icon Manager includes this information in the icons (the information is classified and linked to specific icon attributes). The User then uses

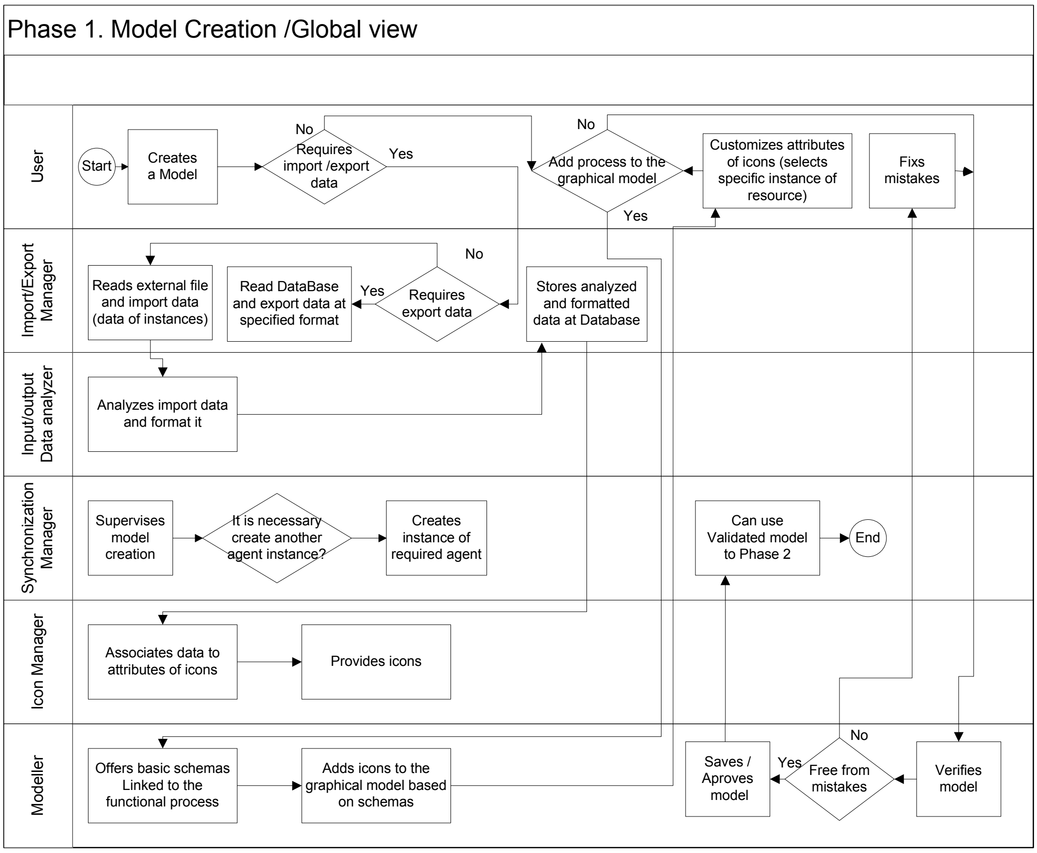

Phase 1. Model Creation /Global view
User
Import/Export Manager
Input/output Data analyzer
Synchronization Manager
Icon Manager
Modeller
Start
Creates a Model
Requires import /export data
No
Yes
Add process to the graphical model
No
Yes
Customizes attributes of icons (selects specific instance of resource)
Fixs mistakes
Reads external file and import data (data of instances)
Read DataBase and export data at specified format
Yes
Requires export data
No
Stores analyzed and formatted data at Database
Analyzes import data and format it
Supervises model creation
It is necessary create another agent instance?
Creates instance of required agent
Can use Validated model to Phase 2
End
Associates data to attributes of icons
Provides icons
Offers basic schemas Linked to the functional process
Adds icons to the graphical model based on schemas
Saves / Aproves model
Yes
Free from mistakes
No
Verifies model

Figure 7: Agent interactions among agents during Phase 1.

the Modeler to incorporate the icons into the Model. The Modeler offers basic schemas that are related to the functional process in order to include the icons (which are provided by the Icon Manager). After an icon is included in a Model, the User customizes its attributes according to the manufacturing system that it needs to simulate (restrictions such as capacity, and availability). When the User has finished the model, the Modeler verifies if the model is free of mistakes. If there are any mistakes, the Modeler informs the User who can edit the model and then the Modeler verifies it again. When the Model is free of mistakes, the Modeler informs the Synchronization Manager that the model is ready to be used in *Phase II*. Thus, the connecting element between the two phases is the Model of the manufacturing system that the system engineer needs to simulate.

5.1.2. Initializing the Database

The database of the prototype is initialized by importing data from the real shop floor of the manufacturing system. Data includes: workers, machines, tools, staff and products. The main goal of a manufacturing system is the production of products. The algorithm used to select and optimize selection of resources is based on the Analytic Hierarchy Process (AHP) Saaty (1988); Mordeson et al. (2013). The AHP is a structured technique for organizing and analyzing complex decisions, based on mathematics. It helps decision makers find one that best suits their goal and their understanding of the problem. It provides a comprehensive and rational framework for structuring a decision problem, for representing and quantifying its elements, for relating those elements to overall goals, and for evaluating alternative solutions. Then, initial setup of the MAS takes into account following criteria: search to do processes at the lower time, avoid bottlenecks during assignation of tasks to resources, and to process with the lower cost and better quality as parameters when a production order is launched to the shop-floor.

During Database Initialization User requests import data from Real Shop floor of specific type of entity (workers, machines, tools, staff, processes, raw materials). Synchorization Manager creates instances of Import/Export Manager and Input/output Analizer. The data must be in a formatted file which Import/Export Manager reads and imports into the system.Then Input/output Analizer validates data (taking into account the specified parameters) and if they are free of mistakes he informs to Import/Export Manager whom stores the analyzed and formatted data at database. Figure 8 shows the main interactions between agents during Initialization of Database.

Therefore, in order to to define a product it is required be specified its routes and bill of materials.

In addition, a User can request the creation of products. To do so the User requests to Modeler to create a product. The Synchorization Manager supervises product definition, if a User request the creation of different models at the same time, then it creates instances of a new Modeler. Thus processes to define the route operations (not linked to resources). Then, the User requests to Modeler the available raw materials, in order to assign them to a bill of materials that can be linked to each route operation. Finally, the User defines the sequence of processes by defining transitions among operations route. Moreover, the prototype allows to execute parallel operations by defining transitions among operations route. Figure 9 shows the main interactions between agents during Product Definition.

In SimIShop it is possible to define complex behaviors that appear in a real manufacturing system, i.e. process N type of products at the same time (by launching N Production Orders in a batch), assign task to resources according to their expertise level (it can be defined by the User manually or he can request the assignation by negotiation when a Production order is created), applying algorithms that take into account the final quality, cost and processing time to select the best one (it can be defined when a Production order is created), and maintaining the resources with a balanced workload without traspassing installed capacity.

5.1.3. Creating a Model

During the Model creation, there is continuous interaction between the User and the Simulation Tool by means of graphical interfaces. The criteria that are taken into account to help the User (Model Designer) during the design of the Model include:

- Providing the User with different help options for adding elements during the model design.
- Defining an appropriate level of detail which allows the User to include basic information that could help to easily describe real issues in the Manufacturing System that is represented.
- A functional design of the Database to efficiently store and recover the information.

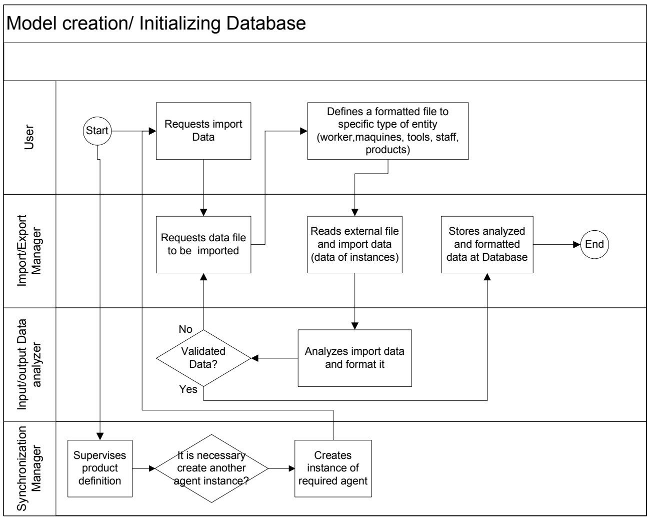

Model creation/ Initializing Database
User
Import/Export Manager
Input/output Data analyzer
Synchronization Manager
Start
Requests import Data
Defines a formatted file to specific type of entity (worker,maquines, tools, staff, products)
Requests data file to be imported
Reads external file and import data (data of instances)
Stores analyzed and formatted data at Database
End
No
Validated Data?
Yes
Analyzes import data and format it
Supervises product definition
It is necessary create another agent instance?
Creates instance of required agent

Figure 8: Agent interactions for Initializing DataBase.

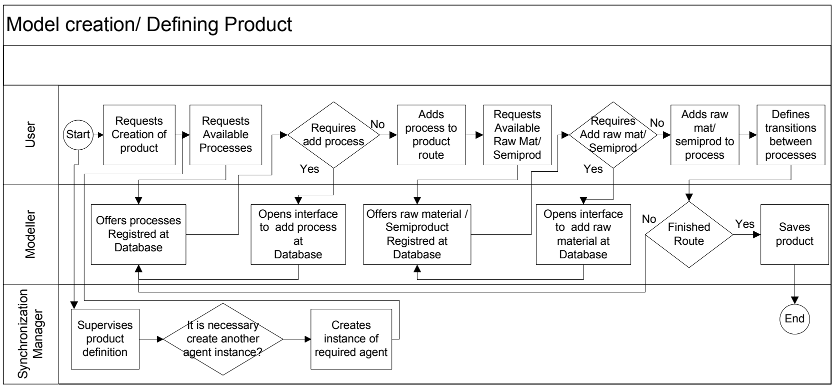

Model creation/ Defining Product
User
Modeller
Synchronization Manager
Start
Requests Creation of product
Requests Available Processes
Requires add process
No
Yes
Adds process to product route
Requests Available Raw Mat/ Semiprod
Requires Add raw mat/ Semiprod
No
Yes
Adds raw mat/ semiprod to process
Defines transitions between processes
Offers processes Registred at Database
Opens interface to add process at Database
Offers raw material / Semiproduct Registred at Database
Opens interface to add raw material at Database
No
Finished Route
Yes
Saves product
End
Supervises product definition
It is necessary create another agent instance?
Creates instance of required agent

Figure 9: Agent interactions for Defining a Product.

- Interfaces where the User can easily understand the kind of information required and the results obtained.

During Phase I, the event detection method is applied to detect the state changes in the system (Carrascosa, 2004). Thus, the Modeler Agent reacts to the presence of events. Events can be triggered both when the User requests an action and when the Modeler has finished specific tasks.

When User is defining a model, User requests Modeler add a new scenario, and it adds icons (worker, machines, tools, staff) to the graphical model. This process is repeated until User finish to add scenarios. The Modeler verifies and saves model if it is free of mistakes, otherwise, it informs User whom can fix mistakes. Figure 10 shows the main interactions between agents during Model Creation.

5.1.4. Including Scenarios in a Model

During the Creation/Edition of a Model, the Modeler Agent offers the User basic schemas in accordance with the scenario type (i.e. cut, welding, finishing, moulding, thermic, chemical). These scenarios can be included in the model. The Modeler Agent proposes a schema that includes the type and minimum quantity of elements suggested (i.e. workers, machines, tools) and also the human resources who supervise the activities (Staff).

Figure 11 shows the main interactions between agents during Scenarios Addition.

A Scenario requires inputs that are processed to produce outputs. Therefore, both Raw Material and SemiProduct elements must be included in the

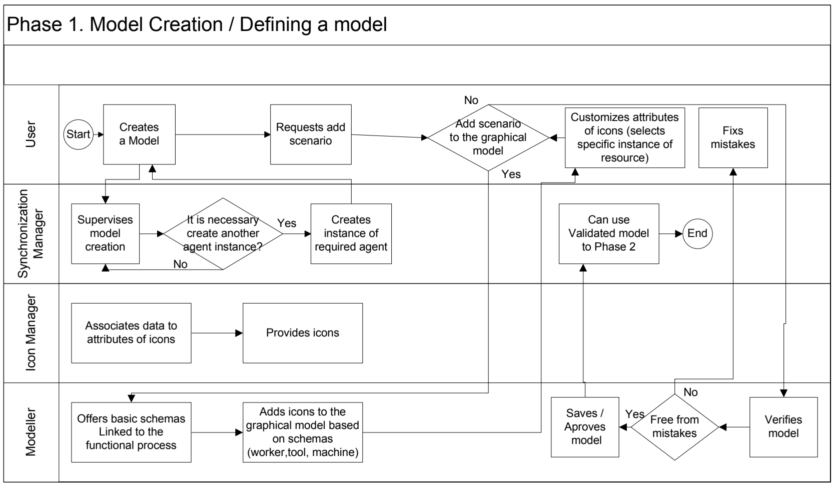

Phase 1. Model Creation / Defining a model
User
Synchronization Manager
Icon Manager
Modeller
Start
Creates a Model
Requests add scenario
No
Add scenario to the graphical model
Yes
Customizes attributes of icons (selects specific instance of resource)
Fixs mistakes
Supervises model creation
It is necessary create another agent instance?
Yes
No
Creates instance of required agent
Can use Validated model to Phase 2
End
Associates data to attributes of icons
Provides icons
Offers basic schemas Linked to the functional process
Adds icons to the graphical model based on schemas (worker,tool, machine)
Saves / Aproves model
Yes
Free from mistakes
No
Verifies model

Figure 10: Agent interactions for Defining a Product.

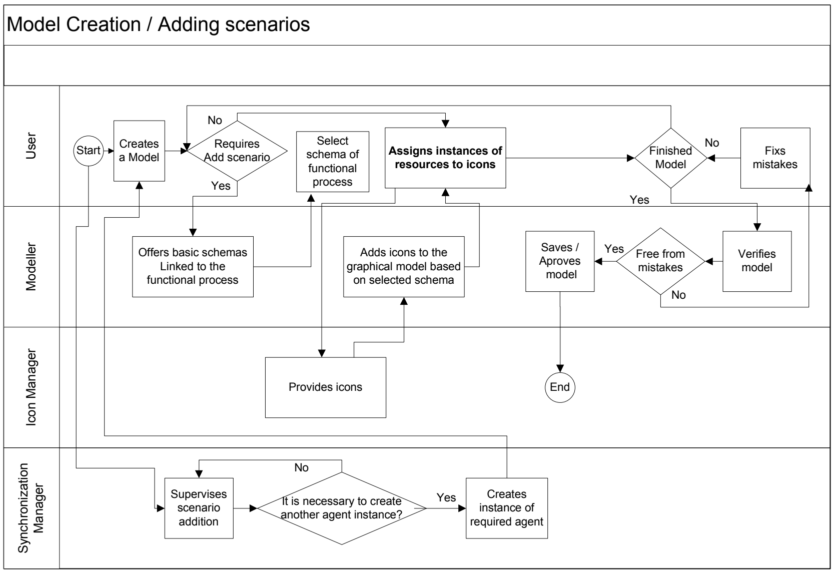

Model Creation / Adding scenarios
User
Modeller
Icon Manager
Synchronization Manager
Start
Creates a Model
No
Requires Add scenario
Yes
Select schema of functional process
Assigns instances of resources to icons
Finished Model
No
Yes
Fixs mistakes
Offers basic schemas Linked to the functional process
Adds icons to the graphical model based on selected schema
Saves / Aproves model
Yes
Free from mistakes
No
Verifies model
Provides icons
End
Supervises scenario addition
No
It is necessary to create another agent instance?
Yes
Creates instance of required agent

Figure 11: Agent interactions for Including Scenarios.

basic Schemas. In this way, the User can create a model with these basic schemas. This model can be modified by including additional elements according to User's needs. Finally, the goal of the Staff is to ensure that the manufacturing processes that are controlled by an Order Agent (Production Order) are fulfilled, as well as to ensure that the Resource Agent (Human Resources, Machines, Tools) are used appropriately during those processes. The Staff also makes sure that the interactions among the Order and Resource Agents are fulfilled in accordance with their previous negotiations during the manufacturing processes. The goal of the negotiation is to optimize Resource usage during a Task according to the Order restrictions (capacity vs time and quality).

5.1.5. Allocating Resources in a Scenario

In comparison with current Simulation Tools, SimIShop allows the User to design a customized Model that can allocate a specific instance which is linked to each one of the elements whenever it is required. The Modeler offers instances of the resources that can execute the indicated activity in the scenario (i.e., Resource: Machine Activity: Cut, Machine: Cutter machine, Molding machine). Then, the User can choose specific instances of workers, machines, tools and supervisors to be included into the scenario. Thus, the agents that represent the Resources during Simulation will offer their capacity according to the specifications of the selected instance. The User can add new Resources, if the User does not want to use the resources that are available. When the User requests the creation of a new resource, the Synchronization Agent triggers a new instance of an Icon Manager, which helps the User to create the resource properly. Figure **??** presents a graphical view of the created model. Figure 12 shows the main interactions between agents during Resources Assignation.

The route operations require specific processing time. The shop floor has different resources that can execute these operations with different capability grades (Beginner, Expert) that allow to process a product with different processing time, quality and cost. According to Dreyfus Model of Skill Acquisition Dreyfus and Dreyfus (1980, 1986), it divide skills level at five stages of increasing skill as follows: Novice, Advanced beginner,Competent, Proficient, and Expert. In the novice stage , a person follows rules as given, without context, with no sense of responsibility beyond following the rules exactly. Competence develops when the individual develops organizing principles to quickly access the particular rules that are relevant to the specific

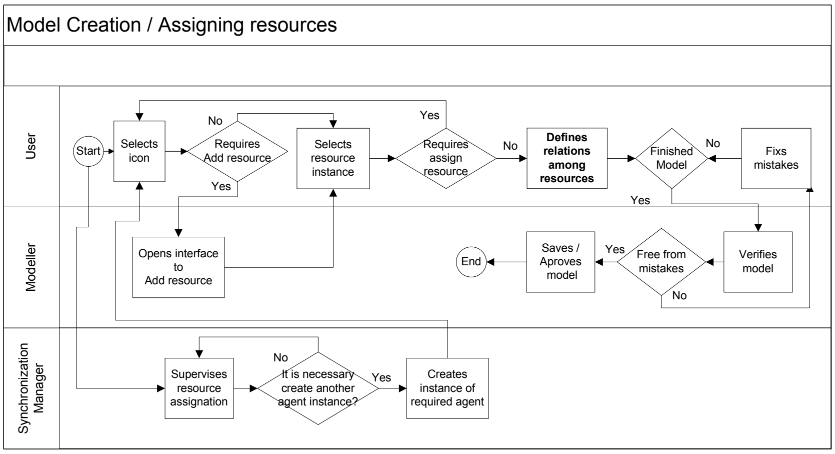

Model Creation / Assigning resources
User
Modeller
Synchronization Manager
Start
Selects icon
No
Requires Add resource
Yes
Selects resource instance
Yes
Requires assign resource
No
Defines relations among resources
Finished Model
No
Fixs mistakes
Yes
Opens interface to Add resource
End
Saves / Aproves model
Yes
Free from mistakes
No
Verifies model
Supervises resource assignation
No
It is necessary create another agent instance?
Yes
Creates instance of required agent

Figure 12: Agent interactions for Allocating Resources.

task at hand; hence, competence is characterized by active decision making in choosing a course of action. Proficiency is shown by individuals who develop intuition to guide their decisions and devise their own rules to formulate plans . The progression is thus from rigid adherence to rules to an intuitive mode of reasoning based on tacit knowledge. Thus, based on this model, the proposal consider three skill levels Beginner (novice), Medium (competent) and Expert. Then, based on their main attributes and **the standard of work** expressed at Scale Novice-to-Expert of Professional standards for conservation from the Institute of Conservation Conservation (2003), there were assigned a "number" that can be used to measure their behavior. According to the capability grade of resources (skill level), an Expert can decrease the standard processing time with a better quality grade and a Beginner can increase that time with lower quality (Table 6). Then, the User can takes into account this feature during resource allocation.

5.1.6. Rules related to organization modeling

SimIShop interacts with the model created in Phase I by means of scenarios (Use Case Interactions) during the simulation process in order to fulfill specific simulation functions of the tool. Before a Model can be simulated, it is verified by the Modeler Agent. The verification is based on the Modeling Organization Theory (Wagner and Hollenbeck, 2004). The basic rules used to verify a scenario include:

1. If a machine or tool must be manipulated by a worker, a worker must

	Standard Configuration		
Selected Type	Beginner	Medium	Expert
Beginner	-	Time +30% Cost −30% Quality −20%	Time +50% Cost −50% Quality −40%
Medium	Time −30% Cost +30% Quality +20%	-	Time +20% Cost −20% Quality −20%
Expert	Time −50% Cost +50% Quality +40%	Time−20% Cost +20% Quality +20%	-

Table 6: Optimization of Processing Time, Cost and Quality according to the Instance Type.

manipulate the machine or tool. Otherwise, if a machine or tool works automatically, no worker must manipulate that machine or tool. (Behavior norm).

2. A worker instance cannot be linked to two different icons in the model. (Capability norm)
3. A machine or tool instance can be linked to different icons as much as instance number of machine or tool are defined in the database.
4. All machine, tool, worker and staff icons must be linked to an instance to delimit their capacities. (Capability norm)
5. When a relation is defined the rules include (Behavior norms): a) if relation type is "manipulate", a worker only can manipulate machines or tools, and b) if relation type is "supervise", a staff supervise workers, machines and tools. This specialization allows during model creation to avoid a staff which is linked to a resource and at the same time have to execute a production process, or a worker without any capability to supervise an area.

These rules ensure that the model has enough information for the simulation, rules 1 to 5 are used when a model is created (Phase I). If some are not fulfilled, the model cannot be simulated. The relationships among production order, workers, machines and tools delimit the model simulation (Phase II).

Rules are applied during model creation when relations among resources are defined. For example, is User selected an icon related to a worker or staff and Modeler offers available relations (according to rules: manipulate,supervise) then User can only select as resource target -machine,tool-

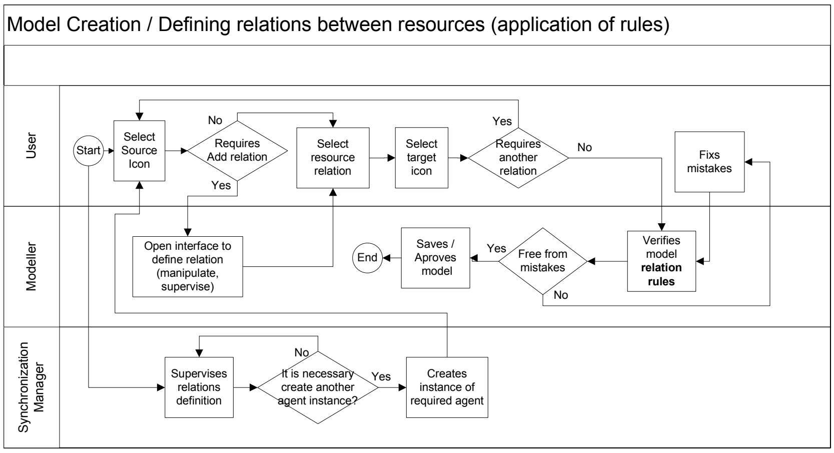

Model Creation / Defining relations between resources (application of rules)
User
Modeller
Synchronization Manager
Start
Select Source Icon
No
Requires Add relation
Yes
Select resource relation
Select target icon
Yes
Requires another relation
No
Fixs mistakes
Open interface to define relation (manipulate, supervise)
End
Saves / Aproves model
Yes
Free from mistakes
No
Verifies model **relation rules**
Supervises relations definition
No
It is necessary create another agent instance?
Yes
Creates instance of required agent

Figure 13: Agent interactions for defining relations and how basic rules are applied.

which can be manipulated o supervised. Also, rules are applied when User selects a specific instance for an icon. An instance can not be linked to more than one icon. Figure 11 shows the main interactions between agents during Relations Definition and how norms are applied.

5.2. Phase II: Model Simulation

This section presents the detailed activities that the Agents execute during Model Simulation (Phase II). The model created and verified in *Model Creation -Phase I-*, is used as input in *Model Simulation -Phase II-*.

The Manufacturing System Model designed in Phase I, is internally an Agent Model that will be used to construct a new Multiagent System that interacts with the Simulation Tool (another Multiagent System) in Phase II (Fig. 3). The Synchronization Manager supervises the model simulation according to the current state of the System and the Model generated as a result of the first phase. The simulation of the Model is executed by the role set that is in charge of the simulation process. The Simulator Agent controls the activity of this role set. This set is composed of the Animation Manager, the Speed Manager, the Event Generator, the Verifier, the Report/Graph Generator, the Input/Output Data Analyzer, the Icon Manager, and the Import/Export Manager. The Agent Simulator receives the model and asks the User to configure the simulation for the model. The configuration includes the definition of the simulation speed, animation type, and stop condition. Then, the simulation starts. The selected speed affects the global clock that is used by all the agents. During the model simulation, the Simulator Agent (who plays the role of "Event Generator") triggers the event (i.e., execute/stop/wait a task, send a message, store results, request a graph or report), according to the global clock. Thus, the Agent Simulator informs the Speed Manager of the changes. Figure 14 shows the main interactions between agents during Model Simulation.

5.2.1. Launching Production Orders

When the User launches a production order the product, required quantity, required date, product route and the shop floor configuration that will be used to produce that order are defined. In this way, each production order has its own configuration. This feature provides more flexibility in order to design experiments and validate the User hypothesis. In addition, the configuration can be defined in two ways: a) the User can select specific resources linked to each one of the route operations or b) the User allows the order to negotiate with the available resources. For the last option a negotiation technique based on auctions has been implemented (Marsa-Maestre et al., 2009). In this way the Order negotiates with the resources according to the best delivery date, the best instance type or the best cost (lowest).

During Launching order, User requests Simulator Agent a product to be produced. then it offers available products and User selects it. User can determine assignation technique: Manual of automatic. If User select Automatic, Resources verify their capacity, if they have available capacity send a proposal to produce the product.

Figure 15 shows the main interactions between agents during Order Launching.

In addition, when the User launches a production order, the Manufacturing System executes its own tasks at the same time: production orders and resources are in contact for task assignation, the resources process accepted tasks and inform the production orders of the progress.

The Agent Simulator (Event Generator role) receives the state changes of the Manufacturing System (i.e. machine X accepts a task, worker Y is stopped, etc.) and translates them into events. The Agent Simulator (Event Generator role) informs the Animation Manager Agent of the event, who animates the icons linked to that event. The Agent Simulator stores the event results when an event is finished. The Agent Simulator (Event Generator role) can produce additional information, which the Synchronization Manager includes during simulation. The Input/Output Data Analyzer Agent

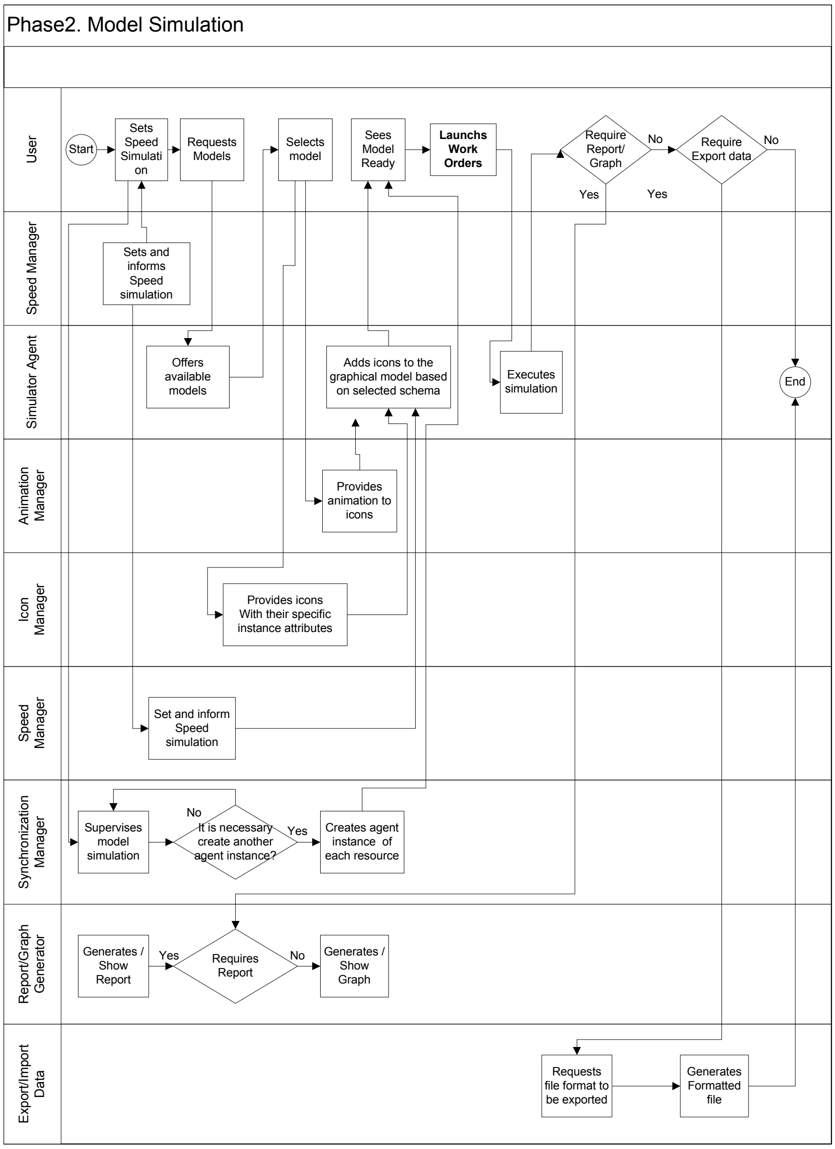


Figure 14: Main Agent interactions during Model Simulation.

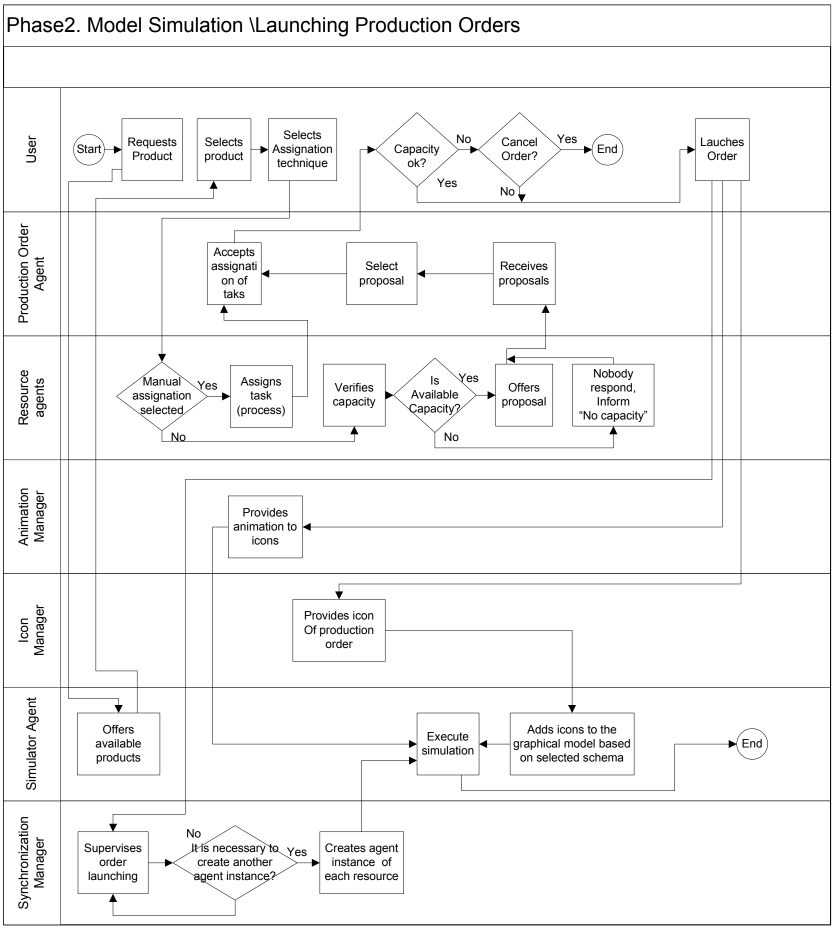

Phase2. Model Simulation \Launching Production Orders
User
Start
Requests Product
Selects product
Selects Assignation technique
Capacity ok?
No
Yes
Cancel Order?
Yes
No
End
Lauches Order
Production Order Agent
Accepts assignati on of taks
Select proposal
Receives proposals
Resource agents
Manual assignation selected
Yes
No
Assigns task (process)
Verifies capacity
Is Available Capacity?
Yes
No
Offers proposal
Nobody respond, Inform "No capacity"
Animation Manager
Provides animation to icons
Icon Manager
Provides icon Of production order
Simulator Agent
Offers available products
Execute simulation
Adds icons to the graphical model based on selected schema
End
Synchronization Manager
Supervises order launching
No
It is necessary to create another agent instance?
Yes
Creates agent instance of each resource

Figure 15: Main Agent interactions during Order Launching.

analyzes the results when the simulation has finished. The Icon Manager Agent reuses the analyzed information, which is linked and included in the icons. The Import/Export Manager Agent allows the User to export the information for external analysis.

5.2.2. Planning

The planning and scheduling optimization is a key feature of SimIShop. Each Production Order can use a different strategy for resource allocation for different type of products in the same execution. After a Production Order is launched, if the the direct allocation has not been selected, the negotiation strategies (i.e. Auctions((Duffie et al., 1998; Veeramani and Wang, 1997; Marsa-Maestre et al., 2009)), Case-based Reasoning(Choy and Lee, 2000),(Ashley, 2006)) are activated for the resource allocation process. The system allows to simulate the addition of a new work order while the capability of the resources to process them is not swamped. If the capability of the resource is swamped, the system informs the User that there is no possibility to process any additional order. During simulation each one of the Production Order Agents and Resource Agents of the Manufacturing System (MSSMS) are in charge of their own planning and scheduling (Fig. 16). These feature promotes the dynamic production planning, scheduling and finally the optimization of the use of resources, which is based on mathematical algorithms to made an AHP (Analytical Hierarchical Process) that includes cost, processing time, quality as parameters. The Resource Agents (workers, machines, tools) have a queue for controlling the production orders that they have accepted to be processed meanwhile the Product Order Agents have a queue for controlling the allocated resources to each step of the production route of their product (See Fig. 16).

5.2.3. Tracing Simulation

During Simulation it is possible to trace the current behavior of each one of the model elements, i.e. resource data and its current state. After a production order has been launched as an agent, this agent can directly assign the required capacity to agent resources or can negotiate with resources to assign it. The Simulator Agent monitors and stores these interactions among agents for tracing behaviour of agents. More over, the Simulator agent also monitors and stores the behaviour for building traces of resource agents when they are informing about the progress of production orders. Figure 17 shows the main interactions between agents during Tracing Simulation.

5.2.4. Finishing an Order

During order process, Resource agents inform about the progress to the Production Order agent. Thus, when Resource agents inform that last process has been finished, the Simulator Agent groups all tracing data and shows User an interface with final data (real processing time, resources and cost - that represent the used shop floor configuration-). Then the User scores the Order that has been processed by using a specific shop floor configuration. When a Production Order is finished, a new Product icon is also added.

Figure 18 shows the main interactions between agents during when a Production Order is finished.

5.3. Model Evaluation

The Agent Results allows the User to score Models that have been created in Phase I. In this way the models are evaluated and compared with others. The evaluation can be made before and after a model is simulated. A model is evaluated based on the instance type of the model elements. In an ideal model all resources are "experts". For this proposal it is assumed that a resource

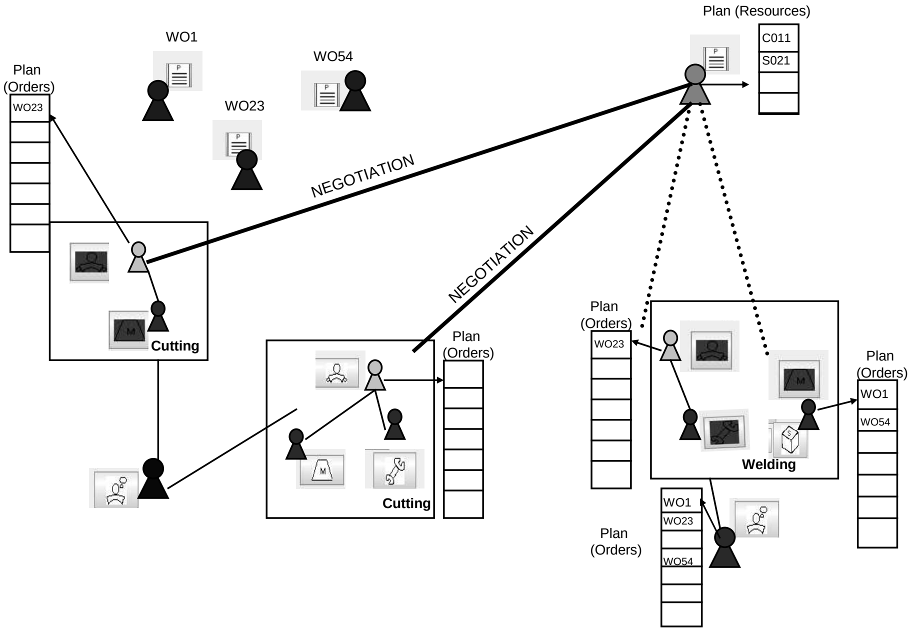


Figure 16: View of planning queues of Resources based on Dynamic and distributed Planning.

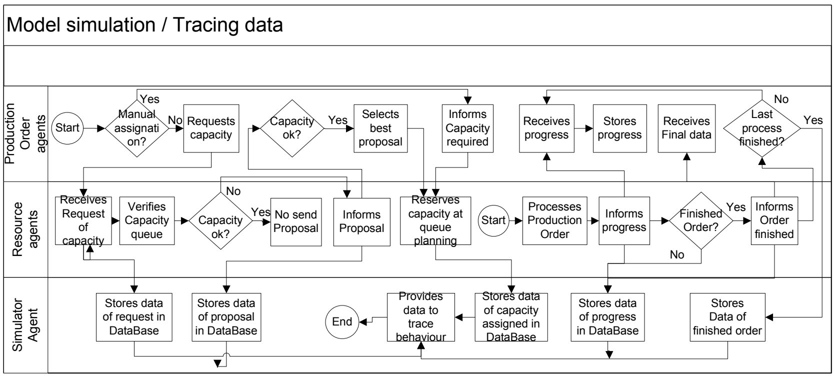

Model simulation / Tracing data
Production Order agents
Resource agents
Simulator Agent
Start
Manual assignation?
Yes
No
Requests capacity
Capacity ok?
Yes
Selects best proposal
Informs Capacity required
Receives progress
Stores progress
Receives Final data
Last process finished?
No
Yes
Receives Request of capacity
Verifies Capacity queue
Capacity ok?
No
Yes
No send Proposal
Informs Proposal
Reserves capacity at queue planning
Start
Processes Production Order
Informs progress
Finished Order?
Yes
No
Informs Order finished
Stores data of request in DataBase
Stores data of proposal in DataBase
End
Provides data to trace behaviour
Stores data of capacity assigned in DataBase
Stores data of progress in DataBase
Stores Data of finished order

Figure 17: Main Agent interactions during Tracing Simulation.

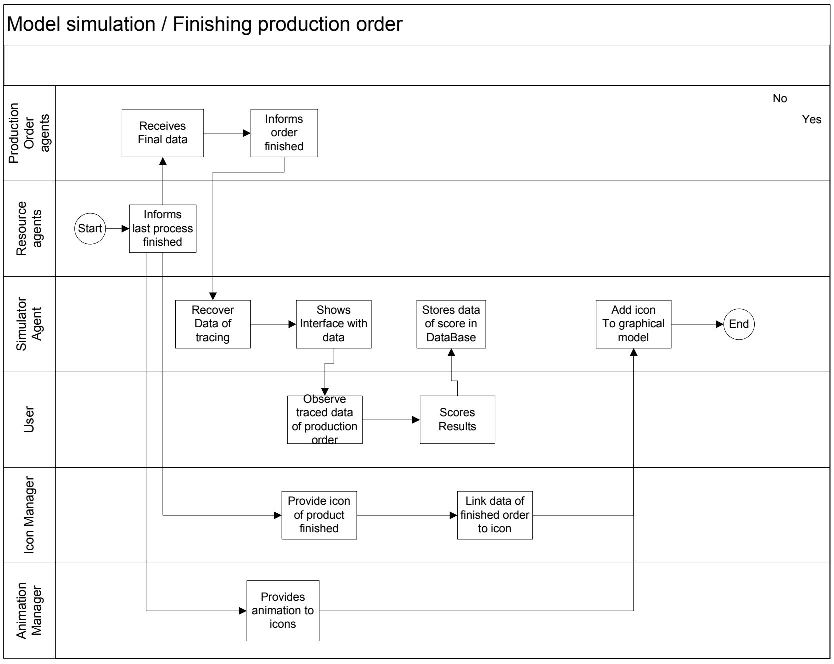

Model simulation / Finishing production order
Production Order agents
Resource agents
Simulator Agent
User
Icon Manager
Animation Manager
No
Yes
Receives Final data
Informs order finished
Start
Informs last process finished
Recover Data of tracing
Shows Interface with data
Stores data of score in DataBase
Add icon To graphical model
End
Observe traced data of production order
Scores Results
Provide icon of product finished
Link data of finished order to icon
Provides animation to icons

Figure 18: Main Agent interactions when a Production Order is finished.

with a capability grade *expert* can carry out tasks with a higher quality grade and requires less time than a resource with a capability grade *beginner* or *medium*. According to this assumption, specific values are assigned to each type of element. The final model score is obtained as follows (Equation 1): a model gets 3 units per each resource with capability grade equal to "expert", 2 units per each resource with capability grade equal to "medium", and 1 unit per each resource with capability grade equal to "beginner".

$$MG = \sum_{i=1}^{N} RG_i; \tag{1}$$

where: MG, is the Model grade; N, is the number of model resources; i, is the model resource; RG_i, is the grade of the i resource.

If the resource number of a model is 30 and all of them are "experts", the ideal model grade (IMG) is $30 * 3 = 90$. The IMG is used to obtain the optimization grade of a model (OMG) (Equation 2).

$$OMG = MG * 100/IMG; \tag{2}$$

where: OMG, is the percentage of the optimization grade of a model; MG, is the model grade; IMG, is the Ideal Model Grade.

If the OMG of a Model is 90, it indicates that it is 10% below its Ideal Model Grade IMG. When the OMG is low, the model has less chance of optimizing the resource use of the model.

5.4. Analyzing Shop Floor Configurations and Simulation Results

During Model simulation, when a User launches a production order, the User specifies the shop floor configuration to be used in that order. The configurations are created in two ways: a) configurations are defined manually by the User when a Production order is created, and b) a configuration is created by negotiations between the production order and the resources. The analysis is based on three goals: reduce processing time of a product, reduce cost of a product, and appreciate capability grade of workers and equipment (machinery and tools) used. The analysis takes into account the production of five Production Orders (A, B, C, D, E) which use the same production route (Table 7).

Route	Standard Configuration			
Operation	Resource Type	Time(Hr)	CostP/hr	Tot.Cost
10	Beginner	4	0,8	3,2
20	Beginner	1	0,8	0,8
40	Expert	2	2,4	4,8
	Total	7		8,8

Table 7: Detail of Standard Route Configuration.

The shop floor offers different resources with different capability grades (Beginners, Experts) to process each operation. According the Dreyfus Model of Skill Acquisition, the novice (beginner)has a rigid adherence to taught rules or plans and no exercise of "discretionary judgment". Thus, its work is unlikely to be satisfactory unless closely supervised. The competent (medium) is "coping with crowdedness" (multiple activities, accumulation of information), it has some perception of actions in relation to goals deliberate planning and formulates routines. Thus, its work fit for purpose, though may lack refinement. Finally, the expert transcends reliance on rules, guidelines, and maxims, it has intuitive grasp of situations based on deep, tacit understanding, has "vision of what is possible", and uses analytical approaches in new situations or in case of problems. Thus, its work is excellence achieved with relative ease.

Therefore at Table 6 if an Expert is selected, it can reduce the standard processing time and produce the product with better quality but the cost per hour is increased. On the other hand, if a Beginner is selected the standard processing time can be increased and the cost per hour will be reduced but with lower quality. Then, the Standard Route Configuration has taken into account a capability grade of instance linked to processing time (Table 7). Table 8 presents an example of how the selected instance compared with the Standard Route Configuration affects the processing time, cost and quality of the final product. In the example, the processing time is reduced but the cost is increased compared with the standard configuration. Moreover, since the use of Experts, in the example according to Table 6, the quality of final product is improved 40% in operation 10 in comparison with Standard Route Configuration. This feature allows to obtain a better final product, and reduce possible reworks and additional cost. Thus, the User can apply these criteria for the optimization of resource use.

Thus, for example, the strategy for resource allocation applied to five Production Orders will affect the optimization of the shop floor. The allocation resource also affects the down time of the available resources of the shop

Route	Selected Configuration			
Operation	Resource Type	Time(Hr)	CostP/hr	Tot.Cost
10	Expert	2	2,4	4,8
20	Beginner	1	0,8	0,8
40	Expert	2	2,4	4,8
	Total	5		10,4

Table 8: Example for Configuration of Allocation Resources.

floor and the delivery time of production orders. The analysis shows scenarios where is presented how the selected strategy for allocation of resources affects all these metrics. Figures 20, 21, 22 show the available resources that can process each route operation (Op), their capability grade, processing time assigned (PT), cost per hour (CpH), down time (DT), total used time (TT), total cost of operation (TC), and finally the delivery date of the five orders and their total processing time. Thus, it is possible to observe three global scenarios:

(a) Manual Task Assignation using the same configuration. In this assignation type, when the User selects resources, this activity requires little time to assign the tasks of a product route. However, this situation gives rise to some problems like bottle necks, resource sub-exploitation and the increase of down time on the shop floor. To solve these problems the User constantly verifies the system. This technique misuses the potential of the agent paradigm and the negotiation protocols. In Figure 20 it is possible to observe this scenario.

(b) Assignation based on negotiation techniques. In this assignation type the bottle necks are reduced, resource usage is improved and down time is reduced. However, this technique requires more time to assign tasks. The use of appropriate algorithms that includes auctions and analytical hierarchical processes helps to make decisions during task assignation in a short time. Moreover, this technique improves the task distribution among available resources. It also reduces the monitoring time of the User. In Figure 21 it is possible to observe this scenario.

(c) Combination of assignation techniques. In this combination down time reduction and resource usage are possible. The manual assignation fosters task delays and bottle necks. The negotiation techniques used for task assignation promote the work distribution. The prolonged use of manual assignation increases the risk of sub-optimization of the available capacity and requires more time for monitoring the model simulation and make decisions. In Figure 22 it is possible to observe this scenario.

During model simulation, when the User launches a production order it is possible to evaluate previous configurations linked to the product. Therefore, a User can select one of them according to its needs. A shop floor configuration can be evaluated before and after it was used for a production order. The evaluation takes into account: capability grade of an element (beginner, medium, expert), cost, processing time and score history. The delivery date of a product can be evaluated taking into account the agent number and its capability grade. The evaluation supposes that an element can reduce more processing time if it is "expert" than if it is "beginner". A configuration can be evaluated based on the instance type of the configuration elements. In an ideal configuration all resources are "experts". Thus, specific values are assigned to each type of element.

A configuration can also be evaluated at route level in order to compare the best allocation at operation level. Each operation is linked to an element. A penalty value is linked to the capability grade of an instance. A penalty value is higher if the capability grade is lesser. Thus, specific values are assigned to each type of element. The final route score and the ideal configuration grade are calculated in the same way than MG and OMG.

The Results Agent can compare the results with different configurations linked to the same product. Thus, the Simulation Agent can suggest the use of a configuration based on the results when the User launches a production order. The resources have a cost per hour. Thus, the configuration cost is obtained using Equation 3.

$$C_R = \sum_{i=1}^{N} (\sum_{j=1}^{M} TmR_{ij} * ChrR_j) \tag{3}$$

where: C_R, is the configuration route cost; N, is the number of route operations; i, is the operation route; M, is the resource number of the operation i; j, is the resource linked to the operation i; TmR_{ij}, is the time assigned to the resource j in the operation i; and $ChrR_j$, is the cost per hour of the resource j.

Also, the best route cost of a specific configuration takes into account the number of route operations, resource cost (a resource has a cost per hour), and the User needs. That is, the User can delimit the maximum and minimum cost. Thus, there are two options: a) if the User only delimits the maximum cost then the best route configuration will be the one with the lowest cost with respect the maximum cost, and b) if the User delimits the maximum and minimum cost then the best route configuration will be the one with the lowest cost with respect to the maximum cost but it must be above the minimum cost.

The expected processing time of a route configuration is obtained using Equation 4.

$$TmpPC_E = \sum_{i=1}^{N} (\sum_{j=1}^{M} TmR_{ij}) \tag{4}$$

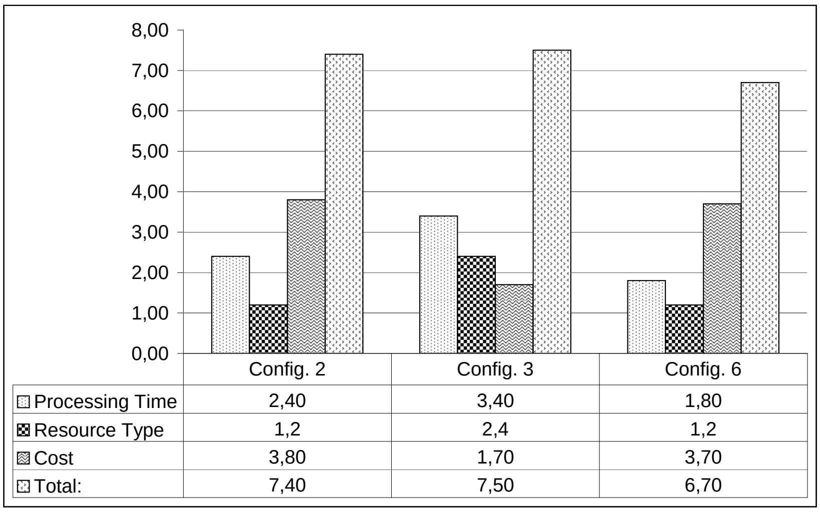
where: $TmpPC_E$, is the expected processing time of a route configuration; N, is the number of route operations; i, is the route operation; M, is the resource number of the operation i; j, is the resource of the operation i; and TmR_{ij}, is the time assigned to the resource j in the operation i.

The User criteria is a valuable feedback that is taken into account by the system due to the User being an expert in the manufacturing domain, and having specific needs. Therefore, when a production order has finished, the system asks the User to score its shop floor configuration with a value between 1 (unacceptable) and 10 (acceptable). Thus, these scores are used to evaluate a configuration. The score is obtained taking into account the number of orders where that configuration has been used and the user score assigned in those orders (Equation 5).

$$CpU_x = \sum_{i=1}^{N} CU_{ix}/N; \tag{5}$$

where: CpU_x is the average grade of the configuration x; N is the number of production orders where the configuration has been used; CU_{ix} is the user score of the configuration x assigned in the order i.

Previous variable scores allow the User to evaluate and compare a configuration with regards to a variable (time, cost, resource type or user criteria). However, the use of multicriteria techniques (Gal et al., 1999) opens up the possibility of evaluating those variables as a set. Thus, these variables are considered like *base criteria.* For this, the score of the *base criteria* is normalized. During the normalization, weights between 0,1 and 0,9 are assigned



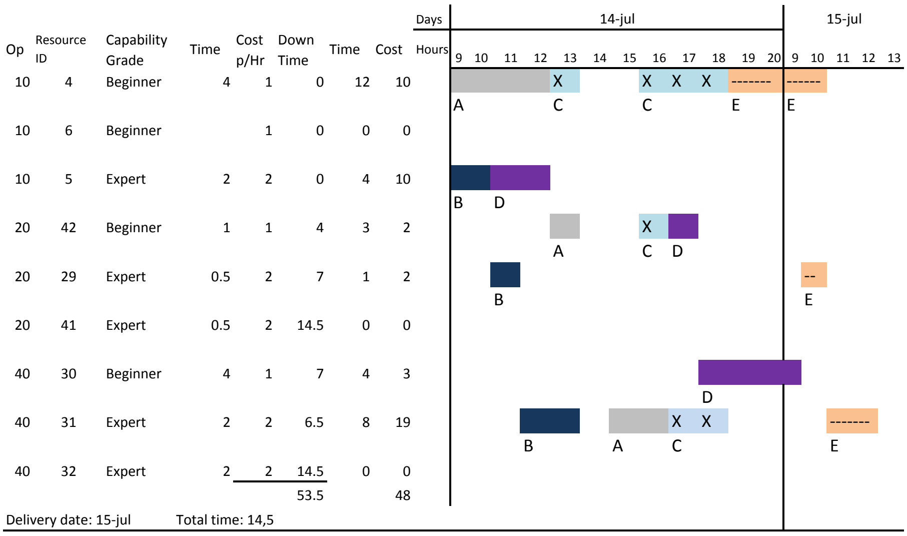
	Config. 2	Config. 3	Config. 6
▤ Processing Time	2,40	3,40	1,80
▩ Resource Type	1,2	2,4	1,2
▧ Cost	3,80	1,70	3,70
▦ Total:	7,40	7,50	6,70

Figure 19: Global evaluation of a configuration set

for each criteria.

For example, in a set that is composed of time, cost and resource type variable, if the processing time weight is $0,4$ => 40%, the cost weight is $0,4$ => 40%, and the resource type weight is $0,2$ => 20%. In this way the configuration with the lowest score will be the best configuration. In Figure 19 an example of the global evaluation of a configuration set is shown, where these weights are used. This figure also shows that Configuration 6 has the best global score although its cost is higher than the other configurations.

In this work, the analysis equations of a Flexible Manufacturing System (Stam and Kuula, 1991) have been used to identify indicators for constructing equations to evaluate and analyze results. Thus, the simulation results are used to measure the resources and model performance both historical and current. The Results Agent provides reports to show the global or individual behavior of the model resources. Some of the results are: a) production order cost (sum of cost per route operation), b) production time of an order (sum of preparation time, queue time and processing time of the route resources assigned to an order), c) total down time of the shop floor, and d) used capacity of the model resources (sum of processing time assigned to resources used in production orders). Figures 20, 21, and 22 present differences regarding the



Op	Resource ID	Capability Grade	Time	Cost p/Hr	Down Time	Time	Cost
10	4	Beginner	4	1	0	12	10
10	6	Beginner		1	0	0	0
10	5	Expert	2	2	0	4	10
20	42	Beginner	1	1	4	3	2
20	29	Expert	0.5	2	7	1	2
20	41	Expert	0.5	2	14.5	0	0
40	30	Beginner	4	1	7	4	3
40	31	Expert	2	2	6.5	8	19
40	32	Expert	2	2	14.5	0	0
					53.5		48

Delivery date: 15-jul Total time: 14,5

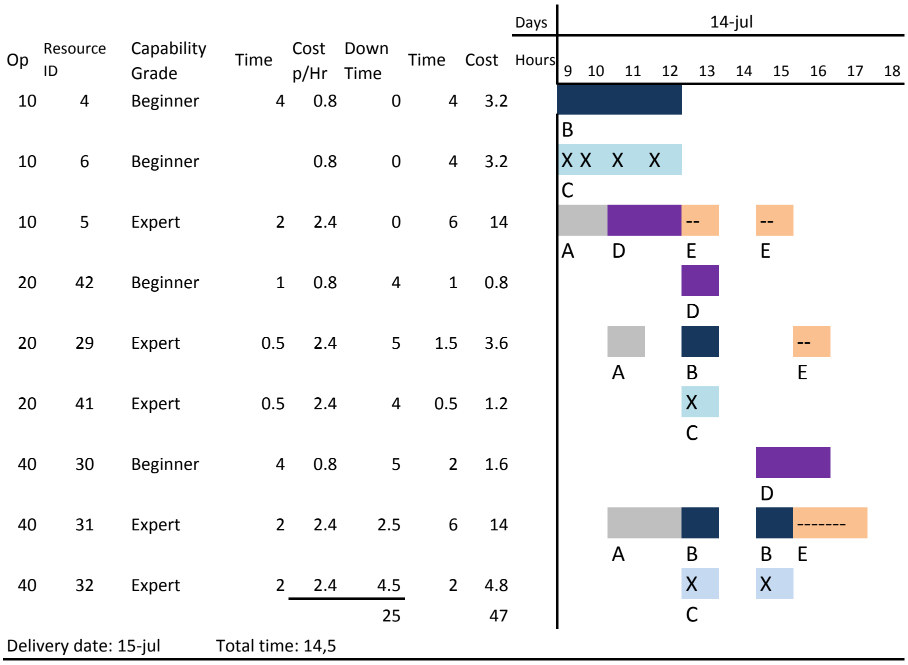
Figure 20: Use of Resources using manual task allocation.

use of shop floor resources when the allocation and negotiation techniques for the task allocation are used. Figure 20 shows results obtained after an experiment in such a planner manually assigns according to its preferences resources to process a production order. It is possible to observe that some resources have a lot of orders to be processed and others not have any one. Then, the plan capacity is sub-used, down time is raised. Figure 21 shows results obtained after an experiment in such system assigns according to task negotiation based on reduce time of processing and get products with an acceptable quality level to process a production order. It is possible to observe that tasks are distributed among all resources. Thus, the use of capacity is improved with respect to manual assignation, and down time is reduced 46.72

6. Discussion

In this section we discuss about how SimIShop, thanks to Agent-based Technology, deals with the new manufacturing requirements (Shen and Norrie, 1999b).

Enterprise Integration: Some tools are focused on a specific process in a manufacturing system such as assembly lines, and material flow (i.e. Extend (Extend, 2005), Quest (Quest and http://www.delmia.com, 2005)). A few of



Op	Resource ID	Capability Grade	Time	Cost p/Hr	Down Time	Time	Cost
10	4	Beginner	4	0.8	0	4	3.2
10	6	Beginner		0.8	0	4	3.2
10	5	Expert	2	2.4	0	6	14
20	42	Beginner	1	0.8	4	1	0.8
20	29	Expert	0.5	2.4	5	1.5	3.6
20	41	Expert	0.5	2.4	4	0.5	1.2
40	30	Beginner	4	0.8	5	2	1.6
40	31	Expert	2	2.4	2.5	6	14
40	32	Expert	2	2.4	4.5	2	4.8
					25		47

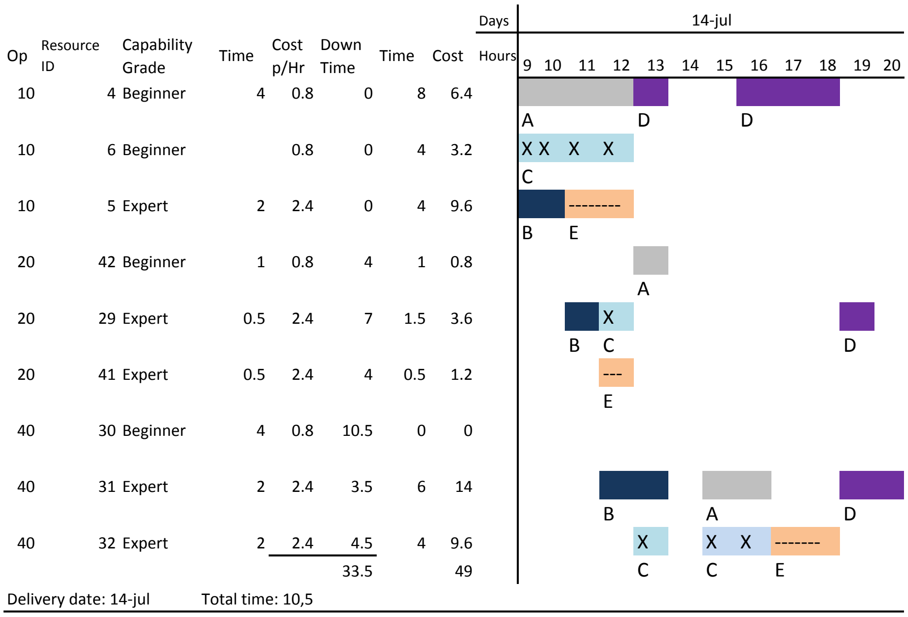
Delivery date: 15-jul Total time: 14,5

Figure 21: Use of Resources using task negotiation.

present-day tools involve the whole life cycle of a product (i.e. modelling of processes, flexible routes, raw materials, human resources, tools, machines; definition of maintenance calendar, normed interactions, animation; simulation of: break-downs and addition of resources "on the fly", during simulation). MAS theory allows us to simulate the global behavior of each entity, mixing complex and simple behavior at different levels, including physical elements (humans, machines, building spaces) and global policies involved in the whole life cycle of a product.

Distributed Organization: Current simulation tools, are generally based on pre-established models. Simulation tools incorporate complex algorithms to optimize processes, but the knowledge base is still centralized on a few points in the organization structure of the system (i.e. Promodel (Promodel and http://www.promodel.com, 2005), Arena (Arena and http://www.arenasimulation.com, 2005)). MAS technology allows distribution and limits the agents's knowledge scope according to the states of the system.

Heterogeneous Environments: Generally, each tool is developed by some specific software, which could be proprietary or commercial such as C (i.e.



Op	Resource ID	Capability Grade	Time	Cost p/Hr	Down Time	Time	Cost
10	4	Beginner	4	0.8	0	8	6.4
10	6	Beginner		0.8	0	4	3.2
10	5	Expert	2	2.4	0	4	9.6
20	42	Beginner	1	0.8	4	1	0.8
20	29	Expert	0.5	2.4	7	1.5	3.6
20	41	Expert	0.5	2.4	4	0.5	1.2
40	30	Beginner	4	0.8	10.5	0	0
40	31	Expert	2	2.4	3.5	6	14
40	32	Expert	2	2.4	4.5	4	9.6
					33.5		49

Delivery date: 14-jul Total time: 10,5

Figure 22: Use of Resources using a mix of manual task allocation and negotiation.

Factor Aim (C) (FactorAim and http://www.wintek.com, 2005), Extend (ModL)). Due to their own limitations they only offer specific integration interfaces(i.e Grasp2000 only establish communication with robots using native robot languages at shop floor). MAS technology allows us to simulate any environment regardless of any physical restrictions or proprietary software in each piece of equipment. These feature is associated to a internal flexibility that offer a software to communicate which other that has been developed in a different programming language.

Interoperability: Most of the tools evaluated offer interfaces between themselves and other software by files formatted by CAD tools, text files, DLLS, ODBC interfaces,etc. (i.e. Arena, Extend). MAS technology allows communication with any kind of software without limitations.

Open and Dynamic Structure: Some of the current tools offer very little in the way of "on the fly" modification methods without disrupting operational functioning of the system. Some of them apply programming layers into hierarchical structures, but the introduction or elimination of any element is not an easy task for the user (i.e. MicroSaint,Witness. MAS agents can be introduced or eliminated automatically without disrupting current system operation according to the needs of the current system state (i.e. test of "what-if" events such as break downs, add machines that do not exist). This feature promotes the scalability of the system.

Integration of humans with software and hardware: The evaluated tools allow the incorporation of a human element, but with limited attributes (i.e. Arena, Extend, FactorAim). MAS theory allows for the incorporation of the complex behavior features of human interaction with software and hardware involved in a system. This proposal allows to define expertise level, knowledge areas of people, define relations among human and machines/tools, and add basic behavior of monitoring task by adding a staff entity as part of simulation.

Agility: Current tools are based on models, the complexity of each model and algorithms can vary, but events are established in advance, including alternative action plans (i.e. Promodel, Arena, Quest). Due to distributed knowledge, features such as autonomy and productiveness offered by MAS technology, it is possible to detect and adjust the system automatically. It allows the creation and execution control of its own plans and/or strategies when an unexpected event occurs during simulation. This system behavior can be implemented by an emergent agent group which perform complex processes by mixing their individual knowledge. Thus, this is the key feature that is considered as the biggest strength of MAS paradigm.

Fault Tolerant: The tools are based on models, some of them offer an additional optimization based on complex algorithms (i.e. Quest). When faults are present, it is necessary to correct the model either stopping the system or "on-the-fly" running. MAS technology, due to autonomy and proactive features of agents, responds and executes appropriate, alternative plans automatically without interrupting/stopping the global operation of the system.

The requirements of the "new manufacturing era" (Shen and Norrie, 1999a) together with a previous evaluation of tools for modelling a manufacturing system (Eldabi and Paul, 2001) associated to the advantages of using the agent paradigm in this proposal were evaluated. Table 9 shows an overview of the evaluation of the different requirements in the tool.

The features included in the proposal solve some problems related to requirements of the new manufacturing era. Mobility and communication support control distribution, they allow models with proactive, reactive, autonomous and self-control elements to be defined in a hierarchical architecture. These features facilitate the definition of a distributed organization.

Requirements of New Manufacturing Era	**Characteristics of Architecture and Metamodel proposed for Agent-supported Simulation**								
	Mobility	Intelligence	Autonomy	Control Distribution	Reconfigurability	Scalability	Flexibility	Proactivity Reactivity	Fault Tolerance
Enterprise Integration		X	X	X		X	X	X	X
Distributed Organization	X		X	X	X	X	X		X
Heterogeneous Environment		X	X	X			X	X	X
Inter-operability		X		X	X		X	X	
Cooperation	X		X	X	X		X	X	X
Human Integration with SW and HW				X	X				X
Agility	X	X	X	X		X	X	X	
Scalability	X	X		X	X	X	X	X	X
Fault Tolerance		X	X	X	X		X	X	X

Table 9: Evaluation regarding the Agent Paradigm and Requirements of New Manufacturing Era.

Regarding the System Reconfigurability, the definition of flexible, intelligent, proactive and fault tolerant elements allow the modeling of flexible manufacturing systems adaptable to the designer requirements. The detail of the attributes level of the model elements facilitates their reconfiguration (i.e. capacity per hour, process type, work center, control algorithms and manufacturing routes of a product). The Model Simulation allows the simulation of production orders for different products at the same time. The goal of the systems is the optimization of resource use by balancing work among all resources and taking into account their capabilities, time restrictions, and quality. This fact allows the complex behavior of a real Manufacturing System to be represented. Moreover, the proposal facilitates the integration of humans with software and hardware in a manufacturing system by using easy-to-use interfaces.

The architecture and the metamodel allow elements to be added or eliminated from the manufacturing system in order to satisfy emergent needs in any area of the shop floor during modeling and simulation. Thus, this feature supports the system scalability. The intelligence, communication and control distribution of the manufacturing elements support the fault tolerance. Agents use complex algorithms to make decisions and cooperate to recover and maintain the system operation.

Figure 23 presents a brief review of the proposal features regarding model creation, simulation and analysis, and animation versus representative commercial and institutional simulation tools.

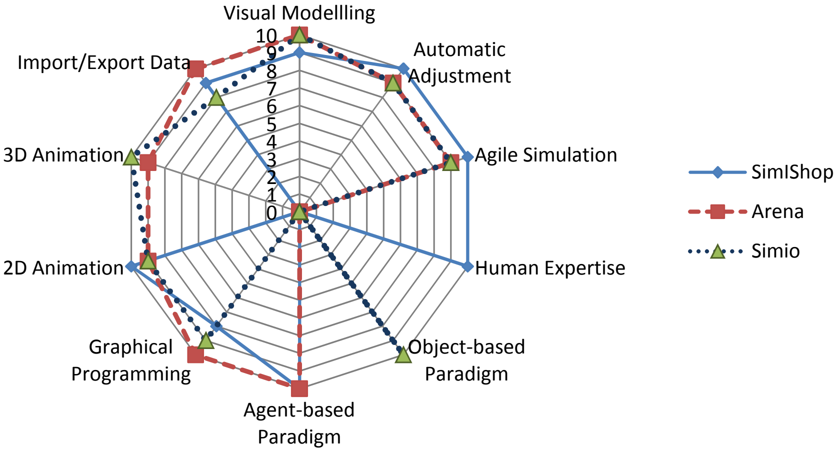


Figure 23: Comparing SimIshop versus representative Commercial and Institutional Tools.

For the Model Creation, the prototype provides a high flexibility grade which allows the User to design and define complex behavior with a sufficient level of detail. The interface guides the User into the manufacturing domain, obtaining the benefits of the Multiagent paradigm. The User can use this proposal to solve planning and control problems.

The proposed architecture allows the definition of manufacturing routes linked to products. In this way, it is possible to simulate any combination of manufacturing routes for the same product or different products in a single simulation. Thus, it is possible to reflect complex behaviors that represent situations that emerge in a real manufacturing system environment.

During simulation, the maximum number of elements in a model is limited by the hardware capacity and JADE platform where it is executed. In comparison with current simulation tools, the agent paradigm allows instances to be added or eliminated during model simulation. This feature automatically adds capacity without disturbing the global behavior of the manufacturing system. Moreover, it provides more flexibility to do complex experiments (i.e. add new resources when a bottle neck is detected or remove a resource to observe how the system reacts regarding an unexpected fault). Also, the prototype allows the User to define the animation type individually and change it before and during model simulation. It is possible to keep the general progress state of production orders and all the model elements during simulation.

7. Conclusion

In this paper a tool has been presented, which allows the validation of the agent-supported simulation architecture for the creation and simulation of agent-supported models presented in previous works. Both agent-supported architecture for simulation and an intelligent manufacturing metamodel have been proposed to solve the requirements of the new manufacturing era. The tool provides enough flexibility for designing complex models and experiments.

SimIShop includes the main tasks of a simulation tool like: model creation, model simulation, animation and distribution of elements. Moreover, the basis for evaluating models and shop floor configurations have been presented. Finally, the contributions of the proposal regarding the requirements of new manufacturing were presented. In addition, we have shown the advantages of using the agent paradigm to simulate the complex behaviors of an Intelligent Manufacturing System. One of the main contributions of this work, is to provide a tool that allows the User to take advantage of the agent paradigm by means of easy-to-use interfaces.

Since the agent-supported simulation of manufacturing systems is a wide-ranging field, some future work includes: creating a recommendation system of configurations, simulating warehouse behavior, adding a new module to link the database of the simulation tool to the database of an ERP, among others.

References

Arena, http://www.arenasimulation.com, 2005.

Ashley, K., 2006. Case-Based Reasoning. volume 21 of *Information Technology and Laywers.* Springer.

Automod, 2014. http://www.automod.com, accessed:2014.

Banks, J., 1998. Handbook of Simulation:Principles, Methodology, Advances, Applications and Practice. New York: John Wiley and Sons. chapter Principles of Simulation. pp. 3–30.

Banks, J., Carson, J., Nelson, B., Nicol, D., 2001. Discrete-Event System Simulation. Prentice-Hall,Englewood Cliffs,N.J.

Botti, V., Giret., A., 2008. ANEMONA. A Multi-Agent Methodology for Holonic Manufacturing Systems. volume XVI of *Springer Series in Advanced Manufacturing*. Springer. 214 p.

Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P., 1998. Reference architecture for holonic manufacturing systems: Prosa. Computers in Industry 37, 255–274.

BV, T., http://www.tryllian.com/, 2005.

Carrascosa, C., 2004. Meta-Razonamiento en Agents con Restricciones Temporales crticas. Ph.D. thesis. Universidad Politécnica de Valencia. Valencia, España.

Choy, K., Lee, W., 2000. Task allocation using case-based reasoning for distributed manufacturing systems. Logistics Informtion Management 13, 167–176.

Conservation, I., 2003. Professional standards in conservation, http://www.icon.org.uk/.

Dale, J. and Knottenbelt J. and http://sf.us.agentcities.net/aap/index.html, 2005.

DE3, 2014. http://www.bygsimulation.com, accessed:2014.

Dreyfus, S.E., Dreyfus, H.L., 1980. A Five-Stage Model of the Mental Activities Involved in Directed Skill Acquisition. McGrawHill, Washington, DC.

Dreyfus, S.E., Dreyfus, H.L., 1986. Mind over Machine: the power of human intuition and expertise in the age of the computer. Blackwell, Oxford, Basil.

Duffie, N., Chitturi, R., Mou, J., 1998. Fault-tolerant heterarchical control of heterogeneous manufacturing system entities. Journal of Manufacturing Systems 7, 315–328.

Dynamics, E., 2014. http://www.incontrol.nl, accessed:2014.

Eldabi, T., Paul, R., 2001. Evaluation of tools for modeling manufacturing system design with multiple levels of detail. Int. Journal of Flexible Manufacturing Systems 13, 163–176.

Esteva, M., Rosell, B., Rodríguez-Aguilar, J.A., Arcos, J.L., 2004. Ameli: An agent-based middleware for electronic institutions, in: Jennings, N.e.a. (Ed.), AAMAS 2004. Third international joint conference on autonomous agents and multiagent systems, ACM. ACM. pp. 236–243.

Extend, a., 2005.

FactorAim, http://www.wintek.com, 2005.

FlexSim, 2005. http://www.flexsim.com, accessed:2014.

Fujitsu, Sun, IBM, HP, Spawar, InterX, http://www.java-agent.org/Documents/OSLFLAvo.4.htm, 2005.

Gal, T., Stewart, T., Hanne, T., 1999. Multicriteria Decision Making Advances in MCDM Models, Algorithms, Theory and Applications. volume 21 of *International Series in Operations Research & Management Science*. Springer.

Grasp2000, 2014. http://www.bygsimulation.com, accessed:2014.

Grasshopper, http://www.grasshopper.de/, 2005.

Heim, J.A., 1997. Integrating distributed simulation objects, in: Proc. of the 1997 Winter Simulation Conference, pp. 532–538.

Information-Technology Promotion Agency, J., Communication Technologies, h., 2005a.

Information-Technology Promotion Agency, J., Communication Technologies, h., 2005b.

JADE, 2005. Java agent development framework, http://jade.tilab.com.

Labs, B., http://193.113.209.147/projects/agents/zeus/, 2005.

Larsen, M., Bilberg, A., Kirkby, P., 1997. Cals in denmark- danish sme's exhange of information throughout the product lifecyle, in: Proceedings of the 8th International Conference and Exhibition on CALS and Electronic Commerce in Europe, CALS Europe 97, Frankfurt/M., Germany. pp. 196–208.

Larsen, M., Kirkby, L., Verterager, J., 1999. Alignment of product models and product state models - integration of the product lifecycle phases, in: The 11th CALS Expo International & 21st Century Commerce 1998 Conference Long Beach, USA,October, Proceedings of the 7th Mediterranean Conference on Control and Automation (MED99, Haifa, Israel. pp. 1990–2006.

Larsen, M., Sorensen, C., Langer, G., 2001. Development of a production meta product state model. Computers in Industry 46, 275–287.

LEAP, http://leap.crm-paris.com/, 2005.

Magnusson, J., Holm, T., 1996. The product model - cals technical goal,(the defence acquisition and maintenance unit). Forsvarets Materielverk .

Marsa-Maestre, I., Ito, T., Klein, M., Fujita, K., 2009. Balancing utility and deal probability for auction-based negotiations in highly nonlinear utility spaces, in: IJCAI 2009. The 21 International Joint Conference on Artificial Intelligence, pp. 214–219.

Maturana, F., Norrie, D., 1996. Multi-agent mediator architecture for distributed manufacturing. Journal of Intelligent Manufacturing 7, 257–270.

MCLean, C., Leong, S., 2001. The expanding role of simulation in future manufacturing, in: WSC 2001. The Winter Simulation Conference, ACM. ACM. pp. 1478–1486.

MicroSaint, http://www.maad.com, 2005.

Mordeson, J., Wierman, M., Clark, T., Pham, A., Redmond, M., 2013. Linear Models in the Mathematics of Uncertainty. Studies in Computational Intelligence. Springer. volume 463. chapter The Analytic Hierarchy Process. pp. 93–117.

Nikoukaran, J., Paul, R.J., 1999. Software selection for simulation in manufacturing: a review. Simulation Practice and Theory 7, 1–14.

Peng, Y., Finin, T., Chu, B., Tolone, W., Boughannam, A., 1998. A multiagent system for enterprise integration, in: Proc. of 3rd Int. Conf. on the Practical Application of Intelligent Agents and Multi-Agents, London, UK. pp. 155–169.

Promodel, http://www.promodel.com, 2005.

Quest, http://www.delmia.com, 2005.

Rabelo, R., Camarinha-Matos, L., 1994. Multi-agent based dynamic scheduling. Int. Journal on Robotics and Computer Integrated Manufacturing II, 303–310.

Ruiz, N., Giret, A., Botti, V., 2006. Towards an agent-based simulation tool for manufacturing systems, in: Proc. of 11th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA06), IEEE Computer Society, Praga, Czech Republic. pp. 797–804.

Saaty, L., 1988. Mathematical Models for Decision Support. NATO ASI Series. NATO. volume 48. chapter What is the Analytic Hierarchy Process? pp. 109–121.

Shen, W., Norrie, D., 1999a. Agent-based systems for intelligent manufacturing: A state of art survey. Knowledge and Information Systems 1, 129–156.

Shen, W., Norrie, D., 1999b. Agent-based systems for intelligent manufacturing: A state-of-art survey. Knowledge and Information Systems , 129–156.

Stam, A., Kuula, M., 1991. Selecting a flexible manufacturing system using multiple criteria analysis. International Journal of Production Research 29, 803–820.

Tonshoff, H., et. al., 2000. A mediator-based approach for decentralised production planning, scheduling and monitoring, in: Proc. of 2nd CIRP Int. Seminar on Intelligent Computation in Manufacturing Engineering, Capri, Italy. pp. 113–118.

Valckenaers, et. al., 1999. Towards a novel manufacturing control principle, in: Proc. of the 2nd Int. Workshop on Intelligent Manufacturing Systems, Leuven, Belgium. pp. 871–875.

Veeramani, D., Wang, K.J., 1997. Performance analysis of auction-based distributed shop-floor schemes form the perspective of the communication system. The International Journal of Flexible Manufacturing Systems 9, 121–143.

Wagner, J., Hollenbeck, J., 2004. Comportamiento Organizativo. Thomson.

Witness, http://www.lanner.com, 2005.

Yu, B., Popplewell, K., 1994. Metamodels in manufacturing: a review. International Journal of Production Research 32, 787–796.