J Intell Robot Syst (2010) 58:17–45
DOI 10.1007/s10846-009-9343-9

Semi-heterarchical Allocation and Routing Processes in FMS Control: A Stigmergic Approach

Thierry Berger · Yves Sallez · Bruno Valli ·
Alain Gibaud · Damien Trentesaux

Received: 13 November 2007 / Accepted: 10 April 2009 / Published online: 9 June 2009

Abstract This paper deals with the production process control in flexible manufacturing systems (FMS), in which heterarchical relations exist between some decisional entities. After presenting a brief state-of-the art of the literature on the heterarchical concept we propose a semi-heterarchical control structure (composed of DAP: dynamic allocation process and of DRP: dynamic routing process), and explain the objective of our study. After presenting the concept of stigmergy, we focus in this paper on our innovative approach to routing in DRP including the active product concept. We then describe our two levels model and its main components (a virtual level VL in which virtual active products evolve stochastically in accelerated time, and a physical level PL in which physical active products evolve deterministically in real time). Our innovative approach exploits the capacity of a stigmergic routing control model to automatically find efficient routing paths for active products in FMS undergoing perturbations. After a brief presentation of the Netlogo simulation context, the qualitative and quantitative results are presented. The results illustrate the advantages of our routing approach and its capacity to surmount perturbations. The integration and implementation of our approach at the AIP-PRIMECA center in Valenciennes France is then detailed. Finally, we provide a brief overview of our future research concerning: firstly, a way to link our DRP model with the DAP distributed control system, secondly, the re-formulation of our model within the HMS (holonic manufacturing system) concept, and thirdly, the development of a new challenging and innovative concept of "hypervision".

This paper is in memory of Prof Noël Malvache, his convivial and inquisitive spirit continues to inspire us all.

T. Berger (✉) · Y. Sallez · B. Valli · A. Gibaud · D. Trentesaux
LAMIH-CNRS, Le mont Houy, University of Valenciennes and Hainaut-Cambrésis,
59313 Valenciennes cedex 9, France
e-mail: thierry.berger@univ-valenciennes.fr

Keywords Stigmergy · Heterarchy · Dynamic routing · Active product · Manufacturing control · Flexible manufacturing system

1 Introduction

This paper addresses the issue of production process control in flexible manufacturing systems (FMS), in which heterarchical relations exist between some decisional entities.

First we introduce the global assumptions on which our proposal relies. Then, we present a brief state-of-the-art of the literature and explain the objective of our study. Next, we define what we mean by heterarchy. Since the task allocation control has already been studied [37], we focus here on our innovative stigmergic-based approach to routing and the integration and implementation of this approach within the procedures for production process control.

2 Global Assumptions

To situate our study in the overall domain of FMS, it is very important to introduce the main assumptions on which our study is based and on which our model relies.

1. *The topology of the FMS transportation system* is assumed to be associated to a strongly connected, directed graph, in which nodes can be both resources and disjunction points, and arcs are the parts of the system that require no decisions during the routing process since the product can only move in one direction towards the next node. Routing times are assumed to be non-negligible compared to production times and non necessary constant. Resources are also assumed to be flexible, meaning that the same operation type can be executed on several resources, with the possibility of different processing times.
2. *Products and production resources* are assumed to be decisional entities. Usually, these products and resources are considered to be passive entities: they never communicate, decide nor act during production process. However, current technology (e.g., RFID, smart cards, embedded systems, wifi, infrared communication) has recently led to some new activities involving "active" decisional entities, in which products and resources are able to act in accordance with the real state of the production system. This new ability can be embedded in the product or resource itself [26]. In this paper, products and resources that can be taken as decisional entities are denoted AP (active products) or AR (active resources).
3. The studied production control system must control the production process in real time. Consequently, its *inputs* are assumed to be information about planned manufacturing orders optimized over the long term (what to do and when?), coming, for example, from an ERP (enterprise resource planning) system. The *controlled production process* is assumed to be composed of several activities, involving mainly the execution of manufacturing orders using production resources and the routing of products within the production system. Given this assumption, dynamic production control systems must deal with many issues: dynamic scheduling/task allocation (who undertakes the tasks required for each manufacturing order?), tool and inventory management, reconfiguration/preparation of

resources, transportation system management, performance indicator updating, and so on.

4. The structure of the production control system is assumed to be non-centralized (i.e., no central data management system can be identified) and not fully hierarchical, which means that non-hierarchical relationships can and do exist in the system (the concept of a non-hierarchical control structure will be described more precisely in Section 4).

3 State-of-the-Art and Objectives

This section provides a brief state-of-the-art of the literature in the domain of production control systems that are not fully hierarchical. In this context, three main modelling approaches can be identified: multi-agent, holonic and bio-inspired approaches.

3.1 Multi-agent Approaches to Production Control

Historically, the initial studies about systems that are not fully hierarchical (e.g., [14] as well as their other studies) did not explicitly refer to multi-agent systems (MAS) and/or contract net negotiation protocols [39], but these studies have inspired many of the current multi-agent models.

Trentesaux et al. [42, 43] defined a multi-agent control system in which resources/agents negotiate using the contract-net protocol to decide in real-time to which resource the current product would be best scheduled, considering simultaneously the real-time resource workloads and pre-determined travel times. If the standard deviation of the workload is too high, a bottleneck resource is identified, and this bottleneck resource is scheduled using a single machine algorithm to maintain the system's overall performance.

Chen et al. [8] introduced a MAS mobile enterprise information portal used to establish real-time data entry for real-time production controls across intra-/inter-organizational supply systems. This portal is used to help the production controller making effective decisions in "just-in-time" manufacturing systems.

Ouelhadj et al. [31] proposed a negotiation protocol based on the Contract Net Protocol, in which agents cooperate and coordinate their local schedules in order to find global near-optimal robust schedules, whilst minimizing the disruption caused by unexpected real-time events.

Reaidy et al. [35] presented a multi-agent system that uses a cooperation and competition paradigm, in which agents are both products and resources of the system and local scheduling is accomplished through a negotiation protocol between agents.

Bousbia et al. [5] and Aissani et al. [1] have introduced reinforcement learning techniques in the agents of multi-agent systems to provide not only real-time effective scheduling but also efficient continuous improvement of the system's ability to react optimally to events. In these MAS, products and resources are agents that negotiate and learn through interaction.

In fact, multi-agent production control systems are widespread, and most of studies focus on the dynamic scheduling of resources through negotiation paradigms, such as contract nets. For more information, the interested reader can consult Cavalieri et al. [7], Lastra and Colombo [22], Maione and Naso [29] or Wong et al. [52].

3.2 Holonic Approaches to Production Control

Holonic manufacturing systems (HMS) transpose the concepts developed by Koestler [20] for living organisms and social organizations to the manufacturing world. Holonic manufacturing is characterized by holarchies of autonomous and co-operative entities, called holons, which represent the entire range of manufacturing entities. A holon, as devised by Koestler, is an identifiable component of a (manufacturing) system that has a unique identity, yet is made up of subordinate components, and in turn is part of a larger whole [23]. The HMS concept is widespread, particularly due to the success of the reference architecture, PROSA, proposed by Van Brussel et al. [46]. According to this holonic approach to manufacturing systems, a holon can be distinguished from an agent because it is composed of recursive holons that have both an informational and a physical component.

Gou et al. [16] proposed a holonic scheduling system for a manufacturing factory. In this system, the cooperation mechanisms among holons are established based on the pricing concept of the market economy according to the mathematical optimization technique, Lagrangian relaxation. Inter-cell cooperation is performed without accessing the individual cell's local information nor intruding on their decisional authority.

Sousa and Ramos [38] proposed a holonic model where holons represent tasks and resources. The Contract Net Protocol was adapted to deal with temporal constraints and scheduling conflicts. The aim of this model was to dynamically assign operations to manufacturing system resources in order to accomplish the proposed tasks.

Ulieru and Norrie [44] proposed a distributed fault recovery technique for designing re-configurable holonic manufacturing systems based on the fuzzy entropy concept. The emerging new structures can accomplish fault-recovery by re-distributing tasks in cases of resource failures, for example.

Baïna and Morel [4] addressed the issue of interoperability of company applications. Their model aims to synchronize both the physical and informational views. A model-driven approach for interoperability is proposed to insure the interoperability of the holonic models with the other applications used by the company.

Verstraete et al. [47] have recently proposed a holonic manufacturing execution system that cooperates with a planning system. This cooperation allows the robustness and flexibility of this system to be combined with the optimization offered by the planning system.

Hsieh [19] has proposed an original formal approach for developing collaborative algorithms to guide the holons to form a holarchy that will move coherently toward the ultimate desired goal state. He has adopted the contract net protocol to model the mutual selection of holons when forming the holarchy. The holarchy optimization problem has been designed to minimize the costs, subject to feasibility constraints.

3.3 Bionic Approaches to Production Control

Bionic approaches exploit the properties of natural systems, transposing biological abilities (e.g., self-organization, evolution, learning and adaptation) to machine design, specifically production control systems in which entities are autonomous and self-organized [30].

One bionic, or bio-inspired, approach applies stigmergic principles to machine or system design. Stigmergy refers to a mechanism of spontaneous, indirect coordination between entities, in which the marks left in the environment by an action stimulates the completion of other subsequent actions. This concept is described more precisely in Section 7.1.

The first experiments related to the industrial use of stigmergy were conducted in the early 1980s by Deneubourg et al. [10], who simulated “ant-like robots”. Since then, many researchers (e.g., [2, 11, 15]) have applied this concept when studying robot collectives and working to solve optimization problems (e.g., travelling salesman problems, network routing for telecommunications and the Internet). Based on the ant foraging analogy, Dorigo and Stützle [13] developed the ant colony optimization (ACO) metaheuristic, a population-based approach to solving combinatorial optimization problems. The basic idea behind ACO is that a large number of simple artificial entities can be used to build good solutions to hard combinatorial optimization problems via low-level communications. The ACO approach can be applied to almost any scheduling problem, such as job shop scheduling and vehicle routing, for example.

Researchers have also applied the stigmergy concept to specific situations in manufacturing control systems:

Brückner [6] applied the stigmergy concept to manufacturing control, and his application is supported by an agent-system approach. He presented an extensive set of guidelines that can be used to design synthetic ecosystems. In this 2000 study, different types of agents were used to model the various elements (e.g., resources, part flow and control units) involved in routing a car body through a Mercedes Benz paint shop. Brückner’s work is based on the PROSA reference architecture [53].

Parunak et al. [32] concentrated on the importance of the environment in agent systems, in which the information flows through the environment complement the classic message-based communications between the agents. In this 2001 study, the environment was computational, and agents moving over a graph were used to study manufacturing company supply networks. The authors focused on the dynamics that emerge from the interactions in multi-agent systems, analyzing these dynamics using methods inspired by statistical mechanics.

Peeters et al. [34] and Hadeli et al. [18] both proposed a pheromone-based control algorithm with a bottom-up design. Like Brückner, Peeters et al. based their work on the PROSA reference architecture [53] (those interested should consult the Mascada-WP4-Report [24] for a description of both the agents and the pheromone life-cycle in the routing of a car body through a paint shop, this one at Daimler-Chrysler). Hadeli et al. [18] emulated a simple flexible manufacturing system characterized by dynamic order arrivals, probabilistic processing times, and several disturbances (e.g., machine breakdowns), with the objective of evaluating the possibility of creating short-term forecasts based on agent intentions. Valckenaers et al. [45] presented a design for the emergent generation of short-term forecasts in multi-agent coordination and control systems. This design was inspired by the food foraging behaviour in ant colonies.

3.4 Synthesis and Objectives

According to surveys of the literature [3, 40], most of the research in the context of multi-agent and holonic approaches to control systems focuses on distributed or decentralised dynamic control of dynamic scheduling. Of course, in other specific fields, both short-term (e.g., re-configuration) or long-term (e.g., planning, supply-chain, interoperability) time windows are addressed. Our initial work in this domain led us to propose a control system for a dynamic allocation process, based upon a product-driven and resource-driven heterarchical control of dynamic scheduling [37]. However, in order to apply the concepts relevant to this domain in a real FMS, dynamic scheduling, planning and reconfiguration are not sufficient; several other process controls must also be developed (e.g., tool management, inventory level control, routing process control). However, little research has been done that takes those real-time controls (i.e., real-time time window) into account, especially in terms of routing problems (e.g., jamming, conveyor system breakdowns, routing path modifications).

Several stigmergic applications have tackled routing problems, mainly the problems of information routing on the internet. However, in these applications, time is not really considered; failed attempts cost the user nothing and only the final solution is taken into account. In a production context, quite the reverse is true: time inefficiency is prejudicial to the system. For example, if too many solutions are tried before finding the right one (e.g., too many shuttles are used to test paths), the overall performance quality will rapidly deteriorate, for instance, in terms of completion times. Consequently, a model is needed that will identify satisfactory solutions quickly in order to accelerate the convergence of algorithms as much as possible.

The aim of this paper is to present a stigmergic routing process control coupled with our existing dynamic allocation process control based upon a classic negotiation paradigm. In response to the usual criticisms of stigmergy (e.g., time inefficiency, numerous attempts needed to find a solution), we propose an original functional architecture with two-levels—physical and virtual—for controlling the routing process.

The next section first defines precisely what we mean by heterarchy, which is important since it is a key concept in the design of our new control structure. Then, Section 5 describes the integration of the two control sub-systems (dynamic allocation and routing) within the new control structure and Section 7 presents the decomposition of the routing control sub-system into two levels: physical and virtual.

4 Heterarchy

In our opinion, heterarchy can be formalized by using graph theory. A directed graph composed of nodes representing decisional entities and arcs representing the master–slave interaction of a decisional entity (master) with another entity (slave) is called "influence graph". Since each node can be considered both a master and a slave, no hierarchy can be identified. The graph is thus considered to be strongly connected, which defines a heterarchy. This formalisation is consistent with the initial heterarchy concept developed by McCulloch [25]. Figure 1 illustrates the difference between hierarchy and heterarchy graphically.

Thus, hierarchy and heterarchy are both relevant to the structure of a system composed of several decisional entities (cf. Fig. 2). Based on these two concepts, different structures can be formed, ranging from fully hierarchical (no heterarchical relationship) to fully heterarchical (no hierarchical relationship). We use the term "semi-heterarchical" to refer to a structure that can not be assimilated to a fully heterarchical system while it contains at least one sub-graph that is fully heterarchical.

5 Proposed Control Structure

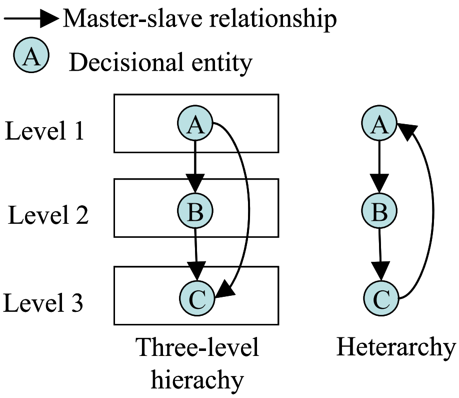
A production control system is designed to determine in real time the list of each successive task (operation, date) that an active resource AR has to realize, taking into account both allocation and routing issues. Thus, the two processes that must be controlled are the dynamic allocation process (DAP) and the dynamic routing process (DRP).

The proposed semi-heterarchical control structure is shown in Fig. 3. In this structure, allocation is controlled prior to routing.

Inputs to the DRP control system are assumed to be composed of the set of couples (*ns*,*nd*) — where *ns* is the active resource source node and *nd* the active resource destination node — that concern one or more products at a given time t. This information is defined by the DAP control system. Outputs from the DRP control system are the optimized real transportation times for routed products. This information will be used by the DAP control system to improve its allocations. Relationship between DAP and DRP control systems is then hierarchical.

Meanwhile, the global structure is partially heterarchical since each of the two control systems is fully heterarchical: no hierarchy can be identified in any of the individual control subsystems (decisional entities) and there is neither centralized data management nor centralized processing capability.

Fig. 1 Hierarchy and heterarchy



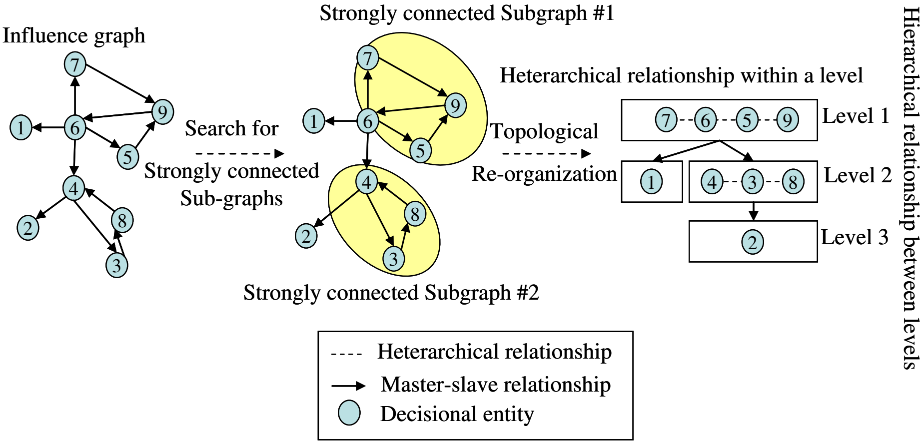


Fig. 2 Hierarchy and heterarchy as building blocks of a system composed of decisional entities

A fully heterarchical structure in which the DRP and DAP cooperate to optimize allocation and routing jointly is possible, but such a structure generates new issues that are currently being studied and have yet not been published.

The following provides the control algorithm supported by our previous control structure model.

6 Proposed Control Algorithm

Our initial research dealt with a theoretical heterarchical DAP control [37]. Here, we improve on this theoretical control model, by combining it with the DRP control in a more global control algorithm, which allows the hierarchical articulation of the two control systems. This new approach is fully reactive since it assumes that manufacturing orders have been planned and optimized using a wider time horizon window. Since our goal is to validate our model on real embedded systems, a real-time

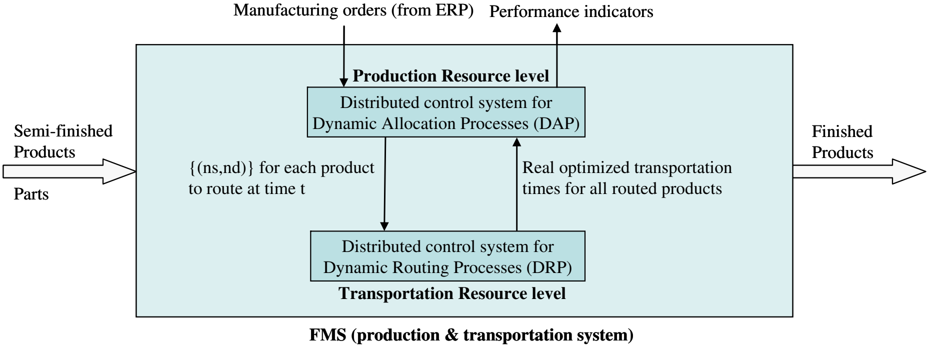


Fig. 3 Proposed semi-heterarchical control structure

programming approach was used (e.g., reduction of the number of loops and non-blocking states).

The following are the main notations used in our model.

Let $N = \{n_i\}$ be the set of considered nodes n_i (active resources—AR) or disjunction points in the transportation system, and $X = \{x_j\}$ be the set of active products—AP. Both AR and AP are decisional entities. For each AR, an input zone exists where an AP waits for a task (i.e., operation to be executed) and an output zone where an AP waits to be transferred. Figure 4 illustrates these notations and shows the two decisional entities AR and AP.

The main algorithm (below) features the two kinds of decisions made by AR and AP:

```
When an operation is completed by AR $n_i$ on AP $x_j$
    AP $x_j$ is downloaded
    AR $n_i$ decisional algorithm is triggered
    AP $x_j$ decisional algorithm is triggered
```

These two decisional algorithms are described in the next two sections.

6.1 Active Resource Decisional Algorithm

When an AR has finished an operation on AP x_j, it downloads x_j to its output zone if space is available (pre-conditions). Then the AR analyzes its input zone, if several AP are available, it chooses the "best" next x_k. The algorithm responsible for this decision has two steps:

```
1. AR $n_i$ identifies next AP $x_k = g(D)$
2. AR $n_i$ loads AP $x_k$
```

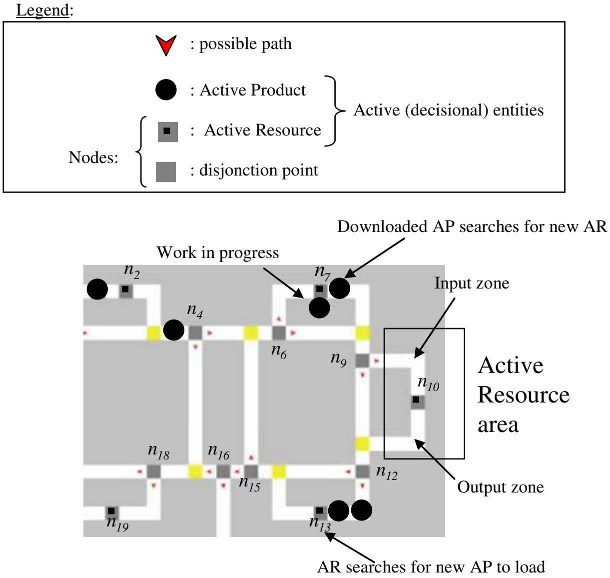


Fig. 4 Notations and decisional activities

In this algorithm, the function `g` depends on evaluating a criterion `D` that is defined according to the global objective assigned to the control system. The choice for the best criterion to achieve this objective is still an open issue. However, `D` is usually based upon such information as shortest processing time, shortest remaining time/tasks or first-in first-out to name a few. The best product to load is the one that proposes the best value for `D` through mono-criterion (min or max) or multi-criteria analysis.

Steps 1 and 2 are supported by the DAP heterarchical control system.

6.2 Active Product Decisional Algorithm

When an operation is completed on an AP, the AP enters the output zone of the AR that has executed the operation. It must then choose (decide) an AR for the next operation (service). To do so, the AP uses the following three-step algorithm:

```
1. AP $x_j$ identifies the next operation to execute on itself
2. AP $x_j$ reserves the "best" destination AR $n_d$ = f(D',M)
3. AP $x_j$ moves from current node resource AR $n_s$ to AR $n_d$
```

The `f` function depends on two parameters: the criterion `D′` used to optimize the operation and the method `M` used to gather information. The best choice for criterion `D′` is, like the choice for `D`, open to debate, but this criterion is usually based upon such information as next available time, shortest remaining number of operations to proceed and/or transfer time, for example. In a heterarchical context, one of the most usual methods chosen for `M` is based upon the contract-Net protocol [39], but a blackboard system could also be used.

Steps 1 and 2 are supported by the DAP control system. Step 2 applies the contract-net to define a negotiation protocol dedicated to dynamic task allocation. This negotiation protocol helps to identify the "best" AR destination for each requesting AP, as follows: when an operation in a particular manufacturing order has been performed, the subsequent AP x_j sends a *request* through the communication network about the next operation to be processed (according to the routing list of the AP's manufacturing order). Each of the AR n_k able to perform this operation returns an *acceptance* and enough local information to allow x_j to evaluate the response (in our case, the estimated next available time, given the size of the product queue and the work in progress). The requesting AP x_j then selects one of the proposed AR by minimizing the `D′` criterion (i.e., the next available time, given the current real travel time to each AR candidate provided by the DRP system). The requesting AP x_j sends a *reservation* to the selected destination AR n_d and a *release* to the others $n_{k \neq d}$. A *discharge* sent to x_j from the selected AR n_d concludes the protocol and validates the allocation "contract". Then, n_d is given, AP x_j being at AR n_s. Step 3 can thus be realized.

Step 3 is supported by the DRP control system. The input for the DRP control system is the set of couples (n_s,n_d). After the routing operation has been executed, $\tau^*_{nd}(n_s)$, denoting the minimal real-time to go from AR n_s to the destination AR n_d, is then updated by the DRP control system. This value is sent back to the DAP control system each time the `D′` criteria must be estimated. As a result, the real state of the transportation system can be taken into account when determining the "best" AR n_d.

The hierarchical loop between the two control systems is then highlighted. This is not a standard regulation mechanism (control loop in continuous or discrete time), but rather an event-driven information exchange between the two levels given that when an AP needs to move to one AR (as a result of the DAP control system), the DRP control system is triggered and must manage this travel without any possibility of reconsidering n_d since the DRP is hierarchically dependent on the DAP.

The following section underlines the innovative aspect of our approach and describes in detail the DRP control system, its simulation and its validation on a real FMS cell. The validation of the coupling of the two control systems will be presented in another paper.

7 The DRP Control System

7.1 Concept of Stigmergy

To make the DRP adaptive and robust, our approach is based on the concept of stigmergy.

7.1.1 The Stigmergy Concept

French entomologist Grassé [17] introduced the term "stigmergy" to describe the mechanism by which termites coordinate their mound-building activities. In such activities, many individuals participate in a collective task, and the stimuli provided by the emerging structure are used to coordinate the individual actions. A similar mechanism is used by ants laying down pheromone trails between a food source and their nest.

Figure 5 portrays a classic experiment in entomology [12]. In this double bridge experiment, the ants collectively discover the shortest path to a food source. Several successive stages can be distinguished:

- Ants wander around their nest in search of food (arrow 1 in Fig. 5a).
- Those finding food carry it back to the nest, simultaneously depositing a chemical substance, called a pheromone, which attracts other ants (arrow 2 in Fig. 5a).
- Other ants, detecting the pheromones, follow the trails back toward the food (Fig. 5b).
- When ants arrive at a bifurcation point, they make a probabilistic choice, based on the intensity of the pheromone: the higher the pheromone concentration found on a particular path, the higher the probability that the ant will follow that path. The first ants to arrive at the food source are those that took the two shortest branches. When these ants start their return trip, more pheromone traces are present on the shortest path than on the other paths. This behaviour has an autocatalytic effect, and thus more and more pheromones are released on the chosen path.
- This effect is accentuated by pheromone evaporation. Since pheromones tend to evaporate over time, it is more difficult to maintain a stable pheromone trail on a longer path. Over time, due to the natural reinforcement of the ants, only the shortest trail remains (Fig. 5c).

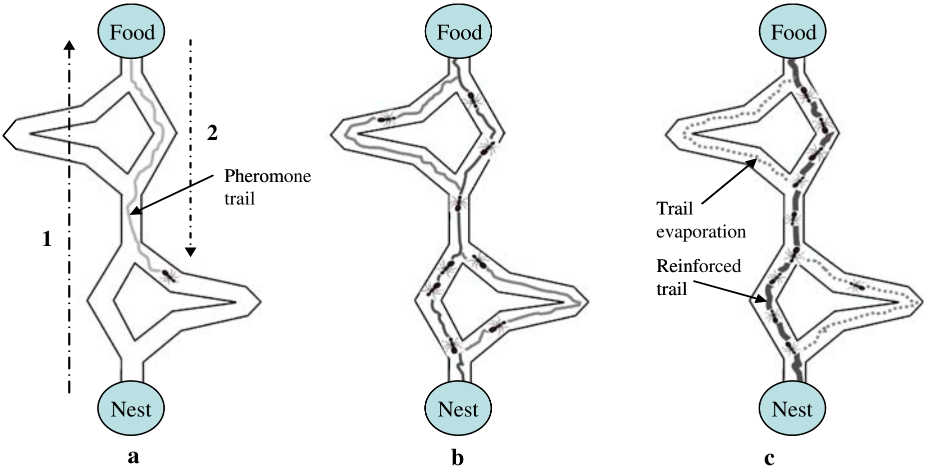


Fig. 5 Stigmergy illustration

As the experiment illustrates, this stigmergic process has a natural optimizing effect. For more information about the history of stigmergy in the context of social insects, see Theraulaz and Bonabeau [41]. Applications in the areas of combinatorial optimization, collective robotics, and routing in networks are also reviewed in White [50].

7.1.2 Basic Operations

In the real world, three basic operations have been associated with the stigmergic process: information fusion, information removal and local information distribution. In the first, deposits from individual entities are aggregated to allow the easy fusion of information. In the second, pheromone evaporation over time is used to remove obsolete or inconsistent information. In the last, information is provided according to pheromone diffusion in the immediate (local) neighbourhood.

In all of these operations, the pheromone field has three main characteristics:

- *Independence*: The sender of a pheromone message does not know the identity of the potential receiver and does not wait for any acknowledgement, which makes pheromone use very effective for communication within large populations of simple entities.
- *Local management*: Because pheromone diffusion falls off rapidly with distance, pheromone interaction remains local, thus avoiding the need for centralized interaction management.
- *Dynamism*: The continuous cycles of reinforcement and evaporation act respectively to integrate new information and to delete obsolete information.

7.1.3 Pheromone Emulation

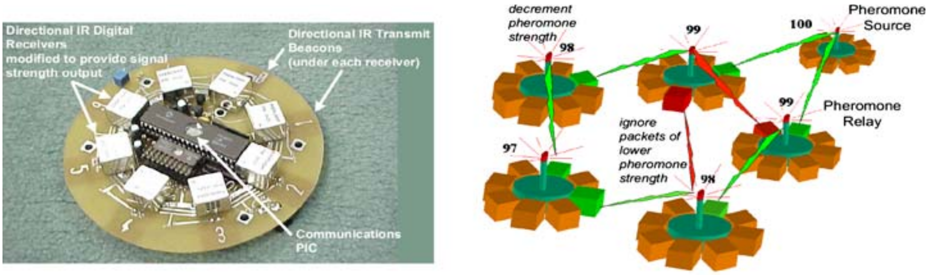
A variety of different approaches to pheromone emulation have been developed. The three most common types are described below:

Common memory In many studies (e.g., [11]), artificial ants cooperate via a common memory that serves the same purpose as the pheromones deposited by real ants. In this common memory, an artificial pheromone is created and accumulated (updated) via a learning mechanism during runtime.

Embedded implementation Using signals transmitted from robot to robot, Payton et al. [33] have implemented “virtual” pheromones that are sustained on board. Simple beacons and directional Ir sensors are mounted on the robots (Fig. 6a), with the virtual pheromones attached to the robots rather than laid down in the environment. Figure 6b shows the pheromones propagation from robot to robot. This particularity is necessitated by the application: guidance of a rescue team in an unfamiliar building.

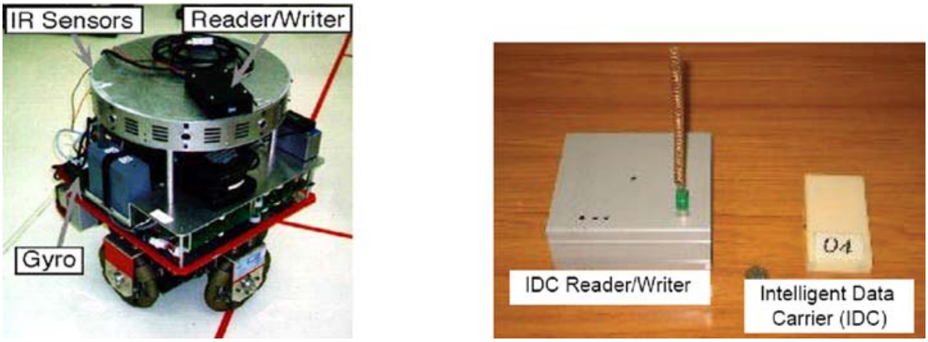
Direct environmental marking This kind of pheromone emulation can be performed in two ways: using real “pheromones” and using deposit tags.

- Real pheromones have been used by researchers in Australia [36] and in Israël [48] to emulate ant behaviour by creating robots capable of laying down and detecting chemical trails (camphor or thinner).
- Deposit tags, such as the Intelligent Data Carrier (IDC), have been developed by Japanese researchers [21]. The IDC system is composed of reader/writer units attached to mobile robots and tags that are carried and located by the robots. These tags are analogous to pheromones in that they store the information used to guide the robots (Fig. 6c).



a. Transceiver for virtual pheromones

b. Pheromone propagation



c. Intelligent Data Carrier (robot and tags)

Fig. 6 Examples of existing pheromone emulation hardware

7.2 Modelling Approach

Figure 7 shows the proposed architecture with its two levels and the intermediary data space (DS) that memorizes the information for both levels:

- The virtual level (VL), where virtual active products (VAPs) move, is an informational domain in which everything is simulated in accelerated time as quickly as possible;
- The physical level (PL) represents the real world, in which physical active products (PAPs) move in real-time.

A node n_i (disjunction point or AR) has three components: a virtual image (in the VL), a data memorization and processing structure (in the DS), and a physical infrastructure (in the PL). Since VL works in accelerated time, adaptation time becomes physically short. On the VL level, lots of VAPs can be used to discover new paths, which helps to keep the number of PAPs circulating on the PL level low. The virtual entities (VAPs) make decisions based on stigmergic principles, which include stochastic decision-making and allow adaptive behavior to deal with unexpected perturbations. VAP travel history is used to update the pheromones (i.e., routing data). From a logical point of view, VAP move within a model of the existing routing network in the PL, which contains PAPs. At the PL level, PAP routing decisions are made deterministically with "best efforts", based upon the optimized results of the VL level.

The following section introduces the notations used in this model, which is then described more precisely.

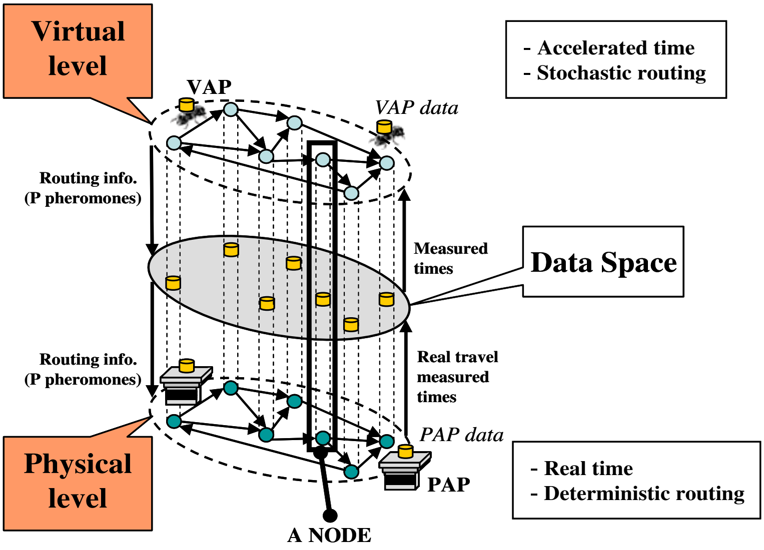


Fig. 7 The two levels architecture

7.3 Notations and Variables

The variable $\nu_{n_i}^{w}$ denotes the w^{th} neighbor of n_i and V_{n_i}, the set of the neighbors of n_i: $V_{n_i} = \left\{ n_j / \exists \text{arc } n_i \to n_j \text{ with } n_i \neq n_j \right\}$, $w \in \left\{ 1 \ldots \text{card}\left(V_{n_i} \right) \right\}$. $N = \{n_i\}$ and V_{n_i} thus describe then the topology of the FMS, including the transportation system. A possible path between node n_i and node n_j is written $u_{ij} = \left[n_i \ldots n_j \right]$ and corresponds to the ranked list of successive nodes to be visited in order to travel from n_i to n_j.

Variables:

- $t_{n_i}\left(\nu_{n_i}^{w}\right)$ and $t_{n_i}^{*}\left(\nu_{n_i}^{w}\right)$ are, respectively, the travel time (measured by a PAP) and the reference time (for a VAP) between a node n_i and its w^{th} neighbor.
- P_{n_i} is the pheromone matrix. Columns of P_{n_i} correspond to all possible destination nodes n_j from n_i and rows contain all existing neighbor nodes $\nu_{n_i}^{w}$. In analogy to biological systems, this matrix characterizes the pheromone rate on the different arcs $n_i \to \nu_{n_i}^{w}$. The value $P_{n_i}\left(\nu_{n_i}^{w}, n_j\right)$ is associated to the preferred path from the current node n_i to destination node n_j, when choosing the possible intermediate neighbor node $\nu_{n_i}^{w}$. Consequently, the following expression is true:

$$\sum_{w} P_{n_i}\left(\nu_{n_i}^{w}, n_j\right) = 1 , \ \forall \left(n_i , n_j\right) \in N^2 .$$

In this paper, the best path is considered to be the path on which the PAPs spend the least time.

Figure 8 gives an example of the use of the above variables. Let a PAP be set on current node $n_i = n_3$, coming from the source node n_1 with the destination node $n_j = n_8$. In this example, the best neighbor is n_4 since $P_{n_3}\left(\nu_{n_3}^{1} = n_4 , n_8\right) = .95 > P_{n_3}\left(\nu_{n_3}^{2} = n_5 , n_8\right) = .05$. PAP will then travel from n_3 to n_4, keeping its destination node n_8 in mind. When arriving at n_4, the time spent $t_{n_3}(n_4)$ is reported to node n_3 via node n_4 and is used by the node n_3 to determine whether or not a perturbation

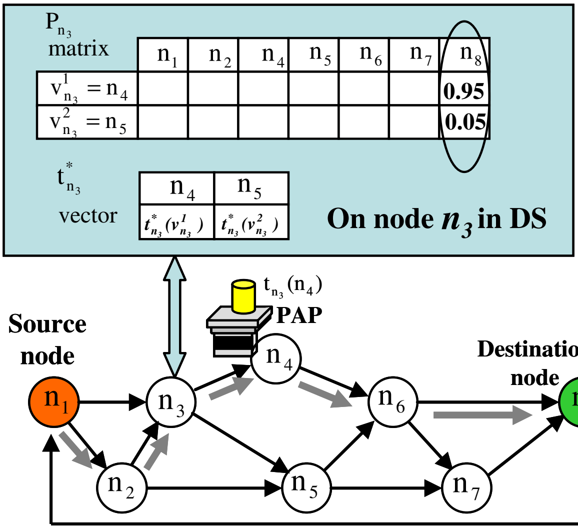


Fig. 8 Variables illustration

(e.g., bottleneck, slow down, break down) has occurred between n_3 and n_4 by comparing $t_{n_3}(n_4)$ with $t^*_{n_3}(n_4)$.

- $\tau^*_{n_i}$ is the minimum time vector. $\tau^*_{n_i}\left(n_j\right)$ thus denotes the minimum measured time to move from node n_i to any possible destination node n_j. This variable is updated every time a lower time is clocked, according to a mobile time window (not described in this paper).
- $h_{vap}(n_i,n_j)$ is the history table. Each VAP traveling between nodes n_i and n_j updates its history table $h_{vap}(n_i,n_j)$, which is used to store the nodes reached and the corresponding arrival dates:

$$\begin{aligned}h_{vap}\left(n_i, n_j\right) = \big(&\left[n_i, n_k, n_m, \ \ldots\ , \ n_u, n_j\right] \ , \\ &\left[d_i, d_k, d_m, \ \ldots, \ d_u, d_j\right]\big)\end{aligned}$$

where d_i is the arrival date at node n_i.

Based on the previous notations, the time spent to travel from n_i to n_j is $\tau_{n_i}\left(n_j\right) = d_j - d_i$.

7.4 Model of the Routing Process Control

Nominal Routing Process Control At a given time, a PAP begins to travel from its AR source node ns to its AR destination node *nd*. The pair (*ns*, *nd*) is provided by the DAP control system. The current node is designated *nc*, thus *nc* = *ns* when the PAP begins to move. The values of $P_{nc}\left(v^w_{nc}, nd\right)$ are used to choose the next neighbor node *nn*. This choice is deterministic since the neighbor with the greater P value is always chosen $nn = \tilde{v}^{nd}_{nc}$ where:

$$\tilde{v}^{nd}_{nc} \in V_{nc} \big/ P\left(\tilde{v}^{nd}_{nc}, nd\right) = \max_{w \in \{1 \ldots card(V_{nc})\}} P_{nc}\left(v^w_{nc}, nd\right)$$

During the real move on the arc *nc*→*nn*, the real elapsed time $t_{nc}(nn)$ is measured. When the PAP arrives at node *nn*, it moves backwards $t_{nc}(nn)$ to node *nc*. A local perturbation is detected on the arc *nc*→*nn* if $\left|t_{nc}(nn) - t^*_{nc}(nn)\right| > \varepsilon$, where ε is a fixed threshold parameter.

VAP Exploration When a perturbation is detected, a VAP exploration is triggered. At each time interval T, a VAP is generated on node *ns* and is assigned the *nd* destination node.

On a current node *nc*, the VAP must choose the next neighbor node *nn*. Since both luck and diversity are important adaptation mechanisms in natural biological systems, the choice of *nn* is not deterministic (as is true for PAP) but stochastic. The different $P_{nc}\left(v^w_{nc}, nd\right)$ are then assigned the probability to reach, from *nc* and to the destination n_j, the neighbor v^w_{nc}. However, if $P_{nc}\left(v^w_{nc}, nd\right)$ is negligible, v^w_{nc} will rarely be chosen. To ensure diversity, a minimum value α is introduced, making the relation:

$$\begin{aligned}&\text{Prob}\left(nn = v^a_{nc}\right) = \max\left(\frac{P_{nc}\left(v^a_{nc}, n_j\right)}{\sum_w P_{nc}\left(v^w_{nc}, n_j\right)}, \alpha\right) = \max\left(P_{nc}\left(v^a_{nc}, n_j\right), \alpha\right) \\ &\text{since } \sum_w P_{nc}\left(v^w_{nc}, n_j\right) = 1\end{aligned}$$

which corresponds to the probability of node v^a_{nc} being chosen as the next neighbor (denoted *nn*).

After reaching node *nn*, the passage date d_{nn} is then stored in the history table h_{vap}. Because the move is virtual, this date is calculated by adding the time reference $t^*_{nc}(nn)$ to the last date in history table.

Then the node *nn* becomes the new current node *nc*, and the VAP continues to travel to the destination node *nd*, iteratively.

Finally, when the VAP reaches the destination node, the routing is finished and the history table is reported to the node *nd*.

Search for the "Best" Alternative Path Like biological ants, which lay down a pheromone trail when returning to the nest, every time a VAP reaches its destination node *nd*, the information contained in its history table h_{vap} is forwarded to each visited node to update pheromone matrices. In our approach, this coefficient updating process is based upon a reinforcement law.

When the forwarded information reaches node *nc*, $\tau_{nc}(nd)$ and the minimal time $\tau^*_{nc}(nd)$ are compared. If $\tau_{nc}(nd)$ is less or equal to the minimal time $\tau^*_{nc}(nd)$ and if *nn* was the neighbor node used to travel from *nc* to *nd*, $P_{nc}(nn, nd)$ is reinforced:

$$\begin{gathered} P_{nc}(nn, nd) \Leftarrow P_{nc}(nn, nd) + r\,(1 - P_{nc}(nn, nd)) \\ \text{With } nn \in V_{nc}. \end{gathered}$$

Coefficients $P_{nc}(no, nd)$ for the destination *nd* of the other neighbors *no* are negatively reinforced through a process of normalization:

$$\begin{gathered} P_{nc}(no, nd) \Leftarrow P_{nc}(no, nd) \; - \; r\; P_{nc}(no, nd) \\ \text{With } no \in V_{nc},\; no \neq nn. \end{gathered}$$

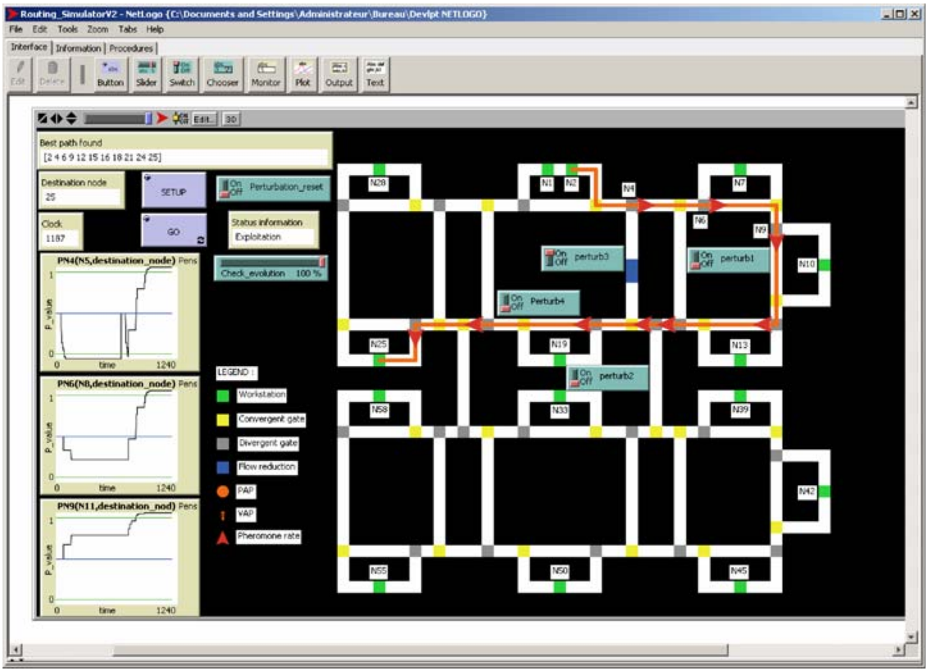
The "best" path emerges when all the P_{nc} coefficients at the reached nodes *nc* are superior to a fixed threshold Ω (0.95 in our study).

8 Simulation of the DRP Control System

8.1 Short Description of the Simulation Tool

The proposed control model is naturally distributed, meaning there is no central memorization and processing system. This property influenced our choice of an agent-based parallel modeling and simulation environment. With NetLogo [51], each modeled entity can be described as an independent agent interacting with its environment. All agents operate in parallel on a grid of patches (i.e., a cellular world) and each agent can read and modify some of the attributes linked to the patches in its proximity. The behavioral rules defined for the agents make it possible to describe agent-environment interaction, which is very important when simulating the stigmergic process.

In the following case study, both the physical and virtual levels are simulated. Figure 9 shows our simulator's interface. An applet is available at url: http://www.univ-valenciennes.fr/sp/routing/.



Best path found
[2 4 6 9 12 15 16 18 21 24 25]

Destination node
25

Clock
1187

Status information
Exploitation

Check_evolution 100 %

Fig. 9 Simulator interface

8.2 Case Study

The proposed approach is currently applied to an FMS built around a conveyer network based on a one-way conveyer system with divergent transfer gates that allow PAPs to be routed toward the different workstations. This FMS, shown in Fig. 10, can be characterized as follows:

- 15 workstations (nodes in medium grey with a black square inside),
- 24 divergent transfer gates (nodes in medium grey), where a routing choice between two adjacent arc must be made,
- 24 convergent transfer gates (nodes in grey with cross bars).

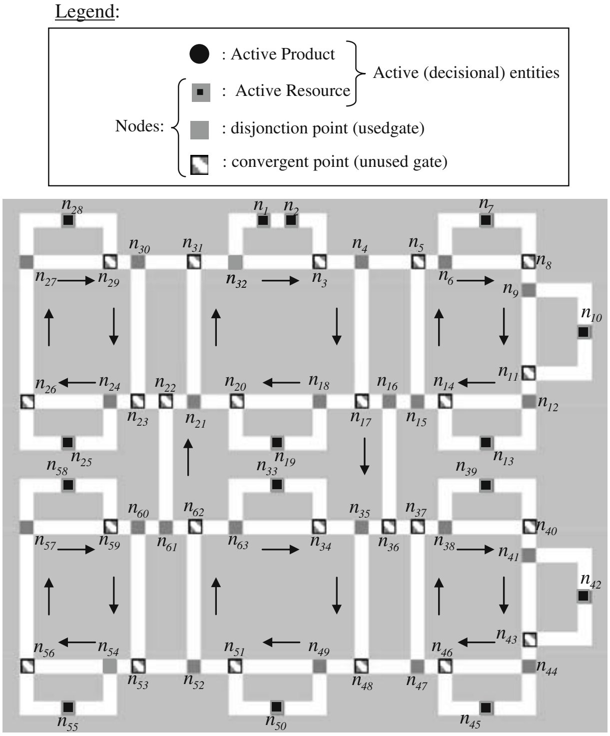
The FMS is divided into two parts—upper and lower—connected by two arcs $n_{16} \rightarrow n_{36}$ and $n_{61} \rightarrow n_{22}$.

8.3 Results

Several scenarii are presented, pointing out the different capabilities of our model with respect to self-adaptation to perturbations. These scenarii are summarized in Table 1.

*Scenario 1: In the first scenario, the impact of a perturbation on an arc that can be by-passed is studied. n_2 and n_{19} are the source and destination nodes, respectively. Qualitative and quantitative results are presented.

Fig. 10 FMS conveying/resource network



8.3.1 Qualitative Results

Figure 11a shows the beginning of the initialization phase. Arrows (near divergent transfer gates) represent the pheromone rate corresponding to the adjacent arcs. Their size is proportional to the pheromone rate (initial values set to 0.5). The VAPs travel randomly along the different arcs, and the path $[n_2, n_3, n_4, n_{17}, n_{18}, n_{19}]$ emerges through reinforcement. At each divergent transfer gate, the bigger arrows indicate the better arcs. The best path is then traced (in black), cf. Fig. 11b.

Table 1 Scenarii description

Scenario	Source node	Destination node	Initial path	Perturbed arc	Alternative path identified
1	n_2	n_{19}	$n_2, n_3, n_4, n_{17}, n_{18}, n_{19}$	$n_4 \rightarrow n_{17}$	$n_2, n_3, n_4, n_5, n_6, n_8, n_9, n_{11}, n_{12}, n_{14}, n_{15}, n_{16}, n_{17}, n_{18}, n_{19}$
2	n_{10}	n_{39}	$n_{10}, n_{11}, n_{12}, n_{14}, n_{15}, n_{16}, n_{36}, n_{37}, n_{38}, n_{39}$	$n_{16} \rightarrow n_{36}$	$n_{10}, n_{11}, n_{12}, n_{14}, n_{15}, n_{16}, n_{36}, n_{37}, n_{38}, n_{39}$
3	n_2	n_{25}	$n_2, n_3, n_4, n_{17}, n_{18}, n_{20}, n_{21}, n_{22}, n_{23}, n_{24}, n_{25}$	$n_{20} \rightarrow n_{21}$	$n_2, n_3, n_4, n_5, n_6, n_8, n_9, n_{11}, n_{12}, n_{14}, n_{15}, n_{16}, n_{36}, n_{37}, n_{38}, n_{40}, n_{41}, n_{43}, n_{44}, n_{46}, n_{47}, n_{48}, n_{49}, n_{51}, n_{52}, n_{53}, n_{54}, n_{56}, n_{57}, n_{59}, n_{60}, n_{61}, n_{22}, n_{23}, n_{24}, n_{25}$

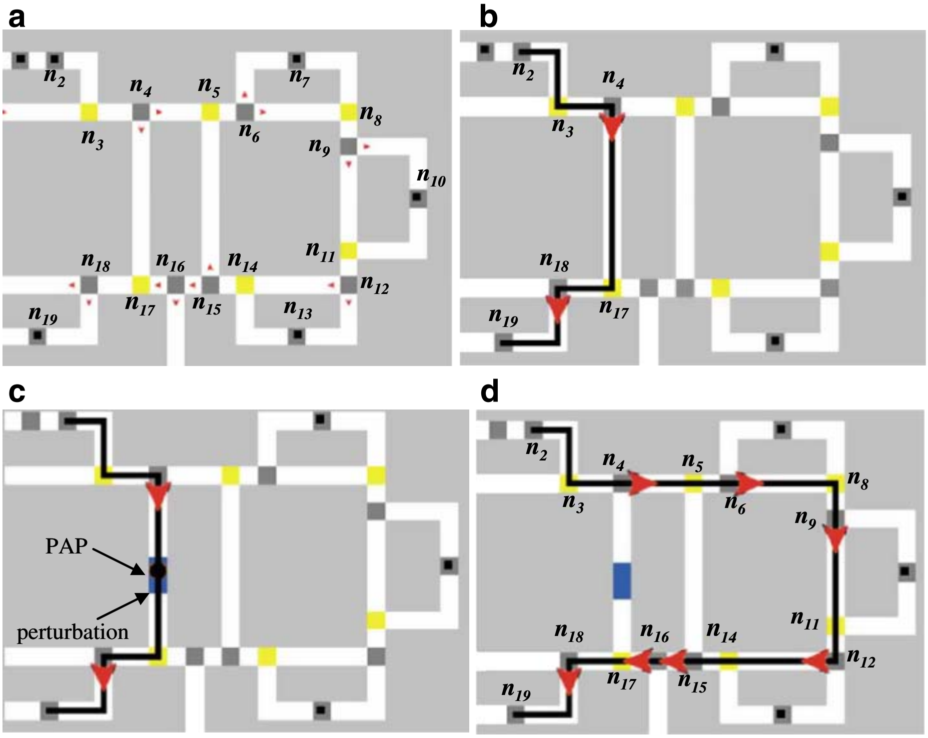


Fig. 11 **a**, **b** Initial state; "best" path found. **c**, **d** Perturbation triggered; alternative path

Figure 11c indicates the presence of a perturbation on the arc $n_4 \rightarrow n_{17}$, affecting traffic fluidity on this arc. (The flow reduction is shown in dark grey).

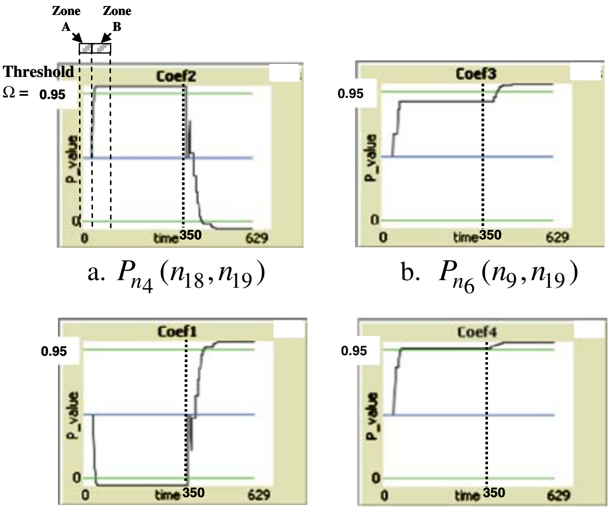
When a PAP (black disc) takes a longer time on this trajectory than expected, a new VAP exploration is triggered via n_4.

The VAPs travelling on the arc $n_4 \rightarrow n_{17}$ perform poorly, thus the appeal of this path decreases. The decreased appeal of this perturbed path consequently increases the appeal of an alternative path $[n_2,n_3,n_4,n_5,n_6,n_8,n_9,n_{11},n_{12},n_{14},n_{15},n_{16},n_{17},n_{18},n_{19}]$ (Fig. 11d), even though this path was originally not the best. This dynamic response is a classic display of the natural routing reconfiguration capacity of the stigmergic approach.

8.3.2 Quantitative Results

Figure 12 shows the evolution of some of the P_{n_i} coefficients. Please note the change that occurs after the perturbation at date $d = 350$ (in simulated time). Figure 13 shows more precisely the evolution of the coefficient Pn_4 (n_{18},n_{19}) for a perturbation occurring at $d = 902$ and for its resolution at $d = 1472$. After the perturbation has been resolved, the coefficient $\mathrm{P}n_4$ (n_{18},n_{19}) again converges to 1. Zones A (transient period) and B (check period) are used to identify the "best path".

Fig. 12 **a** $P_{n_4}(n_{18}, n_{19})$. **b** $P_{n_6}(n_9, n_{19})$. **c** $P_{n_4}(n_6, n_{19})$. **d** $P_{n_9}(n_{12}, n_{19})$



a. $P_{n_4}(n_{18}, n_{19})$ b. $P_{n_6}(n_9, n_{19})$

c. $P_{n_4}(n_6, n_{19})$ d. $P_{n_9}(n_{12}, n_{19})$

*Scenario 2: In the second scenario, a perturbation is introduced on arc $n_{16} \rightarrow n_{36}$. This arc is an obligatory bridge to travel from the upper to the lower section of the FMS and it can't be by-passed. Please note that, after the perturbation, that the alternative path found is the same as the initial path (cf. Table 1). This is possible because the minimum reference time $\tau^*_{n_{10}}(n_{39})$ was constructed using a mobile time window, built with the 20 last measured times. In this context, the previous "good" performance of $\tau^*_{n_{10}}(n_{39})$ is forgotten and the new longer time becomes the reference time.

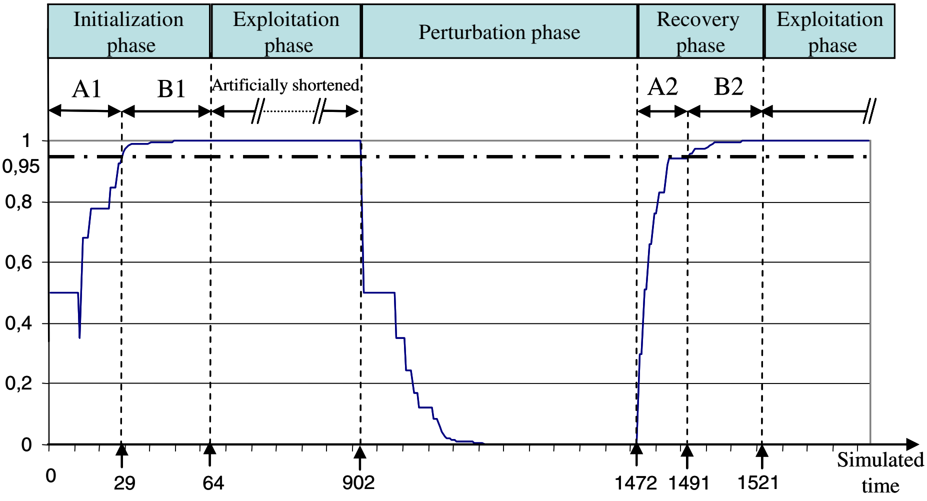


Fig. 13 Evolution of $P_{n_4}(n_{18}, n_{19})$

*Scenario 3: Nodes n_2 and n_{25} are the source and destination nodes, respectively. In the event of a perturbation on the arc $n_{20} \rightarrow n_{21}$, the initial simple path becomes:

$$[n_2, n_3, n_4, n_5, n_6, n_8, n_9, n_{11}, n_{12}, n_{14}, n_{15}, n_{16}, n_{36}, n_{37}, n_{38}, n_{40}, n_{41}, n_{43}, n_{44}, n_{46}, n_{47}, n_{48}, n_{49}, n_{51}, n_{53}, n_{54}, n_{56}, n_{57}, n_{59}, n_{60}, n_{61}, n_{22}, n_{23}, n_{24}, n_{25}]\,.$$

9 Elements for a Real Implementation

Given the promising results of the simulation, we decided to test our model on a real implementation, respecting the assumptions introduced in Section 2. The flexible assembly cell of the Valenciennes AIP-PRIMECA center is being used as experimental support (Figs. 14 and 18).

Three types of equipment were used:

- *A conveyer system* (Fig. 14) Based on Montrac technology [28], this system is a monorail transport system using self-propelled shuttles (Fig. 15) to transport products/materials on tracks;
- *Node instrumentation* (Fig. 16):
  - ◦ a “GC: gate controller” which works to oversee the transfer gate and to help avoid collisions;
  - ◦ a “PUC: positioning unit controller”, on node with resource only, which prepares the product (e.g. unloading/loading) for the resource;
  - ◦ a DS containing P_{nc}, t_{nc}, the two variables required by the PL;
  - ◦ two data communication systems (*Ethernet* for node-to-node interaction, and *IrDA* for node-to-shuttle communication (esb-101 Clarinet system [9]); and
  - ◦ a data processing system, supported by a Wago 750-841 controller [49], which is shared by the VL and PL (if needed one Wago implements the gate controlling).

Fig. 14 The Aip-Primeca FMS conveyer

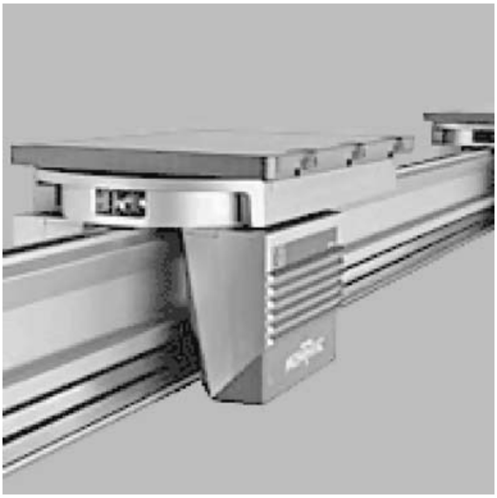


Fig. 15 Shuttle on monorail

- *Shuttle instrumentation* This instrumentation is based on a data processing system embedded in a PDA (personal digital assistant) device (including an IrDA communication device, and the operating system Windows mobile 2005; Fig. 17).

Infrared technology was chosen because of:

- its low energy consumption for a good transfer rate (more than 100 Kb/s),
- its short range, which enables geographical node localization, and
- its light-based physical layer, which is naturally robust to electromagnetic noise.

We are progressing step-by-step in this implementation. The first step is essentially focused on left-hand side of the Aip-Primeca FMS. The right-hand side is only used to loop shuttles from Input to Output (Fig. 18). Active resources are located on the nodes n_2, n_{19}, n_{25} and n_{28}.

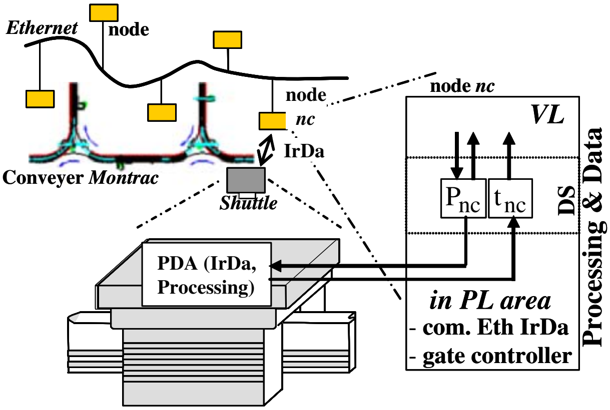
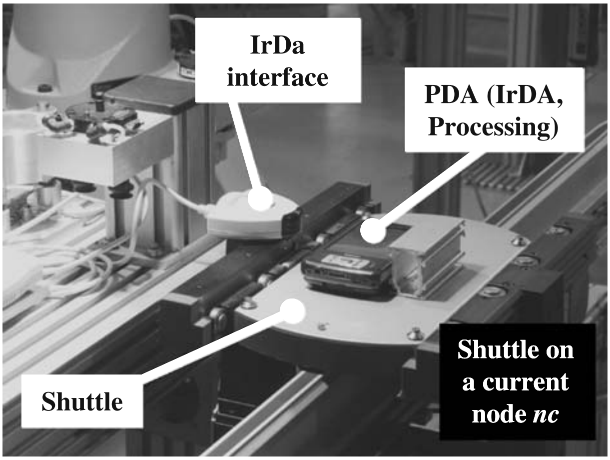


Fig. 16 Node instrumentation

Fig. 17 Shuttle instrumentation



Our stigmergic control model is semi-heterarchical, giving us the choice of implementing it at any point between the two extreme solutions:

- S1: one data processing system for all nodes,
- S2: one data processing system for each node.

The data processing system is a Wago 750-841, which supports programs, tasks and data (VL, DS). We chose an intermediary solution (very close to S2) in which sometimes several nodes are supported by one data processing system. For example,

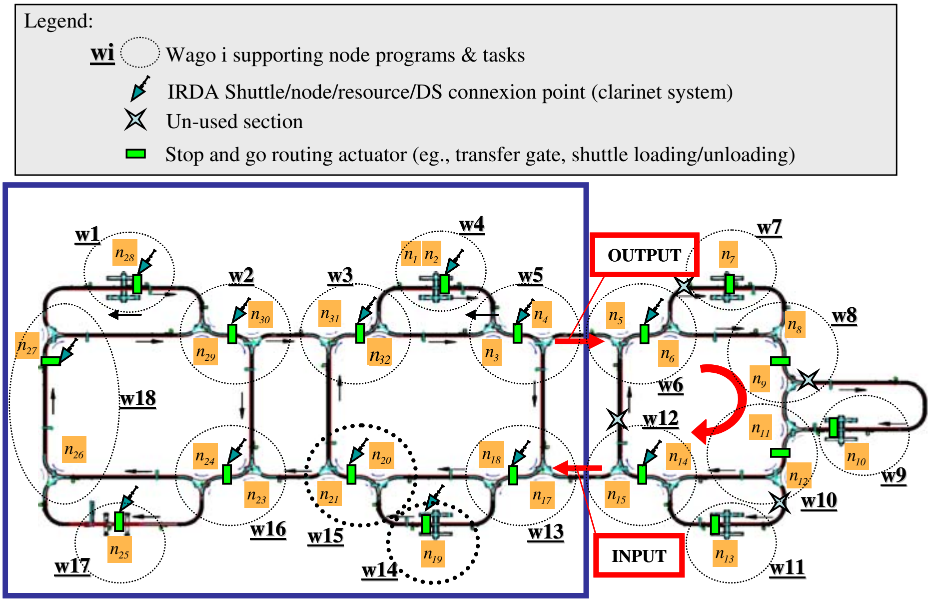


Fig. 18 Functional view of the AIP-Primeca cell case study

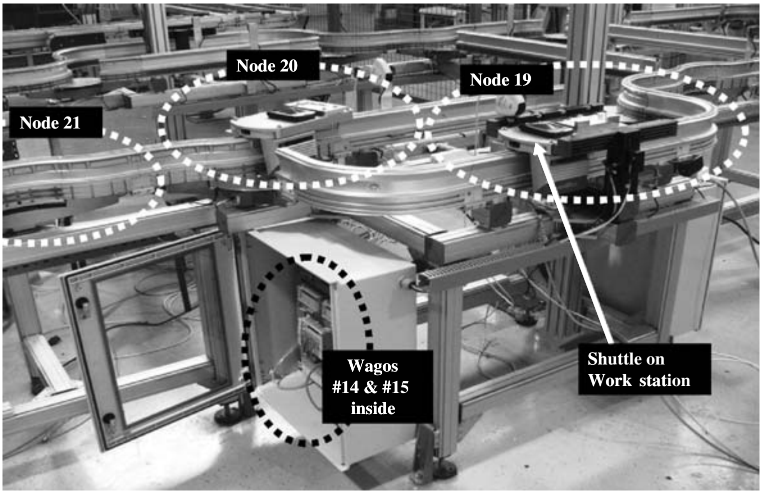


Fig. 19 Real view of nodes 19, 20 and 21 controlled by Wagos no. 14 and no. 15

Wago 14 supports only node 19 (workstation node), and Wago 15 supports the convergent node 20 and the divergent node 21, where routing decision-making is needed. Figures 18 and 19 show the areas controlled by these two Wagos.

The communication implementation is based on well developed technology of Ethernet industrial networks (ModbusTCP, Ethernet/Industrial Protocol). Figure 20 provides the details of the communication architecture and data processing for Wago no. 14 and no. 15.

The communications between node and product were successfully validated. The routing control is now operational, with static pheromones on the nodes, and is supported by Wago controllers. We are continuing to develop Wago programs, and the next steps involve pheromone updating on the virtual level.

At the same time as the deployments and development endeavors described above, we have been working on implementing the supervision system, called Hypervision, based on integrating and blending real and virtual data.

Indeed, usual supervision systems focus on the representation of reality (for example, described in terms of performance indicators) to a human operator. The problem here is that several decisions are taken in the VL and a large amount of information is exchanged at this level. It would therefore be interesting to couple the supervision of both levels: physical and virtual. For example, it would be useful for the human supervisor to supervise not only the real movements of AP and states of AR but also the movements of VAP and virtual active resources and the results obtained at this level. As a consequence, this new Hypervision supervision system should allow the simultaneous supervision of reality and "virtuality". In this context, approaches and models from "mixed reality" [27] could be used to visualize indicators/data concerning nodes, resource, transfer gate, VL, DS and PL.

The Hypervision system is a video system with static cameras positioned to look at the nodes and workstations and moveable and/or rotating cameras that not only provide a global image of the conveyer, but are also capable of zooming in on some

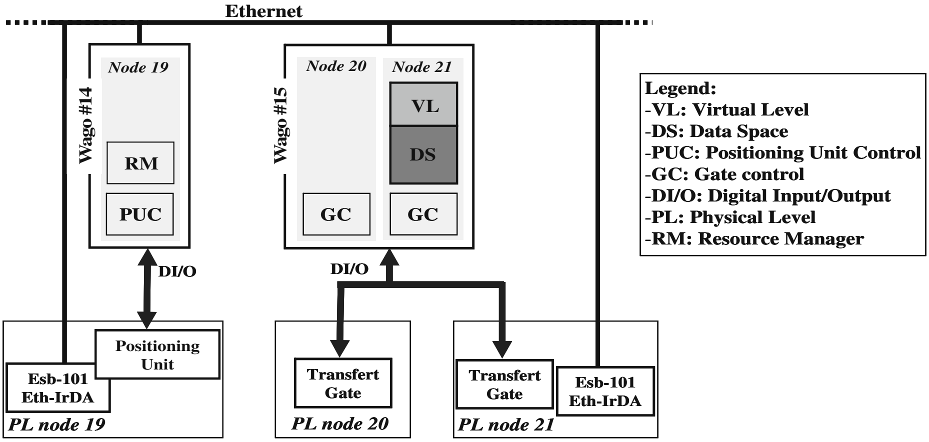
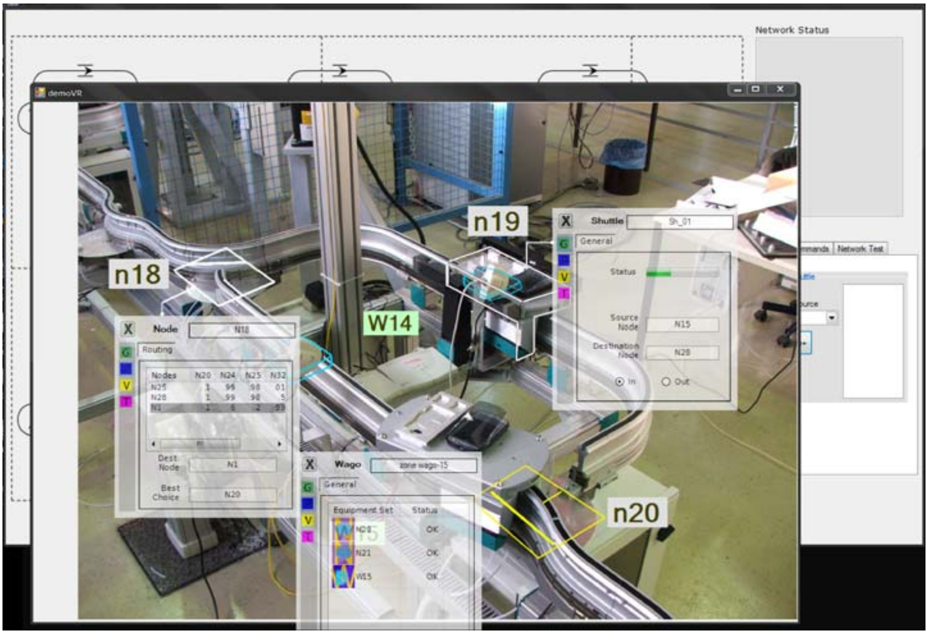


Fig. 20 Communication architecture and data processing for Wagos no. 14 and no. 15

parts and tracking the shuttle, if necessary. The images from the video streams serve as reference for representing operating data, and they support the interaction with the human operator. Since each object detected on the image is a data provider and a potential communicating entity, the human operator can interact with them using the procedures and protocols that we are developing. Figure 21 provides an illustration of this system.



Node: N18

Nodes	N20	N24	N25	N32
N25	1	96	98	01
N28	1	99	98	5
N1	1	6	2	83

Dest Node: N1

Best Choice: N20

Shuttle: Sh_01

Status

Source Node: N15

Destination Node: N28

⦿ In ○ Out

Wago: zone wago-15

Equipment Set	Status
N20	OK
N21	OK
W15	OK

Fig. 21 Screenshot of the Hypervision system

10 Conclusions and Prospects

The results described above illustrate the advantages of our approach. First of all, the PAPs are able to determine the best path from the departure node to the destination node without any centralized control. Secondly, they are able to surmount perturbations by seeking out new paths that bypass the perturbation but still lead to the desired destination.

Other simulations currently under study but not described in this paper highlight the fact that the alternative path found converges towards the optimal path.

The main advantage of our approach is that none of the assumptions on which the system is based relies on the usual simplifications, such as unlimited stock capacity, no jamming, unlimited reliability of the transportation system, and so on. Our proposal should lead to more realistic and deployable real-time control systems.

The main perspectives for future research concern three aspects:

- The first one is to study the different ways to link our DRP model with the DAP distributed control system introduced in part 4.
- The second one is to study the potential benefits of the holonic approach. Indeed, our proposed initial concept of node is defined by a physical level coupled with a virtual level, managing both information and raw material. We can see that from the holonic point of view, a node is a holon. It is interesting to note that our approach was not originally holon-oriented, but now converges towards the holonic approach. A complete re-formulation of our model within the HMS (holonic manufacturing system) concept will help us to identify the potential benefits of an holon-oriented modeling approach compared to our initial approach.
- The last one concerns the supervision of the whole system, VL and PL. We consider the new “hypervision” concept one of our major challenging and innovative short-term perspectives. The difficulty is defining, for a real-time context, a generic model that is able to support several description levels and multiple connectivities, thus specifying the relationship between the real world and the virtual world.

References

1. Aissani, N., Trentesaux, D., Beldjilali, B.: Use of machine learning for continuous improvement of the real time manufacturing control system performances. Int. J. Ind. Syst. Eng. **3**(4), 474–497 (2008)
2. Arkin, R.C.: Behaviour-based robotics. The MIT Press, Cambridge, MA (1998)
3. Baker, A.D.: A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: dispatching, scheduling, and pull. J. Manuf. Syst. **17**(4), 297–320 (1998)
4. Baïna, S., Morel, G.: Product centric holons for synchronisation and interoperability in manufacturing environments. In: IFAC Conference on Information Control Problems in Manufacturing 2006, pp. 543–548 (2006)
5. Bousbia, S., Anli, A., Trentesaux, D., Grislin, E.: Agile scheduling of flexible manufacturing systems of production. In: 17th World Congress—IMACS 05, CD-Rom, paper no. T6-R-00-0316, Paris, July 2005, 8 p. (2005)
6. Brückner, S.: Return from the Ant Synthetic Ecosystems for Manufacturing Control. Thesis Humboldt-University of Berlin (2000)
7. Cavalieri, S., Garetti, M., Macchi, M., Taisch, M.: An experimental benchmarking of two multi-agent architectures for production scheduling and control. Comput. Ind. **43**, 139–152 (2000)

8. Chen, R., Lu, K., Chang, C.: Application of the multi-agent approach in just-in-time production control system. Int. J. Comput. Appl. Technol. **17**(2), 90–100 (2003)
9. Clarinet System: Network connectivity for mobile devices. http://www.clarinetsys.com (2007)
10. Deneubourg, J.L., Pasteels, J.M., Verhaeghe, J.C.: Probabilistic behaviour in ants: a strategy of errors? J. Theor. Biol. **105**, 259–271 (1983)
11. Dorigo, M., Colombetti, M.: Robot Shaping: An Experiment in Behaviour Engineering. The MIT Press, Cambridge, MA (1998)
12. Dorigo, M., Di Caro, G., Gambardella, L.M.: Ant algorithms for discrete optimization. Artif. Life **5**(2), 137–172 (1999)
13. Dorigo, M., Stützle, T.: Ant Colony Optimization. The MIT Press, Cambridge, MA (2004)
14. Duffie, N.A., Prabhu, V.: Heterarchical control of highly distributed manufacturing systems. Int. J. Comput. Integr. Manuf. **9**(4), 270–281 (1996)
15. Ferber, J.: Les Systèmes Multi-agents. InterEditions (1995)
16. Gou, L., Luh, P.B., Kyoya, Y.: Holonic manufacturing scheduling: architecture, cooperation mechanism, and implementation. Comput. Ind. **37**, 213–231 (1998)
17. Grassé, P.P.: La reconstruction du nid et les coordination inter-individuelles chez Bellicositermes natalensis et Cubiter-mes sp. La théorie de la stigmergie: essai d'interprétation du comportement des termites constructeurs. Insectes Soc. **6**, 41–83 (1959)
18. Hadeli, T., Valckenaers, P., Kollingbaum, M., Van Brussel, H.: Multi-agent coordination and control using stigmergy. Comput. Ind. **53**, 75–96 (2004)
19. Hsieh, F.-S.: Holarchy formation and optimization in holonic manufacturing systems with contract net. Automatica **44**, 959–970 (2008)
20. Koestler, A.: The Ghost in the Machine. Hutchinson, London (1967)
21. Kurabayashi, D.: Development of an intelligent data carrier (IDC) system and its applications. In: Proceedings of 4th International Symposium on Artificial Life and Robotics, pp. 34–39 (1999)
22. Lastra, J.L.M., Colombo, A.W.: Engineering framework for agent-based manufacturing control. Eng Appl Artif Intell. **19**, 625–640 (2006)
23. Leitao, P., Restivo, F.: ADACOR: a holonic architecture for agile and adaptive manufacturing control. Comput. Ind. **57**, 121–130 (2006)
24. Mascada WP4 Report: ACA (autonomous co-operating agents) framework for manufacturing control systems (1999)
25. McCulloch, W.S.: Heterarchy of values determined by the topology of nervous nets. Bull. Math. Biophys. **7**, 89–93 (1945)
26. McFarlane, D., Sarma, S., Chirn, J.L., Ashton, K.: The intelligent product in manufacturing control and management. In: 15th Triennal IFAC World Congress. Barcelone, Spain (2002)
27. Milgram, P., Kishino, F.: A taxonomy of mixed reality visual displays—IEICE Transactions on Information and Systems—Special issue on Networked Reality (1994)
28. Montech Technology. Conveyor systems. http://www.montech.com (2007)
29. Maione, G., Naso, D.: A soft computing approach for task contracting in multi-agent manufacturing control. Comput. Ind. **52**, 199–219 (2003)
30. Okino, N.: Bionic manufacturing system. In: Peklenik, J. (ed.) Flexible Manufacturing System: Past–Present–Future, pp. 73–95. CIRP, Paris (1993)
31. Ouelhadj, D., Petrovic, S., Cowling, P., Meisels, A.: Inter-agent cooperation and communication for agent-based robust dynamic scheduling in steel production. Adv. Eng. Inf. **18**(3), 161–172 (2005)
32. Parunak, H.V.D., Brueckner, S., Sauter, J.: ERIM's approach to fine-grained agents. In: Proceedings of the NASA/JPL Workshop on Radical Agent Concepts (WRAC'2001). Greenbelt, MD (2001)
33. Payton, D., Daily, M., Estkowski, R., Howard, M., Lee, C.: Pheromone robotics. In: Autonomous Robots, vol. 11, no. 3, pp. 319–324. Kluwer, Norwell, MA (2001)
34. Peeters, P., Van Brussel, H., Valckenaers, P., Wyns, J., Bongaerts, L., Heikkilä, T., Kollingbaum, M.: Pheromone based emergent shop floor control system for flexible flow shops. In: Proceedings of International Workshop IWES'99, Kobe, Japan, Dec. 6–7 (1999)
35. Reaidy, J., Massotte, P., Diep, D.: Comparison of negotiation protocols in dynamic agent-based manufacturing systems. Int. J. Prod. Econ. **99**, 117–130 (2006)
36. Russell, R.A.: Laying and sensing odor markings as a strategy for assisting mobile robot navigation tasks. IEEE Robot. Autom. Mag. **2**, 3–9 (1995)
37. Sallez, Y., Trentesaux, D., Berger, T., Tahon, C.: Product-based and resource-based heterarchical approaches for dynamic FMS scheduling. Comput. Ind. Eng. **46**, 611–623 (2004)

38. Sousa, P., Ramos, C.: A distributed architecture and negotiation protocol for scheduling in manufacturing systems. Comput. Ind. **38**, 103–113 (1999)
39. Smith, R.G.: The contract net protocol: high level communication and control in a distributed problem solver. IEEE Trans. Comput. **C29**(12), 1104–1113 (1980)
40. Tharumarajah, A.: Survey of resource allocation methods for distributed manufacturing systems. Prod. Plan. Control **12**(1), 58–68 (2001)
41. Theraulaz, G., Bonabeau, E.: A brief history of stigmergy. J. Artif. Life **5**(2), 97–116 (1999)
42. Trentesaux, D., Dindeleux, R., Tahon, C.: A multicriteria decision support system for dynamic task allocation in a distributed production activity control structure. Int. J. Comput. Integr. Manuf. **11**(1), 3–17 (1998a)
43. Trentesaux, D., Tahon, C., Ladet, P.: Hybrid production control approach for J.I.T. scheduling. Artif. Intell. Eng. **12**, 49–67 (1998b)
44. Ulieru, M., Norrie, D.: Fault recovery in distributed manufacturing systems by emergent holonic re-configuration: a fuzzy multi-agent modeling approach. Inf. Sci. **127**, 101–123 (2000)
45. Valckenaers, P., Hadeli, T., Saint Germain, B., Verstraete, P., Van Brussel, H.: Emergent short-term forecasting through ant colony engineering in coordination and control systems. Adv. Eng. Inform. **20**(3), 261–278 (2006)
46. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, L.: Reference architecture for holonic manufacturing systems: PROSA. Comput. Ind. **37**(3), 255–274 (1998)
47. Verstraete, P., Valckenaers, P., Van Brussel, H., Saint Germain, B., Hadeli, K., Van Belle, J.: Towards robust and efficient planning execution. Eng. Appl. Artif. Intell. **21**(3), 304–314 (2008)
48. Wagner, I.A., Lindenbaum, M., Bruckstein, A.M.: Distributed covering by ant-robots using evaporating traces. IEEE Trans. Robot. Autom. **15**(5), 918–933 (1995)
49. Wago system, innovative connections. http://www.wago.com 2007
50. White, T.: Expert assessment of stigmergy: a report for the Department of National Defence, Carleton Univ Ottawa (2005)
51. Wilensky, U.: Center for Connected Learning and Computer-Based Modeling. Northwestern University. Evanston, IL (1999). http://ccl.northwestern.edu/netlogo/
52. Wong, T.N., Leung, C.W., Mak, K.L., Fung, R.Y.K.: Dynamic shopfloor scheduling in multi-agent manufacturing systems. Expert Syst. Appl. **31**, 486–494 (2006)
53. Wyns, J.: Reference architecture for holonic manufacturing systems—the key to support evolution and reconfiguration. PhD thesis K.U. Leuven (1999)