Agent-Based Generative Simulation of an Intelligent Distributed Scheduling World with Netlogo

Milagros M. Rolón
CONICET, Instituto INGAR
Santa Fe, Argentina, S3002 GJC
rolon.milagros@gmail.com

and

Mercedes M. Canavesio
Universidad Tecnológica Nacional, Fac. Reg. Santa Fe
Santa Fe, Argentina, S3004EWB
mcanaves@frsf.utn.edu.ar

and

Ernesto C. Martínez
CONICET, Instituto INGAR
Santa Fe, Argentina, S3002 GJC
ecmarti@santafe-conicet.gob.ar

Abstract

Unplanned disruptive events and disturbances such as arrivals of rush orders or machine breakdowns must be managed locally to avoid propagating the effects along the value chain. To overcome the traditional separation between task scheduling and manufacturing execution systems the novel idea of emergent synthesis/control of schedules for better handling the dynamics at the shop-floor is proposed. A new interaction mechanism for simultaneous distributed scheduling and execution control is evaluated using a generative simulation model in Netlogo. The interaction mechanism has been designed around the concept of order and resource agents acting as autonomic managers within the artificial society of a dynamic Gantt world. The advantages of generative modelling in agent-based simulation are discussed to emphasize how difficult to predict emerging behaviours and bottom-up macroscopic dynamics in a manufacturing case study can be addressed by proper design of agent interactions. Results obtained for different abnormal scenarios are presented to highlight the benefits of simulating artificial societies of intelligent agents.

Keywords: Intelligent distributed scheduling; Agent-based modeling; Interaction mechanisms, Autonomic systems; Generative simulation

Resumen

La gestión local de acontecimientos disruptivos y perturbaciones imprevistas como la llegada de pedidos urgentes o interrupciones en el funcionamiento de los recursos evitan propagar sus efectos a lo largo de la cadena de valor. Para un mejor manejo de la dinámica en el piso de planta, se propone la idea de una síntesis/control emergente del scheduling en contraposición a la separación tradicional entre las tareas de programación y ejecución. A través de un modelo de simulación generativo se evalúa un novedoso mecanismo de interacción que realiza simultáneamente el scheduling distribuido y el control de ejecución, diseñado en base al concepto de agente orden y agente recurso que actúan como gestores autonómicos dentro de la sociedad artificial de un mundo representado por un Gantt dinámico. Se discuten las ventajas del modelado generativo en la simulación basada en agentes en un entorno de manufactura con el objetivo de acentuar la dificultad de predecir comportamientos emergentes y la dinámica macroscópica que surge de la interacción de los agentes, y se presentan los resultados obtenidos para los distintos escenarios para destacar los beneficios de la simulación de sociedades artificiales de agentes inteligentes.

Palabras claves: Scheduling inteligente distribuido, Modelado basado en agentes, Mecanismos de interacción, Sistemas autonómicos, Simulación generativa.

INTRODUCTION

Delays and failures are ubiquitous events at the shop floor which has recently motivated the development of decentralized task (re)scheduling and execution control systems to increase responsiveness and agility in complex production environments. One such manufacturing control architectures is the holonic MES implemented by a multi-agent system in the PROSA architecture [1]. Cooperation between the holonic MES and the optimization performed by the planning system was also studied by Verstraete et al [2]. Along similar ideas, the ADACOR architecture [3] alternates between stationary states, where system control relies on supervisors and coordinator levels, and transient situations, triggered by the occurrence of disturbances where the MES switches its decision-making policy to a heterarchical structure.

Rolón et al. [4] proposed an autonomic MES concept which emphasizes a total integration of decentralized schedule generation and execution control for agility and responsiveness. Such integration is carried out through the concept of order and resource agents acting as autonomic managers [5]. In this paper, generative modeling [6] will be used as a computational tool to analyze emergent properties of this new MES design. This novel approach, based on agents and simulation, will allow an objective assessment of dynamic behaviors of a distributed scheduling system modeled as an artificial society of autonomic agents.

INTELLIGENT DISTRIBUTED SCHEDULING

Manufacturing Execution System or MES [7] is a method that has developed from what tend to be the homogenized and compacted version of the classic disciplines, such as production data acquisition, staff work time logging, quality assurance and finite scheduling. The aim of an MES is to make the value-adding processes transparent and on the basis of this transparency to create not only horizontal but also vertical control cycles. The most common approach of MES implementation is to heavily resort to a given schedule, carried out by a single agent that knows the environment and makes the planning and distribution for each resource so as to focus only on handling details and contingencies in task execution. However, the lack of adaptive behavior and learning capabilities seriously limit the effectiveness of conventional control techniques in MES systems. For efficacy, holonic control architectures were proposed based on the concept of cooperative and autonomous agents associating an agent to every holon in the system ([1], [3], [4]).

According to the Holonic Manufacturing System (HMS) consortium, a holon is an autonomous and cooperative building block of a manufacturing system for transforming, transporting, storing and/or validating information and physical objects. In a holonic MES, in contrast to many decentralized designs, each component is autonomous and possesses local knowledge that is sufficient to accomplish its own task. The task which a single component is unable to finish by itself may require the cooperation of a cluster of holons or agents. The holonic MES designed in this paper is the result of ongoing interactions among decentralizing decision-making agents. These agents are conceived as autonomic manager units and when integrated to a physical or abstract object (managed element) they conform the holons [8]. As shown in Fig. 1, these managed elements interact with other elements via their autonomic managers. Each agent playing its role implements the monitor-analyze-plan-execute (MAPE) loop which comprise both scheduling and control for a given order or resource.

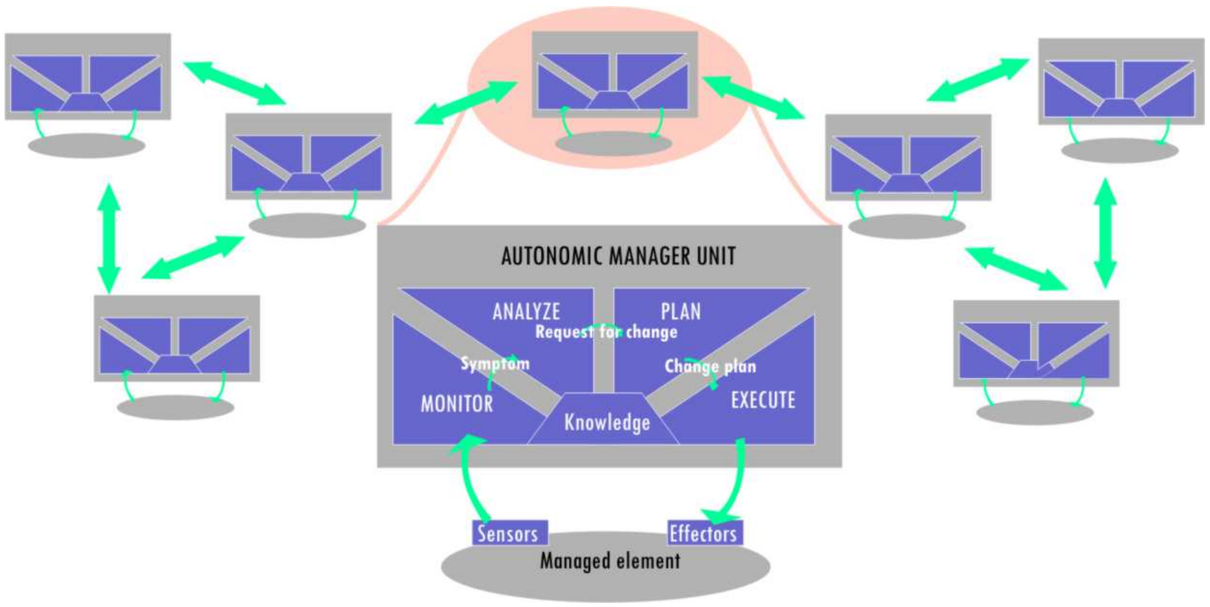

Figure 1. The interaction of the managed elements via their autonomic managers

For the autonomic agent to be self-managing regarding its managed object, it must have an automated method to collect the details it needs from the manufacturing system (monitor function); to analyze those details to determine if something needs to be changed (analyze function); to create a plan, or sequence of actions that specifies the required changes (plan function); and to perform those actions (execute function).

There are two different roles that can be assigned to an agent in the holonic MES, depending of the object being managed by each of them: *orders* and *resources*. The order agent (OA) is responsible for complete the order as required which determines the operations (tasks) for the processing of that order. The order agent chooses the processing route to be communicated as the current order process plan (selecting one best-performing solution) and follows it moving the order between the resources. The resource agent (RA) manages the schedule for a resource and registers its usage and failure state. It is the responsible for the execution of tasks for different orders. The interaction goal hierarchy for the holonic MES is shown in Fig. 2 in which the undirected lines denote sub-goal relationships and the directed lines depict temporal dependency.

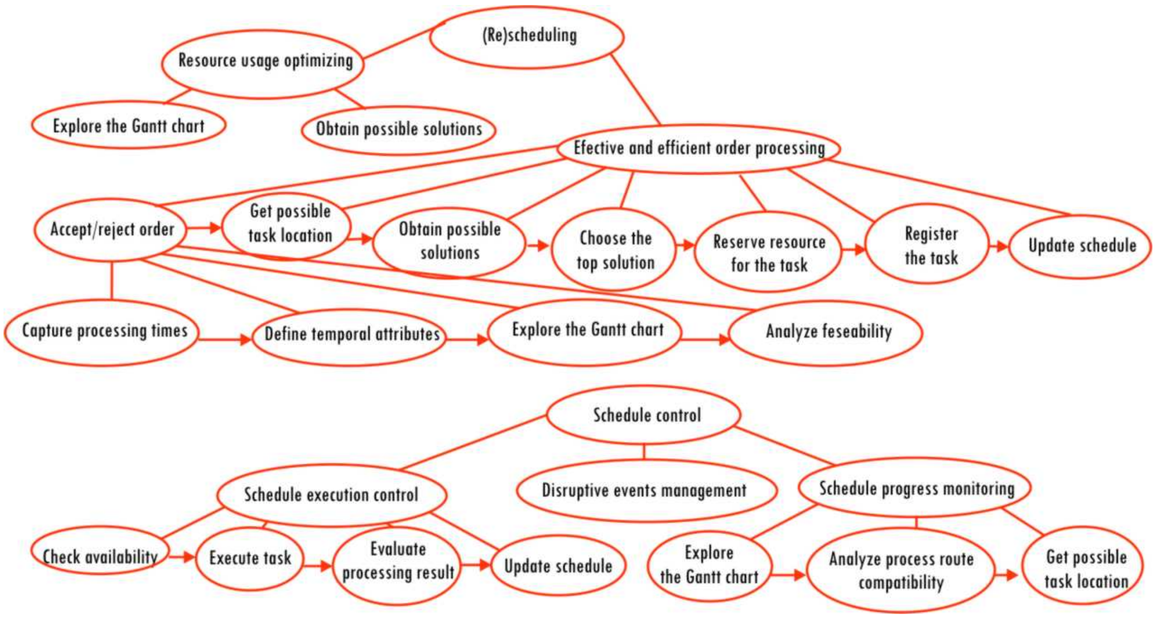
(Re)scheduling
Resource usage optimizing
Explore the Gantt chart
Obtain possible solutions
Efective and efficient order processing
Accept/reject order
Get possible task location
Obtain possible solutions
Choose the top solution
Reserve resource for the task
Register the task
Update schedule
Capture processing times
Define temporal attributes
Explore the Gantt chart
Analyze feseability
Schedule control
Schedule execution control
Disruptive events management
Schedule progress monitoring
Check availability
Execute task
Evaluate processing result
Update schedule
Explore the Gantt chart
Analyze process route compatibility
Get possible task location

Figure 2. Goal Hierarchy in the holonic MES

The holonic MES architecture implements the MAPE loop for both order agents and resource agents of the mechanism for distributed scheduling through indirect interactions using a dynamic Gantt chart while they carry out the automated functions (see Fig. 3) as follows:

- The monitor function includes the schedule monitor functionality, where both of them oversee the Gantt chart looking for schedule updates (the order agent for the current order process route and the resource agent specifically for its resource), the order acceptance functionality, where the order agent finds in the Gantt if a candidate order is feasible and so can be added to the current schedule or otherwise is rejected, and the availability checking functionality, where the resource agent checks at each step of order processing the chosen resource availability.
- The analyze function comprises the process route selection and resource earmarking functionalities, where the resource agents returns the data to constitute the list of alternative solutions from which the top solution will be chosen.
- The plan function includes the resource booking and task registration functionalities, where the order agent asks for slot reservation to each resource agent of the top solution, and the schedule updating functionality, where the resource agents reflect resource commitment in the common Gantt chart.
- The execute function for the resource agent manages the completion of its resource usage plan with due consideration for dynamic updates using the task execution functionality, whereas the execution control functionality of the order agent deals with the local solution of disruptive events by order rescheduling.

When a new order arrives, the corresponding order agent looks up the Gantt in order to discover if the new order is feasible. If so, the order is accepted, and if not, it is rejected to the client with a feasible due date. If it is accepted, the order agent asks candidate resource agents, stage by stage, different options for the probable arriving time at their queue and makes them a request about the probable finalization time at their resource. These options given by resource agents allow making a list of solutions. Given the different solutions provided by resource agents, the order agent selects the top one, with the aim of booking the time slot corresponding to the different tasks in the order. In the current intention selection, feasible orders can be placed according to different resource agent decision rules such as SPT: shortest processing time, EDD: earliest due date and FIFO: first in first out. As a response to confirmation of resource usage by an order agent, the resource agent registers the task and updates the Gantt chart accordingly.

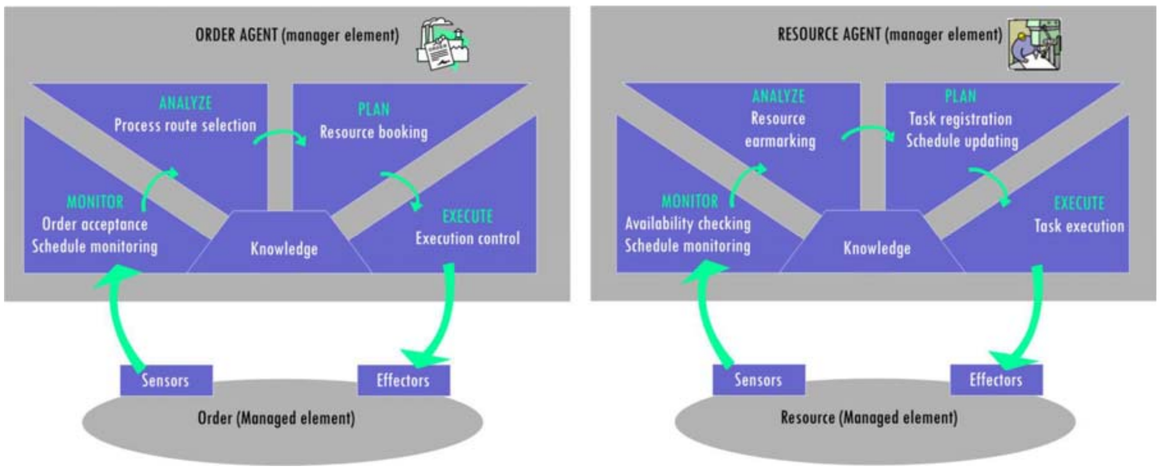

Figure 3. Order agent and resource agent viewed as autonomic managers

GENERATIVE SIMULATION OF THE MANUFACTURING EXECUTION SYSTEM PROPOSED

3.1 Agent-based Modeling and Simulation

Agent-based modeling and simulation (ABMS) is a computational method that enables an analyst to create, analyze and experiment with simulation models composed of agents that interact within a well-defined environment [9]. One of the advantages of computational modeling is that it forces the modeler to be precise: unlike theories and models expressed in natural language, a computer simulation program has to be completely and exactly specified if it is to run on a computer. Another advantage is that *in silico* experiments with a computational model is cheaper or occasionally is the only way to test hypothesis regarding mechanism design. Furthermore, an experiment can be set up and repeated many times, using a range of parameters or allowing some factors to vary randomly.

ABMS is able to discover emergent connections between system components, tie experience with detailed processes to system-level knowledge and archetype patterns, and in this way it may identify possible outcomes that are outside the range of analytical thinking [6]. To this aim, agent modeling must follow an iterative model construction process. This process starts with an initial description of the behavior of individual agent behaviors as well as interaction mechanisms along with supporting data. This conceptual description is then converted to a generative model that can be used to test hypothesis. The resulting model is then simulated and the initial results are examined. The internal behavior definitions and the interaction mechanism design in the model are then updated based on system goals pursued, and the model is simulated again. This progressive refinement process continues until the model reproduces both the behaviors and results of the target system, as can be seen at Fig. 4.

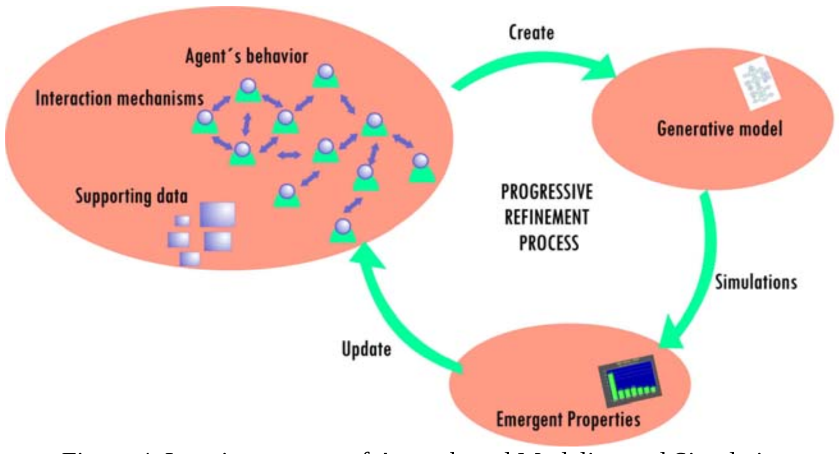

Figure 4. Iterative process of Agent-based Modeling and Simulation

Generative models using agent-based objects are a very natural way for understanding and designing complex adaptive systems. Regardless of the presence of equilibrium, the path from microworlds to emergent properties (macrobehavior) requires systemic synthesis rather analytical tools ([10], [11]).

3.2 Generative Modeling

Generative modeling [12] refers to the application of computational models to understand complex social systems. The usual barriers set by the standard modeling tools, such as the need to keep the model within a reasonable size are overcome with the computational modeling approach. In computational models a controlled micro world simulation is used to analyze the macro level behaviors resulting from on-going interactions on micro scale worlds. The model serves as a tool to validate the hypothesis (simulated experimentation) and as a result to improve the proposed designs [11]. This tool reveals emergent connections between the component systems and makes viable the identification of unexpected consequences of agent interactions. Therefore it allows discovering and analyzing the social structures and the properties that emerge from a given mechanisms design.

Contrasting with the traditional top-down approach (see Fig. 5 for details), a bottom-up emergence in an artificial society of agents which mainly communicate through the actions taken, define patterns that arise from the chosen mechanism design (e.g., sugarscape world, [12]). Moreover, those aggregate patterns should be immune to reasonable variations in the individual agent behavior. Although the agents are simple, the result of its ongoing interaction can be very complex and difficult to predict.

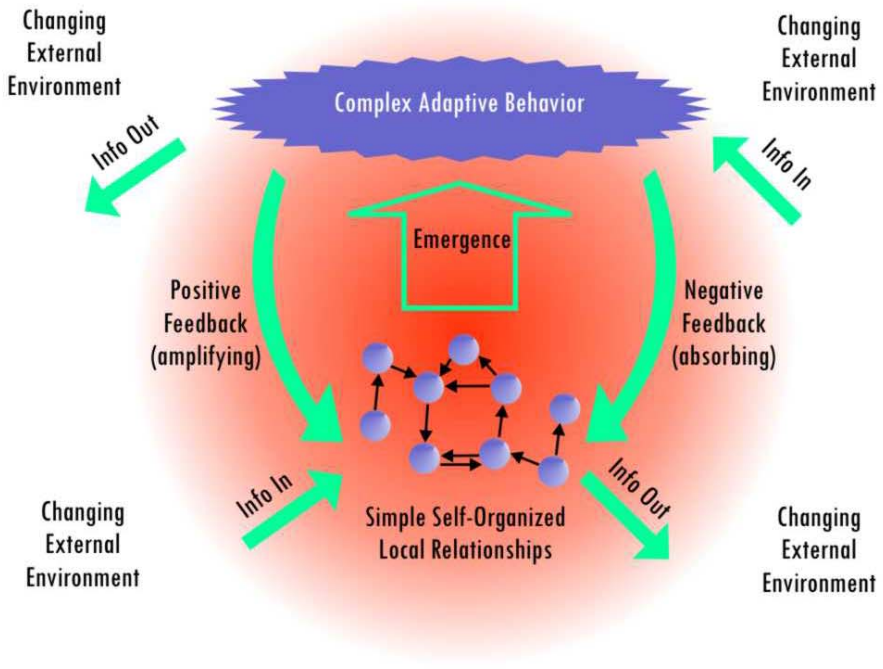
Changing External Environment
Complex Adaptive Behavior
Changing External Environment
Info Out
Info In
Emergence
Positive Feedback (amplifying)
Negative Feedback (absorbing)
Changing External Environment
Info In
Simple Self-Organized Local Relationships
Info Out
Changing External Environment

Figure 5. Bottom up emergence of complex adaptive behavior

When interactions are not independent, feedback regulation can play a substantial role. Feedback fundamentally alters the dynamics of a system. In a system with negative feedback, changes get quickly absorbed and the system gains stability. With positive feedback, changes are amplified leading to instability. Thinking about positive and negative feedback loops provides a rather shallow view into the micromotives for complex archetype behaviors.

Most of the existing analytic tools require that the underlying agents have a high degree of homogeneity. This homogeneity is not a feature often observed in the world but rather a necessity imposed by analytical modeling approach [10]. Unlike traditional tools, computational methods are able to incorporate heterogeneous agents easily. In this way, software agents managing the manufacturing settings (resources and products) can interact with each other and coordinate and plan their activities so that they can further their own design objectives. Essential in this process are the abilities to negotiate and reach deals with other agents (resources, product managers, competitors or providers). Thus the emergent behavior from interactions among heterogeneous agents with different objectives highlights the importance of resorting to the generative modeling approach of ABMS which allows designing meta-rules that change the interaction rules in distributed decision-making.

3.2 The Dynamic Gantt World

In the proposed mechanism design, there is one resource agent for each equipment item whereas each order agent only deals with a specific order type. The communication is achieved through direct contact among concerned agents and indirect interactions through the dynamic Gantt chart which is used as a blackboard as it is shown in Fig. 6.

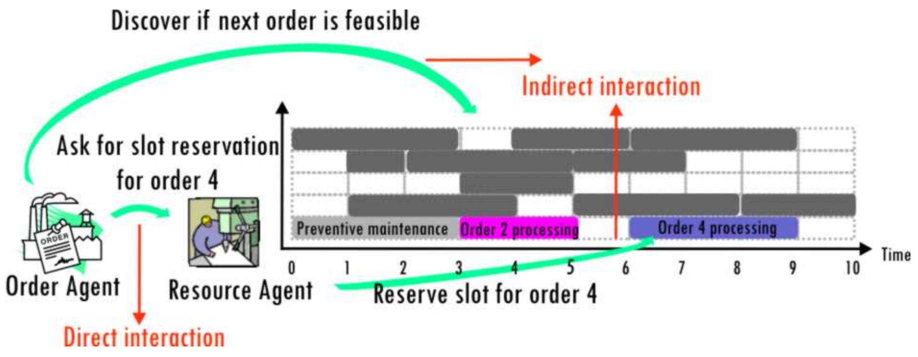

Figure 6. Direct and indirect interaction at the dynamic Gantt world

In this simulation world, order agents (clients) negotiate with resource agents (servers) the provision of required resources to process an order and the dynamic scheduling emerging as a result of such interactions are perceived in a dynamic Gantt chart. There, each resource is a row of the chart and there is a dynamic temporal window with a fixed period of time from the actual time to the future showing if the resources are reserved for a specific order type (colored box) or not (only the grey row), or whether the resource is suddenly broken or prepared to be in maintenance (black box).

3.3 Prototype Implementation in Netlogo

A computational model of the distributed scheduling system was implemented in Netlogo® [13], a software environment specifically designed for generative modeling of artificial agent societies. The Netlogo world is made up of agents. Agents are entities that can follow instructions and develop a decision-making policy. Each agent carries out its own activity, in an asynchronous manner regarding other agents.

In Netlogo, there are four types of agents: turtles, patches, links, and the observer. Turtles are agents that move around in the world. The world is two dimensional and is divided up into a grid of patches. Each patch is a square piece of "ground" over which turtles can move. Links are agents that connect two turtles. The observer doesn't have a location, it is constantly looking out over the world of turtles and patches. For intelligent distributed scheduling, the different tasks involved in each order production were exploded in mobile agents (turtles), one for each time unit, that move over a grid of stationary agents (patches) which represents resource usage times. A set of agents (agentsets) were used to differentiate order types and so the corresponding mean processing times and arrival rates.

The dynamic Gantt chart of the Fig. 7 was made, instead of moving to the right the time window following the system time, asking the turtles (unit time tasks) to move one unit to the left side and to die when they arrive to the patch of the left (actual time). In this way the relative effect is the desired one and the unit time task disappears while it is being executed.

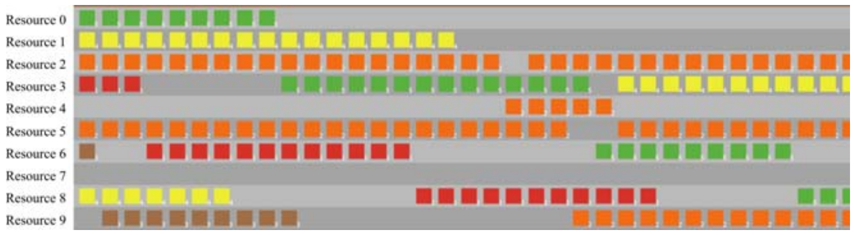

Figure 7. A view of the resultant dynamic Gantt chart

To program the model, global variables were used when the information was public (available for every agent), and turtle variables and patch variables when they were sole private information for a single agent or type of agent, or when their values changes depending on the turtle or patch.

CASE STUDY

4.1 A multi-product batch plant

The proposed design for the holonic MES was modeled for a multiproduct batch plant comprising of 4 stages and 10 units to obtain 5 different products (Fig. 8). Each order has different attributes such product type and due date whereas arrival times, processing times and machine failure rates are stochastic. A batch (order) can follow many different routes through the batch plant while using different pieces of equipment. So there is a great deal of flexibility in order scheduling and it is not obvious how smoothly each order will flow through the manufacturing plant due to a number of disruptive events. It is assumed that order agents have the objective of decreasing late deliveries, so they have decision-making logic needed to choose from the list of solutions the earliest global lead-time alternative. Resource agents resort to the FIFO dispatching rule for request processing at their resources.

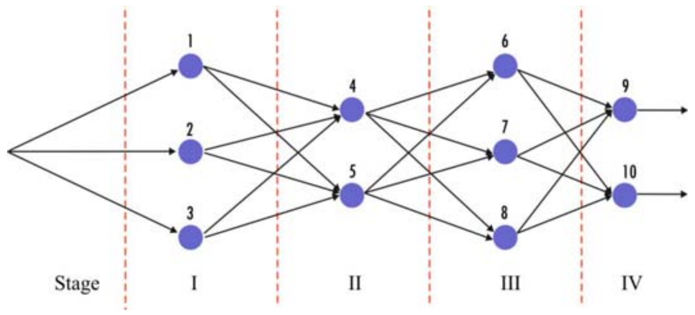

Figure 8. Multi-product plant network structure

4.2 Simulation Results

Fig. 9 exhibits the dynamics of the plant for the normal operating scenario. As can be seen the total processing times of both order types shown tend to stabilize roughly after a time equal to 1000 mins. Fig. 10 shows the impact of total processing time for the order types 2 and 3 when orders of type 3 experiments and instantaneous increase in the processing time at stage II.

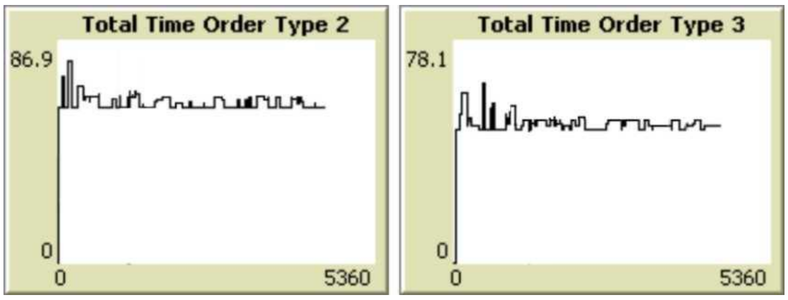

Figure 9. Total processing time for order types 2 and 3 for the normal scenario

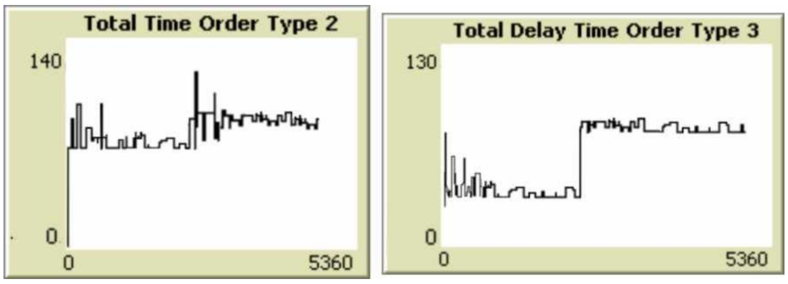

Figure 10. Total processing time for order types 2 and 3 for the scenario when order type 3 becomes more demanding of processing time at stage II since time 2000 min.

Fig. 11 describes the dynamic response of the holonic MES when the resource 1 experiments a sudden increase in its breakdown rate. Order type 3 is somewhat affected whereas type 2 orders are severely disrupted in their processing times even when this order type is not processed at this resource. At Fig. 12, the processing times for the same type of orders are shown, for the scenario where order type 2 suddenly become rush orders. Type 3 orders type are affected by this priority change but then their processing time stabilizes again, yet at a higher average whereas the type 2 orders do not suffer any sudden disruptions and their processing time stabilizes at a lower average.

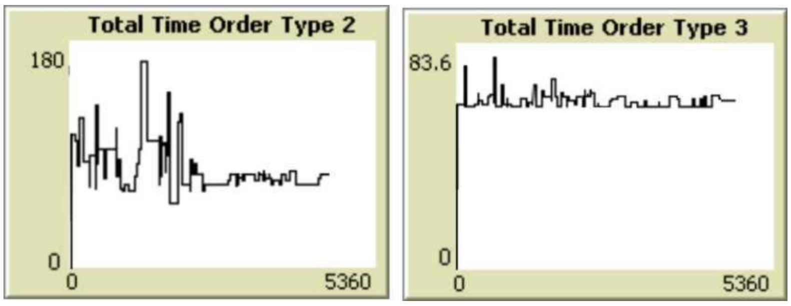

Figure 11. Total processing time for order types 2 and 3 for the scenario where an increase of resource 1 breakdown rates between minute 1000 and 2000 is simulated

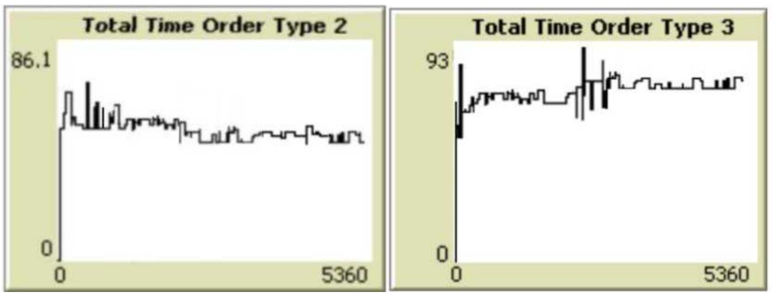

Figure 12. Total processing time for order types 2 and 3 for the scenario where type 2 orders change its priority from normal o rush orders at the 2000 min.

CONCLUSIONS AND FUTURE WORKS

This paper proposes a novel design of an intelligent distributed scheduling based on well-defined interactions between autonomic agents which manage two different types of objects: orders and resources. The design methodology highlights agent interactions to provide a detailed overview of a holonic MES dynamics. To asses the proposed holonic design a generative simulation model has been implemented. Results obtained for different simulated scenarios indicate that the proposed MES is robust and stable despite the total autonomy given to order and resource agents when negotiating resource usage without resorting to a priory defined schedule.

Future research includes the implementation of a more sophisticated prototype with heterogeneous order agents and resource agents with different objectives and strategies to maximize resource usage. Upcoming work will also address more complex manufacturing systems, for instance related to a more complicated set of equipment interconnections, mixed orders and re-processing.

References

[1] Valckenaers, P., Van Brussel, H. Holonic Manufacturing Execution Systems. *CIRP Annals- Manufacturing Technology*, Vol. 54, (2005), pp. 427-432.

[2] Verstraete, P. et al, Towards robust and efficient planning execution, *Engineering Applications of Artificial Intelligence*, Vol. 21, (2008), pp. 304-314.

[3] Leitao, P. et al., ADACOR: A collaborative production automation and control architecture, *IEEE Intelligent Systems*, Vol. 20, (2005), pp. 58–66.

[4] Rolón, M. et al. (2009), Agent Based Modelling and Simulation of Intelligent Distributed Scheduling Systems, *Proceedings of the 19th European Symposium on Computer Aided Process Engineering* – ESCAPE19, Elsevier B.V. (in press)

[5] Kephart, J. O. and Chess, D. M. The vision of Autonomic Computing. *Computer.* Vol. 36, N° 1, (2003), pp. 41-50.

[6] North, M. Macal, M. *Managing Business Complexity, Discovering Strategic Solution with Agent-Based Modeling and Simulation*, Oxford: Oxford University Press. 2007.

[7] Kletti, J. (Ed.). *Manufacturing Execution Systems – MES*. Berlin: Springer-Verlag. 2007.

[8] Holonic Manufacturing System (HMS) consortium, *http:// hms.ifw.uni-hannover.de*/, Last access 05/02/2009.

[9] Gilbert, N. *Agent-based models*. Gildford: Sage Publications. 2008.

[10] Miller, J.H. and Page, S.E. *Complex Adaptive Systems: An Introduction to Computational Models of Social* Life. Princeton: Princeton University Press. 2007.

[11] Schelling, T. C. Micromotives *and Macrobehavior*. New York: Norton. 1978.

[12] Epstein, J. M. *Generativ*e *Social Science: Studies in Agent-Based Computational Modeling*. Princeton: Princeton University Press. 2006.

[13] Wilensky, U. (1999), *Netlogo Modeling Environment*, available at http://ccl.northwestern.edu/netlogo, Last access 05/02/2009.