Article

Bio-Inspired Manufacturing Multi-Agent System for Control and Coordination of AGV Systems

Branislav Micieta, Lukas Durica and Vladimira Binasova *

Department of Industrial Engineering, Faculty of Mechanical Engineering, University of Zilina, Univerzitna 1, 010 26 Zilina, Slovakia; branislav.micieta@fstroj.uniza.sk (B.M.); lukas durica@fstroj.uniza.sk (L.D.)
* Correspondence: vladimira.binasova@fstroj.uniza.sk; Tel.: +421-041-513-2731

Abstract: In this paper, modelling, simulation and verification of multi-agent manufacturing system with application of bio-inspired techniques are addressed. To this end, the new solution of abstract architecture for control and coordination decentralized systems - CODESA is suggested. Centralized architecture suffers from various problems, such as rigidity, scalability, low fault-tolerance or very limited flexibility, agility, energy efficiency and productivity. Prime is concrete application of CODESA in manufacturing domain. The undesirable characteristics of emergent behaviour are the problem to achieve optimization and impossibility to predict future states of the system. CODESA-Prime has been tested by simulations for automatic guided vehicle (AGV) systems guided by magnetic tape in Ella Software Platform.

Keywords: industrial automation; autonomous design; multi-agent systems; industry 4.0; biologically inspired techniques; AGV systems; energy efficiency

1. Introduction

The international stance makes available such abstractions and know-how of agents, and the communications that place among them. These turn out to be important scientific abstractions for multi-agent systems. These abstractions no doubt have much conceptual appeal. Furthermore, there are simple pragmatic and technical reasons for considering them seriously. They are natural to humans, who are not only the designer and analysers of multi-agent systems, but also the end users and requirements specifiers and provide succinct description of, and help understand and explain, the behaviour of complex systems.

Multi-agent systems (MAS) have been applied to areas such as control, production planning, scheduling, resource allocation, vehicle routing, etc. The advantage of such heterarchical architecture is the absence of superior unit, which in practice can be a single point of failure in the system. Individual agents communicate directly with each other. This allows scalability (no need to shut down production system in order to add or remove resources), increases flexibility, modularity and agility.

The multi-agent system itself provides high fault-tolerance. A simple implementation of the control software ensures low maintenance needs and transparency of the source code. The main disadvantage of heterarchical architecture is the phenomenon [1] coming from emergent behaviour. The undesirable characteristics of emergence are, for example, the problem to achieve optimization and impossibility to predict future states of the system.

The current mass customization era requires increased flexibility and agility in the manufacturing systems to adapt changes in manufacturing requirements and environments. An ontology-based reconfiguration agent attempts to reconfigure the manufacturing system after realising changes in the requirements or the manufacturing environment. The benefit of approach is minimizing the overheads of the reconfiguration process by achieving rapid reconfiguration with minimum human intervention [2]. The agent uses ontological knowledge of the manufacturing environment for the purpose of reconfiguration without human intervention. The current mass customization era requires increased flexibility and agility in the manufacturing systems to adapt changes in manufacturing requirements and environments. Ontologies play important role in knowledge sharing and exploration, particularly in communication in multi-agent systems.

Sustainable production can be defined as production and use of products and services in a manner that is socially beneficial, economically viable and environmentally benign over whole product life cycle [3].

The globalized economy is strongly influenced not only by economic cycles but also a rapid change in customer behaviour, which result in turbulences [4].

In the last two decades, the attention of experts is focused especially to a holonic and multi-agent manufacturing system. The solution seems to be used the semi-heterarchical architecture, introducing hierarchies into the heterarchies [5], propose a semi-heterarchical architecture composed by a supervisor (S) and subordinate decisional entities (E). In this approach, simulation-optimization mechanisms are used to reduce the subordinates' myopic behaviour [6]. Semi-heterarchical architectures can merge the benefits of hierarchy in terms of global performance and heterarchy in terms of reactivity, adaptability and fault tolerance. Heterarchical control architectures are essentially founded on cooperation and full local autonomy, resulting in high reactivity, no master/slave relationships and local information retention [7].

The current semi-hierarchical approaches propose structural changes, which are closer to the hierarchy as heterarchy. However, heterarchical architecture is not completely lost; the disadvantages of hierarchy overshadow the advantages of heterarchical principles [8]. Although heterarchical architectures have many advantages, they were rarely implemented due to the weaknesses [9]. A possible solution to this problem is implementation of tools that allow adjusting the degree of autonomy of subordinate control units to the architecture, also controlling the temporal and social myopia and creating of dynamic semi-heterarchical control architecture.

Swarm Intelligence [10] can be used to eliminate some of the previously mentioned negative characteristics (derived from heterarchical architectures, multi-agent systems, lack of adaptability, etc.), using indirect pheromone-based communication. The global market are imposing strong changing conditions for companies running their businesses, something comprising complex and large scale systems [11,12]. Agent systems were introduced in automation and manufacturing when there was the vision of having distributable software components to resolve local tasks, comparable to the heterogeneous nature of the physical domain. Existing models of production, such as Flexible Manufacturing Systems (FMS) [13], Reconfigurable Manufacturing Systems (RMS) [14, 15] is a basis for the application of multi-agent systems to solve and optimize the problems of production, due to the increase of introduced complexity. Among the several publications reporting the application of MAS in automation and production systems, the reader can consult the reference [16] to retrieve more information on this topic [17].

The benefits of agent-based industrial systems are robustness, scalability, reconfigurability and productivity, which are transformed to a greater competitive advantage.

Multi-agents systems is pointed out as a suitable approach to address this challenge by offering an alternative way to design control systems, based on the decentralization of control functions over distributed autonomous and cooperative entities. However, in spite of their enormous potential, they usually lack some aspects related to interoperability, optimization in decentralized structures and truly self-adaptation [18].

Product-driven control may enable manufacturing companies to meet business demands more quickly and effectively. But a key point in making this concept acceptable by industry is to provide benchmarking environments in order to compare and analyse their efficiency on emulated large-scale industry-led case studies with regard to current technologies and approaches [19].

Lepuschitz [20] proposed the overview of the ontologies and their suitability for process automation domain and subdomains. Leitao and Restivo [18] presents a holonic approach to manufacturing scheduling, where the scheduling functions are distributed by several entities, combining their calculation power and local optimization capability. In Hsieh [21], the author defined a holarchy formation problem to lay a foundation to propose models and develop collaborative algorithms to guide the holons to form a holarchy that coherently moves toward the desired goal state with minimal costs.

Industrial requirements have clearly evolved from the traditional performance criteria, described in terms of static optimality or near-optimality, towards new performance criteria, described in terms of reactivity, adaptability and visibility. A growing number of industrialists now want control systems that provide satisfactory, adaptable and robust solutions rather than optimal solutions that require several hard assumptions to be met [22, 23].

2. Materials and method - The characteristic of CODESA-Prime

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation as well as the experimental conclusions that can be drawn. This section will briefly characterize the various technologies used in CODESA. CODESA is the reference technological model, suitable for directing and coordinating of decentralized systems. The application of this technological structure is possible in domains where control of non-linear, heterarchical, large-scale systems in dynamic, heterogeneous and unpredictable environment is needed. The main characteristics of CODESA are [24]:

- fault tolerance (MAS, Swarm Intelligence, etc.),
- high level of modularity and reconfigurability (MAS, Service-oriented Architecture - SOA, Internet of Things - IoT),
- enhanced interoperability (Domain ontology, SOA),
- fast processing and storing large amounts of data (CLOUD, IoT),
- autonomous and intelligent behaviour (MAS, Swarm Intelligence),
- the ability to predict future conditions, and proactively react (Swarm Intelligence),
- adaptation of system and techniques of Bionic layer (Evolutionary Computing),
- use of dual communication: (direct - modified CNP protocol and indirect - based on pheromones in a virtual environment (Software Platform, Swarm Intelligence), etc.

2.1. Proposal of logistics system CODESA-Prime

CODESA-Prime is a concrete application of framework CODESA in the manufacturing domain. Using technological structure designed for production management application domain CODESA-Prime, it is possible to manage manufacturing processes of planning, scheduling, routing product, mobile agents routing, and inventory management through optimization and prediction system. The main part of CODESA-Prime is a multi-agent system. CODESA-Prime features of holonic terms are divided into five control levels:

1. The first control level - looking for free capacities of production and logistics resources through intelligent agent’s orders.
2. The second control level - the routing of intelligent product, agent across production resources.
3. The third control level - the routing of intelligent product agent across the logistics resources [24].
4. The fourth control level - the control and coordination systems through AGV path.
5. The fifth control level - the control model, graphical representation, and physical models.

This paper is focused on the fourth and the fifth control level.

2.2. *The fourth control level*

The fourth holonic level performs function of control, coordination of holonic agent AGV searching and booking of paths to the target point. This level includes the following entities:

- meta-scheme of AGV path,
- AGV path,
- node,
- segment,
- Intelligent Segment Agent - ISA,
- logical segment,
- critical logical segment,
- avatar representing agent of AGV,
- ant-agents of exploration colony,
- ant-agents of information colony,
- ant-agents of reservation colony,
- pheromone container.

2.1.1. Meta-scheme of AGV path and AGV path

One of the fundamental aspects of this level is the meta-scheme of AGV tracks. It is a reflection of the model paths from the fifth control level, represented by an oriented and evaluated graph that contains vertices (nodes) and edges (segments). Into meta-scheme can be inserted digital pheromones. The meta-scheme updates their intensity value, but also can modify their content. AGV path is a part of meta-scheme composed of several nodes and segments. The concept of AGV path is typically used as an indication of the direction of individual entities through nodes and segments.

2.1.2. Node

Nodes in the meta-scheme may be located at crossroads, but also outside them, while they distribute part of AGV path to the individual segments. The node is the inactive entity. It does not have decision-making capacity; therefore it is not represented by agents. The crossroad is the type of to which they are connected more than two segments. Each node has a unique global identifier in AGV path, which is a non-negative integer. Each node is defined by the position - the point in the virtual 3D space. The nodes also include a list of all segments that are attached to them (i.e. they contain initial or end coordinates (position data) with the position of the node in space). The nodes comprise the pheromone container [24].

2.1.3. Segment

A segment is defined by two nodes in the virtual space. It has a unique ID (like a node) that is non-negative number. The segment can be extended by assigning a weight.

- directed both directions,
- directed from its beginning to its end,
- directed from its end to its beginning,
- undirected (i.e. it is not possible to navigate in any direction, generally performs a different function).

In the multi-agent system is a segment or group of segments represented by an intelligent segment agent (ISA).

2.1.4. Intelligent segment agent – ISA

Intelligent segment agent (ISA) offers a service of running over his physical part (i. e. its model). Unlike the node, ISA can communicate with other agents (through direct or indirect communication) in order to authorized crossing. ISA owns AGV crossings schedule. In the case of indirect communication, this time is represented by pheromone container, which are stored reservation pheromones to allocate certain time slot (time slot) of AGV crossing.

2.1.5. Logical segment

Logical segment is composed of more than one segment. Logical segment creates an intelligent segment agent and it shall not be less than the size of holonic agent AGV (including safety sensor). This means that the AGV can occupy two segments at a time, but in no case three segments. Logical segment is mostly generated, if it follows consecutively more segments without crossroad. It is also used in areas where the AGV stops for a purpose. In this case, the logical segment is also created with segments connected to the crossroad.

2.1.6. Critical logical segment

Critical logical segment is a critical type of segment that connects two nodes, and between the nodes there is no other path. These segments are defined by the user in the design of meta-scheme. All critical segments in paths must be reserved in chronological order. Holonic agent AGV found on this type of segment have the maximum priority for the next segment that is critical for the output from the logical segment.

2.1.7. Avatar

Avatar is the representative of AGV agent on this holonic level of control and coordination of direct and indirect communication. Its function is searching and booking of the path to selected destination, while it communicates with the intelligent segment agents. In the case of indirect communication is his task creating ant colonies, setting their properties (number of ants in one colony, the maximum number of passes through nodes - hops, ID colonies, etc.) and informing of agent about their results. Avatar does not have decision-making capacity. Unlike the agent, avatar can relocate at any point on meta-scheme.

This feature is used when the AGV agent already has a path to the desired objectives and the selected one of the avatars it cyclically reserves, while the agent has additional transport requirements. In this case, the agent creates another avatar, which will move to where the holonic agent AGV ends the current path and through ant colonies books next path. Another use case for properties of the reverse is by the collection of information from ant-agents that have completed their life cycle.

One AGV Agent can have multiple avatars. In the event of termination of life cycle of the ant-agent agent (task completed or task failed), the avatar (who stood for the birth of his colonies) is moved to the place where extracted from ant necessary information concerning the path and in addition the pheromone on the success of tasks and possibly and because of its failure to fulfil. The information transmitted to AGV agent, which is also based on them firmly committed to another intention.

2.1.8. Ant colonies

Due to the minimization of computational complexity, ant-agent can in every step only perform one of the following actions:

- omit one step (ant-agent remains idle) (IDLE)
- movement in the desired direction at a defined speed and collection of information about the path which pass through (MOVE_AND_GATHER)

- insert pheromone (LAY_PHEROMONE)
- observe already submitted pheromones (OBSERVE_PHEROMONES)
- modify already created pheromones (MODIFY_PHEROMONE) and
- delete pheromones (WIPE_OFF_PHEROMONE).

Ant-agent may be present in only one state of their life cycle:

1. IDLE,
2. MOVING_AND_GATHER,
3. ARRIVING_AT_NODE,
4. AT_NODE,
5. LEAVING_NODE,
6. ENTERING_SEGMENT,
7. TASK_COMPLETED and
8. TASK_FAILED.

Inserting of pheromone is generally carried out in the state at node or in the state entering segment. If he is unable to complete its task, ant releases pheromone to the environment and with a short report containing justification why could not accomplish the desired task:

1. NO_HOPS_LEFT,
2. IMPASSE,
3. PHEROMONE_NOT_FOUND,
4. RESERVATION_IMPOSSIBLE.

Number of ant-agents in the colony, their service life (number of hops, respectively length of the life cycle), period of their creation is defined by agent that this information transmit to each of avatar. The colonies are cyclically casted, and they travel through meta-scheme with several times higher speed than the actual holonic agent AGV. That means that any disturbances in conditions of the logistics system, the AGV agent has the necessary information in advance, and within reasonable timescales can change their decision. Exploring ant colony (i.e. the information that is gathered and transmitted) is represented agent's beliefs in terms of BDI. Reservation colony represents agent's intentions in terms of BDI.

2.1.9. Exploring ant colony of holonic agent AGV

Exploring ant colony of holonic agent AGV searches the meta-scheme in order to find specific segment or node, which its avatar agent of the holonic agent has selected. Each ant-agent in colony of the holonic agent carries its unique identifier that is composed of the following components:

- unique name of AGV agent,
- a time stamp that is generated at the birth of ant colony,
- unique identification of each individual (ant-agent) in the colony, which is represented by a non-negative integer and
- underscore between the mentioned part of the identifier.

Ant-agents store this identifier to each pheromone that is inserted into the environment. With this mechanism, the uniqueness of individual colonies and ant-agents is achieved. An example of a unique identifier of an ant colony is illustrated in Figure 1. Using this generated mechanism is guaranteed by the uniqueness of the individual in a colony and each colony, and thus also the labelling of driven path. The time stamp consisting of the date (yyyy-mon-dd) and time (hh: mm: ss.microsec). Time stamp granularity is in microseconds.

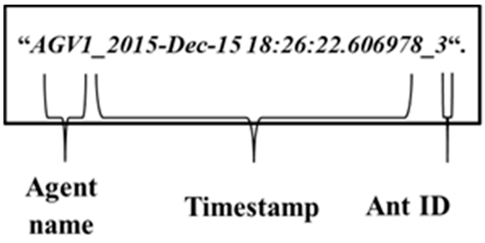

Figure 1. The example of a unique identifier of an individual ant colony and also the pheromone of timestamp.

Exploring ant-agents passing a meta-scheme can choose their way at random, or they may choose another direction (depending on user settings) by observing pheromones of the colony. Information about the path is recorded in each step:

- the total length of the path,
- the total travel time of a path,
- the cost of the path (from the valuation of individual segments),
- the earliest possible departure date of the holonic agent AGV,
- additional custom information.

After passing the node the avatar extracts the pheromone giving information on the success of tasks, pheromone giving information on possible departure schedule, the total distance traveled and time from the ant-agent and the environment as well and forwards it to the avatar representing an intelligent agent segment. With every passing through the node the number of hops decrements. Possible departure schedule pheromone contains reserved slots into which each intelligent agent segment shall pass information on the path by calculating the values of total distance traveled and the holonic agent speed of the ant-agent, thus giving information about the time it takes for the AGV to arrive. This time is then subtracted from the already reserved slots and then is passed to the possible departure schedule. Projecting time reservations of individual segments into the possible departure pheromone is shown in Figure 2. The ant with the updated pheromone goes on to the next segment until it finds the desired destination node, or until his life cycle ends. By sticking to this procedure it is possible to avoid inconveniences that may arise in the event that two holonic agents block each other.

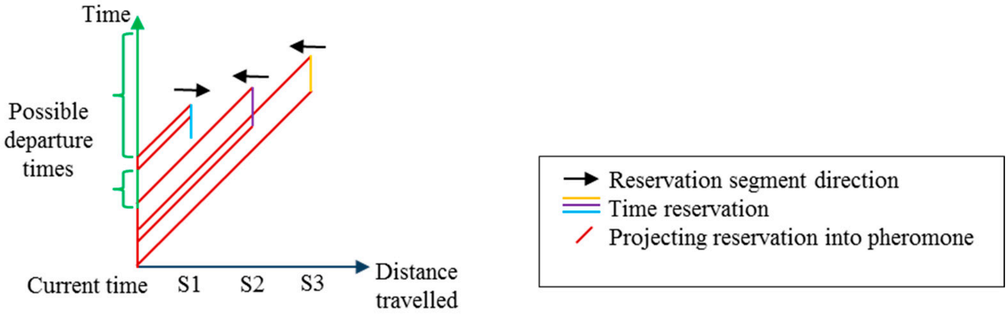

Figure 2. Projecting time reservations of individual segments into the possible departure pheromone.

2.1.10. Information ant colony of holonic agent AGV

After the holonic agent AGV agent decides which path to take based on information provided by the ant colony, the avatar sends out information colonies to inform intelligent segment agents of its decision. Ant colony responsible for conveying information usually consists of a single agent. In the event the segment is already reserved, or another information colony having higher priority has passed through a given section already and reported on passing through the overlapping time interval, the segment refuses reservation. If an intelligent agent segment has already been informed about passing the interval <t-5, t + 10> by the colony A, and then the colony B informed the intelligent segment about holonic agent AGV passing the aforementioned overlapping interval with higher priority, the agent A will find out about this only after trying to reserve the segment without much success. The agent A will begin the process of finding and reserving again. Where the priority is the same, the intelligent segment agent keeps information of possible future reservation without modification. UIA can be used as an element through which the agent may change the order of reservations of the same priority in future. If the ant colony passes through a critical segment, it assumes the maximum priority (i.e. dynamic adjustment of priorities) and applies it to the next segment in the way. In this way, it is possible to avoid blocked holonic agent AGV in the event one of them comes out of the critical segment and cannot change its path. Ant colony conveying information is a modification of the original delegated multi-agent systems. It was designed to bring the option of priority into the system.

2.1.11. Reservation ant colony of holonic agent AGV

The reservation ant colony is created by the agent, as is the case with exploring and information colonies. Ant colonies are usually made up of a single agent that passes through a meta-scheme and reserves the crossing through intelligent segments. In doing so he is guided by a pheromone placed by a member of exploring colony, which indicates a path to the destination node/ segment selected by the holonic agent AGV agent. An ant-agent of the reservation colony collects information about the already travelled path in the form of unique segment identifiers. If the event the lifetime of any ant-agent comes to an end (task_completed, or task_failed), the avatar extracts information about the task from the last pheromone the ant-agent put into the environment. The avatar also extracts information from the ant-agent about any reserved segments, and then he presents information to holonic agent AGV agent. If the path to the destination node is successfully reserved, holonic agent AGV may leave the currently occupied segment. The agent can reserve the travel time as needed, however, to do so he must know the full path (exploration colonies) to the destination node (in order to reserve segments in the direction to the destination node/ segment) if the path consists of the critical segment or if the reservation ends at the node that marks the beginning of the next critical segment and the agent intends to pass through it. In this case, the agent will be forced to inform and reserve the critical segment and the immediately following segment (on the path the agent decides to take).

2.1.12. Pheromone container

For the possibility of insertion of pheromones in the environment have some entities the pheromone container. The container has several features that allow:

- reservation of required segment for a particular time slot,
- searching of road sign by a unique identifier of ant-agent,
- Updating of the departures maps.

2.1.13. Pheromones

Pheromone is a form of indirect communication and is inserted in the environment that can be observed and modified by ant colonies. The fourth control level contains six types of pheromones (Figure 3):

- the pheromone timestamp,
- the path signs pheromone,
- the reservation pheromone,
- the pheromone of the future reservation,
- the possible departure schedule pheromone.

The pheromone timestamp contains a time stamp and it also offers a possibility to leave a short message. Other pheromones sprung out from therefrom – they took over its properties. When a pheromone is inserted into the environment, the current timestamp as well as a unique identifier of the ant-agent who created it are incorporated into it. The path signs pheromone contains in addition to the time stamp also a path sign. The path sign is represented by a non-negative integer, indicating the following entity - a segment or node as needed. The sign can lead to a segment through which the ant-agent plans to go or through which he has already passed, or optionally a node that features an entity (source and so on). As a rule, signs are inserted into segments pointing to nods and vice versa. In addition to the information contained in path sign pheromones the reservation pheromone also contains the reservation date at which it was inserted into the segment. Reservation has three dates:

- date of AGV's arrival to the segment,
- date of AGV's departure from the segment,
- date on which the entire AGV becomes present at the segment.

The pheromone of the future reservation contains the same information as the reservation pheromone; however, it is only informative and non-binding. The pheromone of promotion in addition to carrying a timestamp and path signs pointing to its source also contains brief information about the services that it promotes in – name of the agent providing the service, the name of the service etc. The possible departure schedule pheromone contains the projected intervals at which the AGV can leave the current segment. In the event the time interval that the agent projects into the segment already exists or overlaps with another one, they merge so there are as few pheromones in these intervals as possible. Allocation pheromone does not contain a timestamp and the dates of the start and end of the allocation of the required services or operations. The pheromone is used by ant colony for the third and the fourth levels.

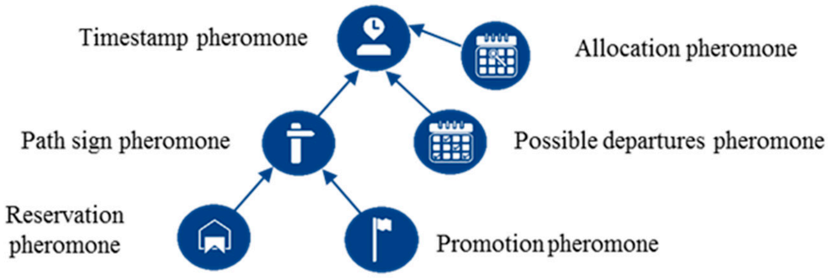

Figure 3. Inheritance in pheromones, descendants acquire properties of their parents.

2.1.14. Other parameters of the fourth management level

Inertial parameter is very important for the fourth holonic level. As we have already mentioned, one of the advantages of indirect communication based on pheromones and their evaporation and renewal is a quick adaptation of the system to changes in production/ logistics processes. However, this adaptation may be too fast given the already allocated or scheduled tasks and consequently may degrade the overall system performance. Inertial parameter indicates how much the price of the new path must go down as opposed to the price of the current path. The evaporation rate of pheromone is closely related to a period of ant colony founding. If the pheromone evaporation rate is higher than the rate of ant colonies founding, it may give rise to a situation where the reservation pheromone evaporates even before it is restored (reservation pheromone). This can cause problems if another holonic agent AGV reserves this path in a given time interval. Ultimately, the holonic agent AGV would be forced to choose a new path. However, it is undesirable in certain situations (in case there is no alternative path and holonic agent AGV are moving towards each other). However, if the pheromone evaporation rate is slow, it could bring the resulting effect onto the whole system and cause its rigidity. In the event there is a rush order in the production, it is necessary that it passes along the whole production as soon as possible. Therefore,

doi:10.20944/preprints201609.0003.v1

the priority parameter has been introduced. Higher priority decides what holonic agent AGV takes precedence. This means that if the information colonies with different priorities inform the segment agent about the future reservation, the agent gives priority to the colony with higher priority.

3. Logic verification of the various scenarios for the fourth holonic level

Exploration colony decides at the nod what path to take (if possible) next. This decision can be random (generated by pseudo-random algorithm), or based on information in pheromones. In this case, certain parts of paths or entire paths of individual ant-agents are completely identical, failing to fully utilize the potential of exploration colony. In this case, it is necessary to have more colonies. It is possible for an ant to make its decision based on information that another ant-agent from its colony already took the given path (segment). Therefore, the ant-agent may choose another free path. By doing so we would increase the explored area of the meta-scheme. However, we shall bear in mind that this would probably reduce the likelihood of achieving the optimum result. Another possibility is cloning agents to cover the entire meta-scheme. That is, the ant at the node would clone itself that many times how many segments are connected to the node. The life cycle of the ant-agent would end (in case other than completing the task) in the event of a dead end or if an agent from the same colony has already passed through the node. Therefore, the ant does not need to clone itself. In that case, the demands on computing power would significantly increase. Example of possible decisions at the node is shown in Figure 4.

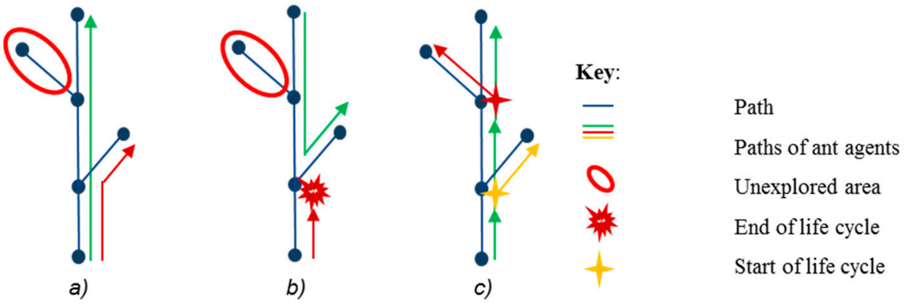

Figure 4. Example of possible decisions at the node: a) decision-making takes place at random, b) decision-making takes place through pheromones left by the same colony, c) cloning ant-agents at each node.

3.1. Nodes promotion and Dependence of reserved segments on the success of priority and future state predictions

At previous holonic levels production agents promoted their services, helping exploration colonies to find their path towards them. It is questionable whether it is necessary (in the event they are represented in the multiagent system by an agent) to promote their position (or their direction). A large number of nodes could cause an increase in the required processing power. However, in the event that nods representing production or logistic source were the only ones that are promoted at this level, the results of the finding process would improve.

The number of reserved segments is a parameter that contains a non-zero positive number indicating the number of segments the AGV agent will reserve on its way to a target node (segment). This number can be dynamically driven by the possibility to predict the expected future states of the system.

Estimated future states are formed by information and reservation colonies that have information about the future movement of the holonic agent AGV. The states that were estimated in the past are now compared with current states. Percentage ratio indicates success - the number of correct predictions against those that failed. The higher the percentage of successful predictions, the greater the number of segments AGV agents can reserve. Conversely, the lower the number of successful predictions, the smaller the number of segments AGV agents can reserve. This is a hypothesis that could help the system to adjust to the dynamics of the system. The more the system encounters unexpected changes, the lower would be the number of successful predictions and hence distances that agents would be able to reserve would shorten. This means that they would be prepared for changes much better. An interesting option is the increased number of reserved segments in the case where the AGV agent has a higher priority. The agent would then manage to reserve segments in advance and thus get better choice when choosing its path. An interesting idea appears to be reserving all possible segments on the path leading to the target point without depending on predictions or priorities. However, in this case we can anticipate rigidity widening.

3.2. Merging of exploration and information colony and reserving the initial segments

The original multi-agent systems were extended by the information colony due to a new feature - the ability to set priority on holonic agent AGV. This priority could be implemented well in the exploration ant colony, but if those ants beared the priority information (one exploration colony consists of several ant-agents), the situation in the system would have become less clear, since agents of the intelligent segment would change the order of crossings under this priority much more frequently. For this reason the information colony consisting of only one ant-agent passing the path chosen by the holonic agent AGV has been established. To better illustrate the operation of the first holonic levels different scenarios for testing management and coordination of holonic agent AGV have been created. To make the situation even harder, the logical holonic agent AGV is not allowed to go back or rotate at the segment. It can move only forward or stand. Holonic agent AGV have in scenarios the same priority.

Initial segments reservation refers to a condition of reserving and occupying segments even before the holonic agent AGV finds, selects sand reserves the path. In this case, the knowledge base of the AGV agent fails to provide the intelligent segment agent with information about when he will leave the physical part he represents in the multi-agent system. The reservation is set to infinity (to a sign not_a_date_time). At the moment when the knowledge base knows the time of departure, it informs the segment. Until then the segment is impassable. The same applies if he occupies a group of segments.Segment (or segment groups) is indeed impassable, but that does not mean that another holonic agent AGV cannot stop at the given place. For this situation it is necessary to calculate what portion of the segment (s) is being occupied by the holonic agent. This is calculated from its position and length. If several holonic agent AGV s stand on a single segment (or a segment group) and there is no other alternative path to take, the length of the reservation is infinite. This means that none of AGVs standing in a row, except the first one, release its ant colonies to find a path because they do not know the possible departure time of the first standing holonic agent. If the first holonic agent chooses and reserves the first possible path, it changes the infinite reservation status of the segment (segment groups) that it currently occupies. The holonic agent that is second in the order receives information that the reservation is not infinite anymore, and the situation is repeated until the segment fully passable. During the infinite reservation the segment cannot be passed and reservations cannot be made (except for the first one in order).

3.3. A path in a restricted environment with a single loading and unloading area

This scenario features two holonic agent AGV represented on the fourth holonic level by avatars (Avatar A, located in the unloading area - node 1, and Avatar B, located in the loading area - node 2). They share the same path along which they move from loading into the unloading area and vice versa. Travel time is consistent with loading / unloading operations. Avatar A is to find a path to the loading area with the best price e and Avatar B is to find a path to the unloading area with the best price. Imagine an initial situation in which the two avatars release their ant colonies at the same time. Since in this case there is no alternative path, exploring colonies will find only one possible path. The colony of the Avatar A moves from the loading zone through the segments in the specific order: 7, 2, 3, 4, 5 and 6. The colony of the Avatar B moves at the same time in the opposite direction starting from the loading area through the segments in the specific order: 6, 5, 4, 3, 2 and 1. The example of a possible scenario which includes two avatars, loading and unloading area is shown in Figure 5. Due to limited space they share one path.

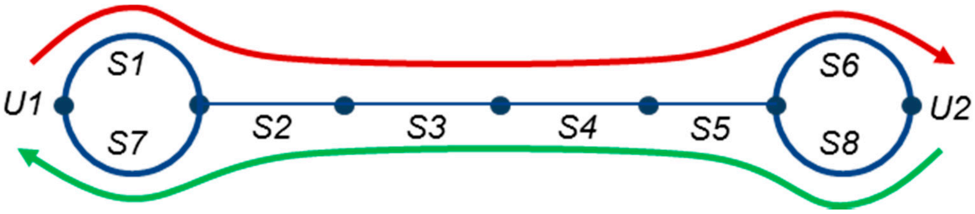

Figure 5. Example of a possible scenario which includes two avatars, loading and unloading area. Due to limited space they share one path.

In the case of the original character of D-MAS ant-agents should receive from each intelligent segment agent an answer to the question WHAT-IF: "If the Holonic agent AGV comes at this time, at what time will he leave the segment?" Exploring ant colony A would get an answer regarding the time at the segment 2 and segment 3. The answers would follow up on themselves, i.e. the Holonic agent AGV could enter the segment 2 and leave it by going to the segment 3 without delay. The colony B is a similar case (segments 4 and 5). However, if the colony A entered the segment 4 and at the same time the colony B crossed the segment 3, intelligent segment agents would inform them that time slots at which AGVs were supposed to enter the area are occupied. Therefore they move the reservation to the place in their schedule where the passing Holonic agent AGV (for the segment 3 it would be the Holonic agent AGV A and for the segment 4 it would be the Holonic agent AGV B) should have already been present. Having this information, the ant colony would continue and adapt the inquiry aimed towards intelligent segments agents according to information received at the segments 3 and 4. After the discovery of the loading and unloading area by both ant colonies, an avatar would extract information about the desired segment and possible time reservations at segments that have already passed through. Holonic agent AGV avatars would then send out ant reservation colonies (each should consist of one ant reservation agent) which would then reserve the segments for specified time intervals. After passing information about successful reservation of all segments that are needed to achieve the goal, both Holonic agent AGV s are supposed to go out from their places. However, they would meet at the node between segments 3 and 4 (at best) thanks to their safety sensors that give information about free path ahead (to a certain distance). The sensors would detect each other and stop until the operator resolves the situation by manually withdrawing one of Holonic agent AGV s out of the way. Such behaviour of the logistics system is undesirable. One approach to solve this situation is to assign segments 2, 3, 4 and 5 (segments which follow each other without intersection) one intelligent segment agent (i.e. to draw them into a coherent logical segment) while its orientation would change with the change of the reservation direction (in a given period of time the segment can be passed only in one direction).

3.4. A path in a restricted environment with a single loading and unloading area with an intersection in the middle

We can extend the previous scenario by including an intersection – adding additional segments to the already existing node (join segments S9 and S10 to U3 node). This scenario with added intersection is illustrated in Figure 6.

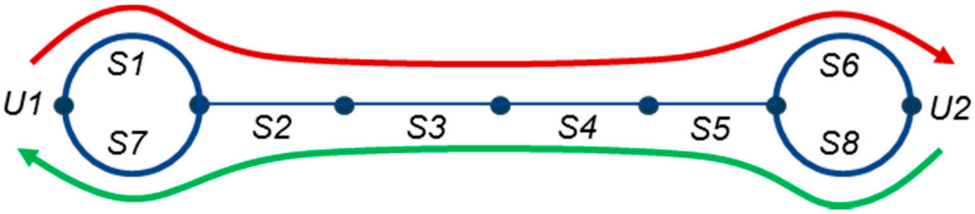

Figure 6. Previous scenario with added intersection

In this case, it is possible to create logical segments out of the segments S2 and S3, S4 and S5, S9, S10. By doing this we once again arrive to the situation outlined in the previous chapter. The solution to this situation would be marking the logical segments in question as critical. In that case, the agent would have to reserve the entire path composed of critical segments in the following time sequence: one Holonic agent AGV after another will pass through S2 to S5 segments respectively according to their given priority / UID.

3.5. Slowdown on the way

Imagine a scenario where Holonic agent AGVs are on their way slowed down or stopped because of an unexpected blockade or obstacle. The Holonic agent AGV has reserved a certain path and repeats the reservation during its journey. If the Holonic agent AGV slows down or stops, it still sends its ant colonies out to search for the path but now with updated arrival time for next segments. If the obstacle is removed, it will select the earliest possible time from the possible departures schedule.

4. Discussion

The authors have proposed the technological structure to eliminate these disadvantages and to add more useful features for control distributed systems. This new abstract architecture for control and coordinating decentralized systems is named CODESA and is shown in Figure 7. Application of this technological structure is possible in domains where control of non-linear, heterarchical, large-scale systems in dynamic, heterogeneous and unpredictable environment is required.

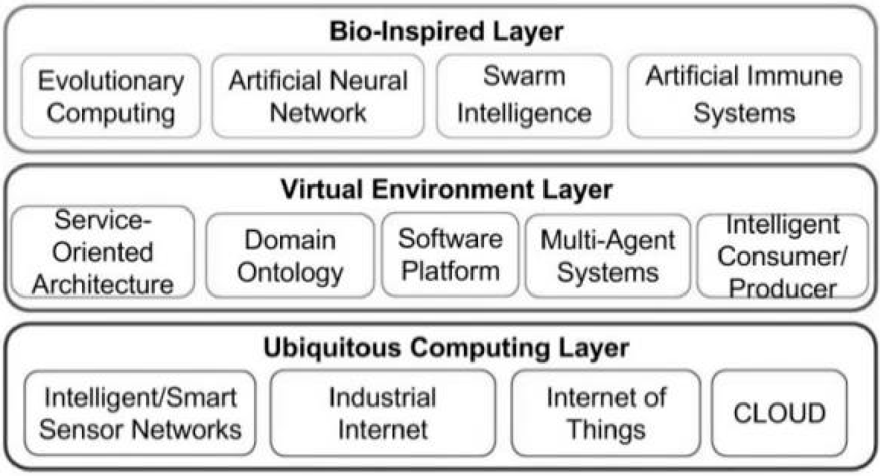
Bio-Inspired Layer
Evolutionary Computing
Artificial Neural Network
Swarm Intelligence
Artificial Immune Systems
Virtual Environment Layer
Service-Oriented Architecture
Domain Ontology
Software Platform
Multi-Agent Systems
Intelligent Consumer/ Producer
Ubiquitous Computing Layer
Intelligent/Smart Sensor Networks
Industrial Internet
Internet of Things
CLOUD

Figure 7. New abstract technological structure CODESA.

This technological structure consists of three layers: Ubiquitous Computing Layer, Virtual Environment Layer, Bio-Inspired Layer.

4.1 Ubiquitous computing layer

Ubiquitous environment is the concept of software engineering and computer science, where computing can be applied to natural interface, context-aware, and automate capture. The term potential of ubiquitous computing means everywhere-presence information technology and computing power that is present in everyday objects that we use. The idea of integrating computers seamlessly into the world at large runs counter to a number of present-day trends. Even the most powerful notebook computer, with access to a worldwide information network, still focuses attention on a single box. This potential of ubiquitous computing can be applied to whole areas of industrial production and private daily activities. This situation is made possible thanks to technological advances, availability, small size and price of computing devices. This layer provides industrial ubiquitous internet of things with a unique identity, communicating in real time with the possibility of processing and storing data on the side of the device or on the side of high-performance-computing system - the CLOUD.

4.2 Virtual environment layer

Virtual environment layer provides virtual entities, which can reflect the entities in real world. These entities are autonomous, intelligent and capable of decision-making, communicating with each other using the domain ontology and integrating sub-systems and systems into self-organizational structure. Lepuschitz [20] evaluated the overview of the ontologies and their suitability for process automation domain and its subdomains. This layer provides a picture of the real factory in a virtual environment. Agents can be located in the Software Platform or can be merged with devices (resources, products, etc.). The target cost of route of the services c (1) in routing intelligent products and allocation of production capacities based on weights of the following priorities:

- w_time is weight of time,
- w_qos is weight of quality of service,
- w_envimp is weight of environment impact.

To determine target (lowest) cost of all services in route(c) defined by weights:

$$c = \frac{\sum_{Si\in r}(c\_time(s_i))\times w\_time}{V_{r\,j\in P}((\sum_{Si\in r_j}(c\_time(s_i)))} + \frac{\sum_{Si\in r}(c\_qos(s_i))\times w\_qos}{V_{r\,j\in P}((\sum_{Si\in r_j}(c\_qos(s_i)))} + \frac{\sum_{Si\in r}(c\_envimp(s_i))\times w\_envimp}{V_{r\,j\in P}((\sum_{Si\in r_j}(c\_emvimp(s_i)))} \qquad (1)$$

Where w_time, w_qos, w_envip $\in$R, $0 \leq$ w_ x ≤ 1, si is target service, rj is specific route, W is the set of all possible routes, c_time is cost for time, c_qos is cost for quality of target service, c_envimp is cost for environment impact. The higher is the weight of given priority, the lower is the total price, and therefore more acceptable. Target cost of route (2) in routing the logistics resources through the network of roads is based on weights of the following priorities:

- w_dist is weight of distance,
- w_time is weight of time,
- w_qos is weight of quality of service,
- w_envimp is weight of environment impact.

To determine target (lowest) cost of route(r) defined by weights:

$$r = \frac{\sum_{S i \in r}(c\_dist(s_i)) \times w\_dist}{V_{r\,j \in P}((\sum_{S i \in r_j}(c\_dist(s_i)))} + \frac{\sum_{S i \in r}(c\_time(s_i)) \times w\_time}{V_{r\,j \in P}((\sum_{S i \in r_j}(c\_time(s_i)))} + \frac{\sum_{S i \in r}(c\_qos(s_i)) \times w\_qos}{V_{r\,j \in P}((\sum_{S i \in r_j}(c\_qos(s_i)))} + \frac{\sum_{S i \in r}(c\_envimp(s_i)) \times w\_envimp}{V_{r\,j \in P}((\sum_{S i \in r_j}(c\_emvimp(s_i)))} \tag{2}$$

Where w_dist ∈ R, 0 ≤ w_x ≤ 1, si is one segment of route, rj is concrete route and P is the set of all possible routes. The higher is the weight of given priority, the lower is the total price, and therefore is more acceptable. Then c_dist is cost per distance, c_time is cost per time, c_qos is cost per quality of service, c_envimp is cost per environment impact.

4.3 Bio-Inspired Layer

Bio-Inspired Layer represents a set of biologically inspired techniques and technologies. Swarm Intelligence can be used to eliminate some of the previously mentioned negative characteristics (derived from heterarchical architectures, multi-agent systems, lack of adaptability, etc.), using indirect pheromone-based communication.

As mentioned, the application of technological structure of the CODESA is possible in areas where it is necessary to control non-linear, heterarchical, large-scale systems in dynamic, heterogeneous and unpredictable environment. Possible application domains are: CODESA-Prime is compatible with the Industry 4.0, transportation, agriculture, healthcare, military, power grids, etc. With technological structure designed for manufacturing domain CODESA-Prime, it is possible to control, plan, schedule, manufacturing, optimize processes, route product and mobile agents, manage inventory, create prediction, etc.

4.4 CODESA-Prime: Virtual factory layer

In CODESA-Prime is Virtual environment layer called Virtual factory layer. Three main types of agents characterize Prime:

- Intelligent Order Agent (IOA),
- Intelligent Product Agent (IPA),
- Intelligent Resource Agent (IRA).

These three types of agents form the Holon Smart Factory. The relationship is shown in Figure 8.

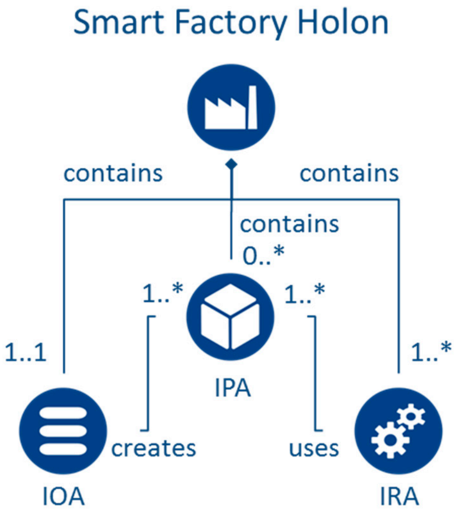

Figure 8. The relationship between the main types of agents in the architecture CODESA-Prime.

CODESA-Prime has been tested in simulations for Automated Guided Vehicle systems (Holonic agent AGV s), which have been tested basic aspects of multi-agent systems in Ella Software Platform. The example from the simulation is shown in Figure 9.

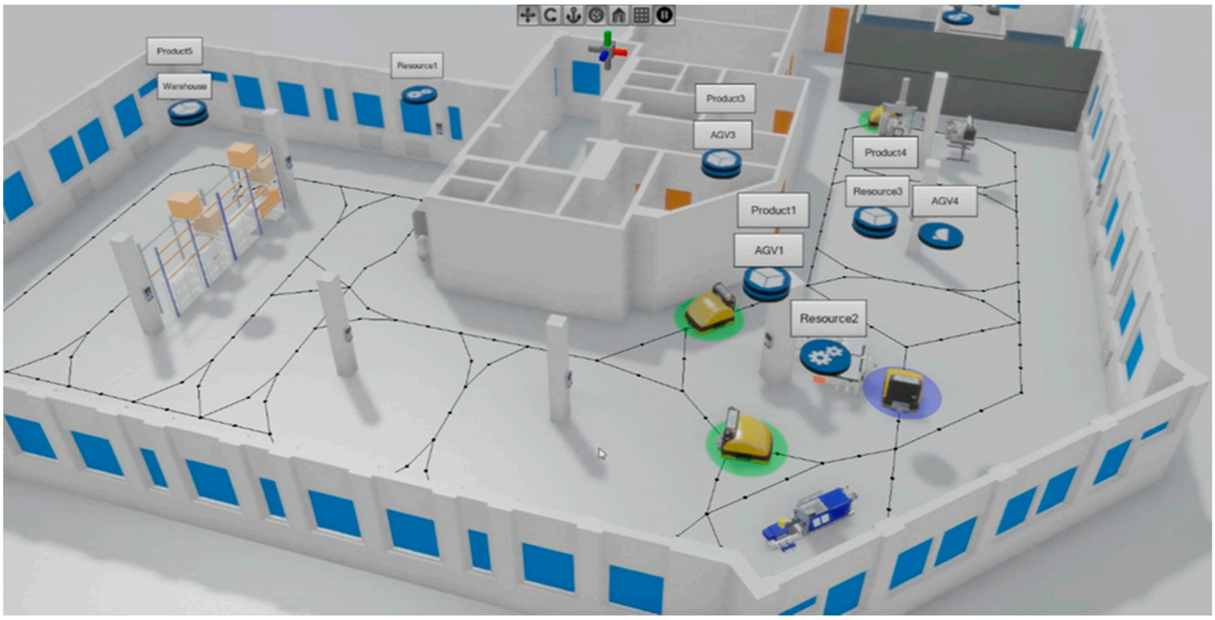

Figure 9. Testing of CODESA-Prime for Holonic agent AGV in Ella Software

Each holonic agent AGV is represented by an Intelligent Resource Agent and presents its service to Intelligent Product Agent. Agent Holonic agent AGV reserves different path segments, which are also represented by the agent (Intelligent Agent Segment). However, such encapsulation of the services must be dynamic. Ella Software Platform provides services, such as the virtual environment that is the model (reflection) of the real factory, in which the agents can travel through, base for dual communication, and meta-schema through which ant-agents can move and indirectly communicate using pheromones. The connection of scheduling capabilities for logistics systems and routing through production optimize the use of logistical resources, as well as individual road segments. Simulation has been verified that the CODESA-Prime has proved for logistics systems. CODESA-Prime: Bio-Inspired Layer Agents can create (cast) lightweight agents, also called ants or ant agents, and delegate them across the virtual environment that can be represented as graph.

These lightweight agents have a short life (defined by agents) and can perform only certain specific activities. This means that these ant-agents are computationally efficient. They can deposit, locate, observe and modify digital pheromones in the graph (virtual environment). These digital pheromones represent indirect communication. Pheromone-based communication is suitable to control social and temporal myopic behaviour that is needed to handle uncertainties in manufacturing processes. As mentioned, CODESA uses dual communication. Indirect communication belonging to Bio-Inspired Layer has the following properties:

- It is inspired by food foraging in ant colonies.
- Replaces direct communication.
- Communication is done through virtual environments (represented by weighted graphs) by depositing, observing and modifying the pheromones.
- Delegated, lightweight agents called ants deposit pheromones in virtual environments. These ants are casted periodically by main agents (IPA, IRA).
- Pheromones decay in time. In graph (meta-scheme) are relations between individual main agents (or theirs avatars) are represented by graph´s edges, meta-schema (Figure 6).
- Ants deposit pheromones at graph´s vertices.
- Each ant has temporary memory, limited processing power and limited life.

5. Conclusions

The use of indirect communication in CODESA-Prime is used at:

- Routing intelligent product through factory´s services and allocation of the resources.
- Routing logistics systems through a network of paths.

The main scientific contribution is reference model CODESA that can be used to control and coordinate decentralized systems in the heterogeneous applications and first concrete application into the manufacturing domain CODESA-Prime. This section presents Manufacturing Multi-agent system with bio-inspired techniques: CODESA-Prime for control and coordination decentralized systems that are applied to manufacturing domain. Prime uses indirect communication, based on pheromones, that eliminates drawbacks of multi-agent systems and can provide the distance of reliable prediction horizon. Indirect communication is used for routing intelligent product through factory and resource allocation and routing logistics systems over a network of roads. Flow of pheromones can be adapted by the testing of ant-agent colonies ant genetics algorithms. Prime consists of three main agents, namely: Intelligent Order Agent, Intelligent Product Agent and Intelligent Resource Agent. Prime can be eco-friendly, supports cleantech, creating eco-innovation, monitor energy consumption, and propose corrective action. CODESA-Prime has been tested in simulations for Holonic agent AGV s, which have been tested the basic aspects of multi-agent systems in the Ella Software Platform. In further research, results should be transferred into the industrial practice to control and coordinate Holonic agent AGV s by the company Central European Institute of technology (CEIT) located in Zilina, Slovakia.

Acknowledgments: This paper was created with support of APVV project: APVV-14-0752: Reconfigurable logistic system for new generation of Factory of the Future manufacturing systems (RLS_FoF).

Author Contributions: Professor Micieta conceived the structure and research direction of the paper; Lukas Durica wrote the paper and completed the simulation for case studies, provided algorithms and, wrote programs; Vladimira Binasova revised the writing of the entire manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Al-Safi, Y.; Vyatkin, V. An Ontology-Based Reconfiguration Agent for Intelligent Mechatronic Systems. In *Holonic and Multi-Agent Systems for Manufacturing*; Springer Berlin Heidelberg: Berlin, Heidelberg, 2007; pp. 114–126.

2. Alsafi, Y.; Vyatkin, V. Ontology-based reconfiguration agent for intelligent mechatronic systems in flexible manufacturing. *Robot. Comput. Integr. Manuf.* **2010**, *26*, 381–391.

3. Dulina, L.; Bartanusova, M. ScienceDirect CAVE Design Using in Digital Factory. *Procedia Eng.* **2015**, *100*, 291–298.

4. Krajcovic, M.; Bulej, V.; Sapietova, A.; Kuric, I. Krajčovič, M. et al. (2013) Intelligent Manufacturing Systems in concept of digital factory. *Commun. Sci. Lett. Univ. Žilina* **2013**, *15*, 77–40.

5. Barbosa, J.; Leitão, P.; Adam, E.; Trentesaux, D. Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. *Comput. Ind.* **2015**, *66*, 99–111.

6. Zambrano Rey, G.; Pach, C.; Aissani, N.; Bekrar, A.; Berger, T.; Trentesaux, D. The control of myopic behavior in semi-heterarchical production systems: A holonic framework. *Eng. Appl. Artif. Intell.* **2013**, *26*, 800–817.

7. Zambrano Rey, G.; Bonte, T.; Prabhu, V.; Trentesaux, D. Reducing myopic behavior in FMS control: A semi-heterarchical simulation–optimization approach. *Simul. Model. Pract. Theory* **2014**, *46*, 53–75.

8. Barbosa, J.; Leitão, P.; Adam, E.; Trentesaux, D. Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. *Comput. Ind.* **2015**, *66*, 99–111.

9. Thomas, A.; Trentesaux, D.; Valckenaers, P. Intelligent distributed production control. *J. Intell. Manuf.* **2012**, *23*, 2507–2512.

10. Pannequin, R.; Morel, G.; Thomas, A. The performance of product-driven manufacturing control: An emulation-based benchmarking study. *Comput. Ind.* **2009**, *60*, 195–203.

11. Leitão, P. Towards Self-organized Service-Oriented Multi-agent Systems. **2013**, 41–56.

12. Leitão, P.; Mařík, V.; Vrba, P. Past, present, and future of industrial agent applications. *IEEE Trans. Ind. Informatics* **2013**, *9*, 2360–2372.

13. Tombak, M.; De Meyer, A. Flexibility and FMS: an empirical analysis. *IEEE Trans. Eng. Manag.* **1988**, *35*, 101–107.

14. Mehrabi, M. G.; Ulsoy, A. G.; Koren, Y. Reconfigurable manufacturing systems: Key to future manufacturing. *J. Intell. Manuf.* **2000**, *11*, 403–419.

15. Koren, Y.; Shpitalni, M. Design of reconfigurable manufacturing systems. *J. Manuf. Syst.* **2011**.

16. Vrba, P.; Tichý, P.; Mařík, V.; Hall, K. H.; Staron, R. J.; Maturana, F. P.; Kadera, P. Rockwell Automation's Holonic and Multiagent Control Systems Compendium. *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.* **2011**, *41*, 14–30.

17. Mendes, J. M.; Leitão, P.; Restivo, F.; Colombo, A. W. Service-Oriented Agents for Collaborative Industrial Automation and Production Systems. In; Springer Berlin Heidelberg, 2009; pp. 13–24.

18. Leitão, P.; Restivo, F. A holonic approach to dynamic manufacturing scheduling. *Robot. Comput. Integr. Manuf.* **2008**, *24*, 625–634.

19. Trentesaux, D.; Thomas, A. Product-Driven Control: Concept, Literature Review and Future Trends. In; Springer Berlin Heidelberg, 2013; pp. 135–150.

20. Lepuschitz, W.; Lobato-Jimenez, A.; Axinia, E.; Merdan, M. A Survey on Standards and Ontologies for Process Automation. In; Springer International Publishing, 2015; pp. 22–32.

21. Hsieh, F.-S. Dynamic composition of holonic processes to satisfy timing constraints with minimal costs. *Eng. Appl. Artif. Intell.* **2009**, *22*, 1117–1126.

22. Thomas, A.; Trentesaux, D. Are intelligent manufacturing systems sustainable? Are intelligent manufacturing systems sustainable?. 3rd Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics. **2013**, 105–114.

23. Durica, L.; Micieta, B.; Bubenik, P.; Binasova, V. Manufacturing multi-agent system with bio-inspired techniques: CODESA-Prime. **2015**, *2015*, 829–837.

24. Durica, L.; Micieta, B.; Bubenik, P.; Binasova, V. MANUFACTURING MULTI-AGENT SYSTEM WITH BIO-INSPIRED TECHNIQUES: CODESA-PRIME. *MM Sci. J.* **2015**, *2015*, 829–837.