Agent-Based Systems for Manufacturing

L. Monostori[1,2] (1), J. Váncza[1,2] (2), S.R.T. Kumara[3] (1)

[1]Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary
[2]Department of Production Informatics, Management and Control,
Budapest University of Technology and Economics, Budapest, Hungary
[3]The Pennsylvania State University, Industrial and Manufacturing Engineering,
University Park, PA 16802, USA

Abstract

The emerging paradigm of agent-based computation has revolutionized the building of intelligent and decentralized systems. The new technologies met well the requirements in all domains of manufacturing where problems of uncertainty and temporal dynamics, information sharing and distributed operation, or coordination and cooperation of autonomous entities had to be tackled. In the paper software agents and multi-agent systems are introduced and through a comprehensive survey, their potential manufacturing applications are outlined. Special emphasis is laid on methodological issues and deployed industrial systems. After discussing open issues and strategic research directions, we conclude that the evolution of agent technologies and manufacturing will probably proceed hand in hand. The former can receive real challenges from the latter, which, in turn, will have more and more benefits in applying agent technologies, presumably together with well-established or emerging approaches of other disciplines.

Keywords:
Agent, Multi-agent systems, Manufacturing

1 INTRODUCTION

The past decade has witnessed an explosive growth in computer, communication, and information technologies. High-performance computing, the world-wide web, universal access and connectivity, virtual reality, and enterprise integration are but a sampling of this revolution's many facets. At the same time, organizations and markets have also changed dramatically, represented by developments such as virtual organizations, business process integration, customer-centric supply chains, and electronic commerce. And, although industry strategists and academics continue to debate the precise future trajectory of changes in technologies and organizations, they agree that **information** - its availability, and the ability to exchange it seamlessly and process it quickly - lies at the core of organizations' abilities of meeting escalating customer expectations in global markets.

Agent-based computation is a new paradigm of information and communication technology that largely shapes and, at the same time, provides supporting technology to the above trends [1] [2] [3]. Agent theories and applications have appeared in many scientific and engineering disciplines. Agents address autonomy and complexity: they are adaptive to changes and disruptions, exhibit intelligence and are distributed in nature. In this setting computation is a kind of social activity. Agents can help in self-recovery, and react to real-time perturbations. Agents are vital in the globalization context, as globalization refers to an inherently distributed world both from geographical and information processing perspectives.

Agents – and similar concepts – were welcome in **manufacturing** because they helped to realize important properties as autonomy, responsiveness, redundancy, distributedness, and openness. Agents could be designed to work with uncertain and/or incomplete information and knowledge. Hence, many tasks related to manufacturing - from engineering design to supply chain management - could be conducted by agents, small and large, simple and sophisticated, fine- and coarse-grained that were enabled and empowered to communicate and cooperate with each other.

The goal of the keynote paper is to explore the software agent technology and clearly show the applications and opportunities for agent technology in manufacturing. Though we survey the field from both manufacturing and computing perspectives, we pay special attention to the work done by the CIRP community. For other reviews about the application of agent technology in manufacturing, see, e.g., [4] [5] [6].

The structure of the paper is as follows: We start with the new emphasis in manufacturing, and how information technology is changing the face of manufacturing. This is followed with the definitions of software agents and multi-agent systems in Section 3. Section 4 introduces basic agent technologies concerning individual agents, agent interactions as well as organization models of multi-agent systems. In Section 5 we undertake a fairly exhaustive and rigorous survey of applications of agent technology in various domains of manufacturing. Departing from this detailed survey, next (in Section 6) we sum up the methodological issues of agent applications, with a special emphasis on running industrial applications in manufacturing. Section 7 deals with open issues and strategic directions. We conclude the paper in Section 8.

2 PARADIGM SHIFTS IN MANUFACTURING

In discrete manufacturing, developments in information technology led to the realization of computer integrated manufacturing systems. With all of its merits, integration resulted in rigid, hierarchical control architectures whose structural complexity grew rapidly with the size and the scope of the systems. Moreover, integration resulted also in complex decision problems [7] [8]. Disturbances in manufacturing, as well as changing market demand provided an unstable environment. By now it has become clear that increased volatility of market conditions disfa-

doi:10.1016/j.cirp.2006.10.004

vors these rigid, hierarchical architectures. No matter how, it is next to impossible to be prepared with pre-programmed, top-down responses to abrupt changes and to complete real-time computations on sophisticated decision models before the results are invalidated.

Long ago, manufacturing system theory suggested cautious and rather pessimistic organizing principles for systems exposed to substantial internal and external uncertainties [9] [10]. These principles say that it is better to recognize ignorance than to presume knowledge; that it is better to regard the future as unpredictable than only to be prepared for expected events. From this stance, there follows a system structure with distributed responsibilities, tasks and resources. Accordingly, manufacturing systems should be organized as loosely coupled networks of communicating and cooperating components [11] or agents [12]. Redundant functions and responsibilities are necessary for absorbing sudden changes, regardless of whether they stem from internal disturbances or from volatile market conditions. When doing so, there is an increasing need for giving real-time responses [13].

Decision rights should be co-located with the pertinent information; time should be seriously considered as a limiting resource of decision-making, and the systems should have changeable, easy-to-reconfigure organizational structures. Growing complexity is another feature showing up in production processes and systems, furthermore, in enterprise structures as well [14].

The concepts of process modelling [15], virtual manufacturing [16], the digital enterprise [17], i.e., the mapping of the key processes of an enterprise to digital structures by means of information and communication technologies (ICT), give a unique way of managing the above problems. By using recent advances in ICT, theoretically, all the important production-related information is available and manageable in a controlled, user-dependent way [18]. Product-related information can follow the products throughout their life-cycles [19] [20] [21].

Earlier, it became evident that the management and exploitation of this huge amount of information cannot be imagined without the effective application of the methods and tools of artificial intelligence [22] [22] [24], or more specifically, machine learning techniques [25]. In the current age of globalization, virtual enterprises and production networks [26], some other requirements are more relevant than ever to be considered: we have to solve the coordination of various production entities that may have very different objectives, goals, capabilities, and even cultures. New organizing principles and methods are needed for supporting the interoperability of dynamic virtual alliances of agile and networked enterprises which - when acting together - can make use of opportunities without suffering from diseconomies of scale.

Various solution proposals unanimously imply that the future of manufacturing lies in the loose and temporal federations of cooperative autonomous production entities. However, nodes of a network organization can never be completely aware of each other's goals and objectives, so they have almost necessarily adverse incentives. Hence, coordination, collaborative work and cooperation may play an important role in managing these complex structures [27]. Since the new information technologies allow members of a network to widen their span of interest and control, the distribution of decision rights and actions introduces some new elements of uncertainty that can be resolved by communication (i.e., information sharing) and cooperation only. The interaction of individuals may lead to the emergence of complex system-level behavior [28]. Evolutionary system design relies on emergence when modeling and analyzing complex manufacturing and, in a wider context, eco-technical systems.

Under the pressure of the above challenges, the transformations of manufacturing systems and organizations are already underway [29]. The need for novel organizational principles, structures and method has called forth various approaches [30] in the past decade, such as holonic [31] [32], fractal [33], random [34], biological [35], and multi-agent manufacturing systems [36], to name but those investigated by CIRP colleagues.

All the above approaches are similar in that they assume network-like, dynamic, open and reconfigurable systems where decisions are made and production is carried out by more or less independent and cooperative partners. In the next two sections we present the information technology background of these recent developments.

3 AGENTS AND MULTI-AGENT SYSTEMS

The theory of computational agents goes back at least a quarter of a century when research in distributed artificial intelligence (DAI) had been initiated [37]. In the early 90's the notion of agents appeared simultaneously also in information and communication technology (mobile, interface and information agents). Agents made the real breakthrough a decade ago or so when the emphasis in the mainstream AI research shifted: the focus on logic was extended and attention changed from goal-seeking to rational behavior; from ideal to resource-bound reasoning; from capturing expertise in narrow domains to re-usable and sharable knowledge repositories; from the single to multiple cognitive entities acting in communities [38]. These developments also coincided with the evolution of network-based computing technology, the internet, mobile computing, the ubiquity of computing as well as novel, human-oriented software engineering methodologies.

All these achievements led to what is considered now the **agent paradigm** of computing [1] [2] [39]. While this novel paradigm has several roots as far as theory, enabling technologies and applications are concerned, there is a general consensus about its two main abstractions:

- An **agent** is a computational system that is situated in a dynamic environment and is capable of exhibiting autonomous and intelligent behavior.
- An agent may have an environment that includes other agents. The community of interacting agents, as a whole, operates as a **multi-agent system**.

In the coming paragraphs we give a concise characterization of agents and multi-agent systems.

3.1 Basic properties of agents

An agent operates in an **environment** from which it is clearly separated (Figure 1). Hence, an agent (1) makes observations about its environment, (2) has its own knowledge and beliefs about its environment, (3) has preferences regarding the states of the environment, and finally, (4) initiates and executes actions to change the environment.

Agents operate typically in environments that are only partly known, observable and predictable. **Autonomous** agents have the opportunity and ability to make decisions of their own. **Rational** agents act in the manner most appropriate for the situation at hand and do the best they can do for themselves. Hence, they maximize their expected utility given their own local goals and knowledge. Rationality can be bound by the computational complexity of a decision problem, the limitation of computing resources, or by both. An agent with optimization objectives but with limited means is a **bounded rational** agent.

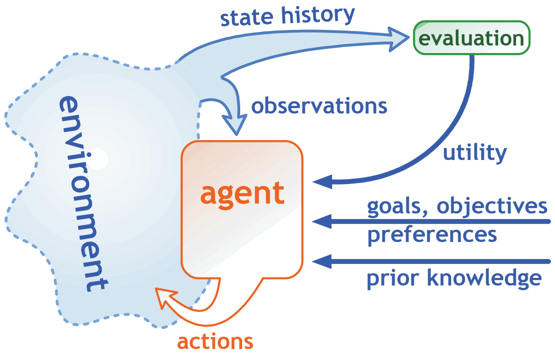

Figure 1: The agent and its environment [40].

The most important common properties of computational agents are as follows:

- Agents act on behalf of their designer or the user they represent in order to meet a particular **purpose**.
- Agents are **autonomous** in the sense that they control both their internal state and behavior in the environment.
- Agents exhibit some kind of **intelligence**, from applying fixed rules to reasoning, planning and learning capabilities.
- Agents **interact** with their environment, and in a community, with other agents.
- Agents are ideally **adaptive**, i.e., capable of tailoring their behavior to the changes of the environment without the intervention of their designer.

Further agent properties, characteristic in particular domains and applications, are **mobility** (when an agent can transport itself to another environment to access remote resources or to meet other agents), **genuineness** (when it does not falsify its identity), **transparency**, and credibility or **trustworthiness** (when it does not communicate false information willfully).

Even though they exhibit only some of the above properties, agents relax several strong assumptions of classical computational intelligence: they typically have incomplete and inconsistent knowledge as well as limited reasoning capabilities and resources.

3.2 Multi-agent systems

A multi-agent system (MAS) is formed by a network of computational agents that interact and typically communicate with each other.

The agents may have only a partial (and, in a sense, distorted) model of their environment: they may possess a limited set of means for the acquisition and integration of new knowledge into their models and for pushing the system's state towards their own goals. The knowledge of two agents, referring to the same things, is not necessarily commensurate and may have different representations. No closed-system assumption has to be maintained: the MAS is submerged into and interacts with its environment, which is not described completely (or is difficult to describe) by formal means. Whenever novel kinds of interaction with the environment may occur, the MAS should be open and able to evolve.

In a multi-agent system the decisions and actions of various agents do necessarily interact. However, just due to **interaction**, a multi-agent system can occasionally solve problems that are beyond the limits of the competence of the individual agents and/or may exhibit **emergent behavior** that cannot be derived from the internal mechanisms of the components.

In a community an agent has to **coordinate** its actions with those of the other agents; i.e., to take the effects of other agents' actions into account when deciding what to do. Coordination models provide both **media** (such as channels, blackboards, pheromones, market, etc.) and **rules** for managing the interactions and dependencies of agents. Coordination requires some regulated flow of information between the agent and its surrounding environment – in other words, **communication**. Note that in a MAS coordination is possible both by indirect communication via the environment, or by direct information exchange between specific agents. In any case, communication needs some **language(s)** with syntax and semantics, at least partially known for each communicating agent.

Collaboration means carrying out concerted activities so as to achieve some shared goal(s). For instance, in a scheduling domain machine agents may agree on executing each task of a job with the aim of completing an order by the given due date. The shared goal (completing an order) can be achieved only if all agents commit themselves to carrying out the actions they have agreed upon. In general, meeting high-level objectives and satisfying system-wide constraints need **cooperation** in a multi-agent system where agents are self-interested and autonomous.

The overall operation of a multi-agent system is affected by an **organization** that is imposed on the individual agents. Even though there may be no global control or centralized data and the computations are asynchronous, some organizational rules always exist. The organization determines the “sphere” of the activity of agents, as well as their potential interactions (see Figure 2).

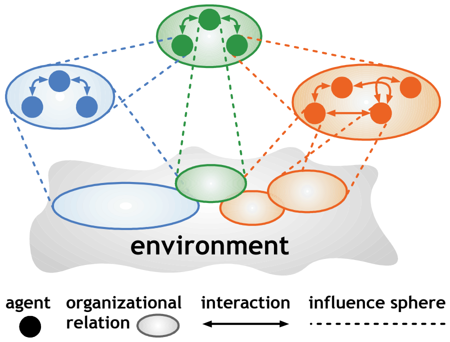

Figure 2: Generic scheme of a multi-agent system [41].

The various organization patterns of multi-agent systems, such as teams, coalitions, markets, as well as hierarchic and heterarchic (including holonic) architectures provide different ways to achieve system-wide design objectives and/or to facilitate the emergence of desired properties in multi-agent systems.

4 AGENT TECHNOLOGIES

4.1 Agent level

Summing up, agents are individual problem-solvers with some capability of sensing and acting upon their environment, for deciding their own course of action, as well as for communicating with other agents. Depending on the actual problem and available technology at hand, agents can apply various **faculties of problem solving**, including searching, reasoning, planning, and learning. The notion of agents has a strong synthesizing power; hence the applied techniques may include both symbolic and sub-symbolic methods, classical and quantitative decision theory, as well as knowledge-based reasoning and sophisticated belief-desire-intention (BDI) models [42] [43].

There are several approaches for realizing rational agents. Following the principles of classical decision theory, an agent makes choices from a set of alternative actions so that it can maximize the expected utility of its decisions. If a utility function cannot be expressed, or not all the required input data are available, qualitative models are applied that work with preferences.

An agent may have explicit knowledge of how its actions can change the states of its environment. Given some states to achieve – so-called **goals** – the reasoning over future courses of actions is a key component of rational behavior. Artificial intelligence (AI) provided a host of **planning** methods to solve this problem under various assumptions; many of them already have been proved to be applicable in real-life domains [44].

The most expressive model of an agent and its knowledge about the surrounding environment and itself is the so-called **BDI model**. This assumes the agent has both certain and uncertain knowledge – **beliefs** (B) – regarding the states of its environment. States to achieve are expressed in terms of goals, and states preferred in the long-term are represented by **desires** (D). Decisions concerning future events have motivations and prearrangements in the past: these are expressed by the so-called **intentions** (I) that represent the commitments of the agent made previously. Intentions can be thought of as past decisions behind a yet uncompleted plan. Sticking to intentions stabilizes the behavior of the agent. Agents who are situated in a dynamic environment can benefit from having **plans** which are collections of actions on a time line. Plans, on the one hand, can constrain the amount of reasoning, and, on the other hand, can make coordination possible [2] [45].

The BDI model is used for describing an agent's **cognitive state** and casts its decision problem in terms of cognitive concepts, such as knowledge and belief, desires and goals, plans and intentions [43] [40]. A BDI agent is continually updating its beliefs based on perceptions, using its beliefs to reason about possible plans, committing to certain intentions based on its beliefs and desires, and realizing these intentions by acting. The BDI formalism offers means to add and retract goals and generate actions.

4.2 Interaction

Agents necessarily **interact** with each other – either indirectly, via the environment, or by direct communication. The various coordination and cooperation mechanisms range from emergent methods (without explicit communication among agents) to coordination protocols, coordination media and distributed planning [1].

Coordination protocols control the interactions of agents in order to reach common decisions. For instance, a widely used protocol is the **contract net protocol** (CNP) [46]. Here, a manager agent wants to have some task performed by one or more other agents. CNP is modeled on the contracting mechanism used by businesses to govern the exchange of goods and services. The manager agent announces the task to other agents and requests bids for the execution of the task. These bids are evaluated according to specified criteria (price, quantity, due date, distribution of tasks), and the winning bidder gets the contract. A **coordination media** provides a shared memory space for communicating data in an asynchronous way. Typical examples are blackboards [3] or the pheromones in stigmergy-based coordination [47] [45]. Goal-oriented agents forming a community may have disparate and conflicting goals. For resolving conflict situations, various **negotiation** mechanisms were developed, including auctions, one-to-one negotiation, bargaining, and argumentation-based negotiation [2].

If coordination is based on communication, agents must be able to talk to each other through some agent communication language. Collaborative acting and planning involve the intentions of multiple agents. Since the presence of other agents is always a source of uncertainty (beyond other possible sources), collaboration requires an integrated treatment of the beliefs and intentions of the agents who may take part in a collaborative act. That is why the BDI model provided the theoretical basis for **agent communication languages** (ACL), including the widely used Foundation for Intelligent Physical Agents (FIPA) standard [48]. ACL was developed based on the speech act theory [49]. Speech act theory views human natural language as actions, such as requests, suggestions, commitments, and replies. It uses the term **performative** to identify the intended meaning of utterances. Examples of performative verbs are *request*, *tell*, *insist* and *promise* to name few. The first ACL was the Knowledge Query and Manipulation Language (KQML) that included many performatives, assertives and directives which agents use for telling facts, asking queries, subscribing to services and/or finding other agents.

Communication and interoperability requires consensual knowledge of a community. A so-called **ontology** is an explicit specification of the conceptual structures of a given domain. It is usually expressed in a logic-based language that makes it possible to distinguish classes, instances, properties, relation and functions in a clear-cut, consistent way. Consensual means that the whole community has a common understanding both on the content and form of the expressed knowledge. Ontologies also can facilitate machine processing: automated reasoning, as well as the inter-operability of different agents. Note that current agent representation and communication techniques are presented in Section 6.2.

4.3 Organization

In any human community, goals are achieved by the **division of labor** and **coordination**. There are a number of organization structures that define various patterns of decomposing work, assigning responsibilities to those who do the work, as well as collecting and combining results. Like any community, a MAS is formed by agents that are aimed at achieving some purposes, be them individual, system-wide or both. It is no wonder that multi-agent systems adapt all the basic human organization patterns.

An organization of a MAS assigns **roles** to the agents and specifies their **relations**. The role identifies the agent and specifies its function, its rights and obligations, and the rules of its interaction with other agents and the environment. There are two basic types of roles: the **operator** accomplishes a task – a piece of work – usually, bound by some resource and time constraints. The **manager**, on the other hand, exercises control: assigns tasks, monitors task execution, collects and combines results. Depending on the actual organization scheme, agents may perform different roles at the same time. The position of an agent in an organization is given by the role(s) it fulfills and its relations to the other agents. Finally, the admissible interactions of agents are defined by ordered sets of messages, i.e., by **protocols**. A particular organization structure comes together with rules concerning the conditions of participating in a MAS, the assignment of the roles and relations, as well as the use of protocols - all of which together realize a particular **coordination mechanism**.

Organizational structures and coordination mechanisms of most MAS systems are adapted from some form of human organizations. Below, following [50] and [51], we present basic coordination mechanisms prevalent in multi-agent systems. In Figure 3 and Figure 4, requests from the environment (or other parts of the MAS) are transmitted by a requesting agent (R), handled by a manager agent (M), if any, and processed by operator agents (O). Agents without prior role are depicted as (A). The figures show the organizational relations and the main messages of the protocols.

In **direct supervision** a manager has control over some operator agents (see Figure 3). The manager takes responsibility for the work of the operators and arranges their coordination. This pattern may be repeated at several levels of a hierarchy. Typically, one way to take care of global objectives and system-wide constraints is to channel all/some interactions through a central coordinator (manager) agent. Due to its position, this agent obtains information essential to the operation of the complete system. Direct supervision reduces the number of inter-agent relations, but also makes the MAS very dependent on the central coordinator.

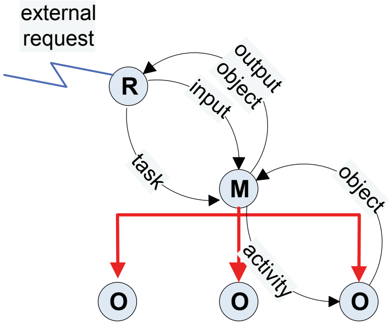

Figure 3: Direct supervision [51].

Standardization of work and/or skill delegates part of the control to the operators (see Figure 4): the manager instructs the operators with some procedures (i.e., task specification), but leaves the responsibility of execution in their hands. The operators work in a decentralized way. Consequently, they control the flow of objects. Operators are selected for taking part in a task execution on the basis of their competence and current load. A typical example of this coordination mechanism is when a manager agent arranges the work of a community of experts.

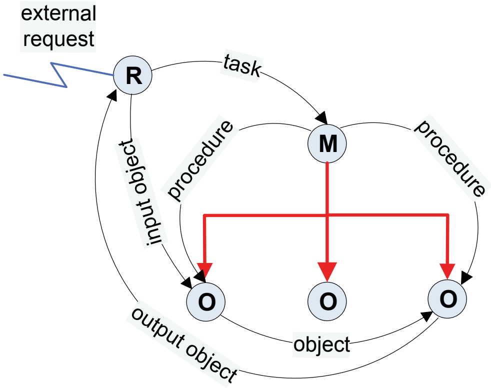

Figure 4: Standardization of work/skill [51].

Mutual adjustment is the coordination mechanism of a MAS without a priori organizational structure. Control and work are not separated; manager and operator roles are not distinguished any longer. The agent who gets a request cannot but communicate (by passing objects and/or tasks) with the other agents, though cannot instruct them (see Figure 5). This is the simplest coordination method, however, with no (or very limited) guarantee that the request will really be responded appropriately by the MAS, as a whole. One should not expect agents to behave in a cooperative, altruistic way. The power and benefits of individual control can be destroyed by poor cooperation at the community level. Trade-offs can be found by **negotiation** methods. A widely used mutual adjustment coordination technique is the formation and dissolution of **teams**.

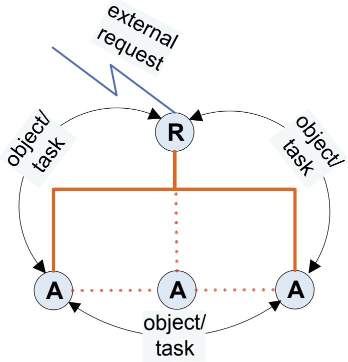

Figure 5: Mutual agreement [51].

In a MAS, however, human organizational models can be surpassed. The ideas of decentralized problem solving, including the resolution of the conflict between individual and collective good, are widely studied in a number of other disciplines, such as economics, game theory, political science, biology, and ecology. Computational models of agency borrowing analogies from these fields are similar in that they rely on some form of **self-organization**. No central control is exercised, and the system adapts its structure and functionality to the changing requirements and environmental conditions. Typically, members of the systems are individually able to achieve simple tasks, but their interactions lead to the **emergence** of complex collective behavior [28] [52]. In self-organizing systems, agents also communicate with each other indirectly: they use stigmergy (following ant communities) [45] [47] [53] or attraction-repulsion fields [54] [55]. Though, currently there is no generic coordination method that could be applied in engineering self-organizing systems [1].

Organization patterns in manufacturing

Numerous variants of the above organization models and cooperation patterns have appeared throughout the history of manufacturing. In fact, manufacturing called for new, more robust, adaptable, fault-tolerant, decentralized and open organizational structures even before the idea of agents and MAS emerged from artificial intelligence and computer science [10] [27]. Departing from the generic ideas of Koestler [56], these requirements led to the **holonic** organization model developed and applied particularly in manufacturing [45] [32] [57]. A holonic system consists of so-called **holons** all of which are assumed to be both autonomous and cooperative. The control is distributed: based on their own situation assessment and local knowledge, individual holons decide over their actions. On the other hand, holons are expected also to cooperate, i.e., to coordinate their actions in order to meet global goals and to satisfy system-wide constraints. The system is open and no explicit control structure is imposed on it, but thanks to cooperation, the members may form also temporal federations. Thus, a holonic organization tries to combine the responsiveness and robustness of decentralized, network-like organizations, and the stability and efficiency of hierarchical control architectures. A parallel trend was making some explored mechanisms of Nature operational in the organization and control of manufacturing systems. This approach resulted in the so-called biological manufacturing systems [35] [58] [59] and later on, led to the engineering's concept of emergence [28].

5 APPLICATION DOMAINS IN MANUFACTURING

In this section we present characteristic applications of agent technology in the main manufacturing domains. The survey below is far from being exhaustive; it focuses on pioneering works, as well as on developments where an agents-based approach seems to be inevitable for meeting the requirements of modern manufacturing.

5.1 Engineering design

In the past decade, both the scale and the scope of engineering design had been changed and much enlarged. Design activities in various branches of engineering (mechanical, electrical, control, etc.) are now being **integrated**. Furthermore, acknowledging that engineering design must take into account the intrinsic requirements and properties of the processes that bring to life, create, maintain and re-cycle artifacts, **concurrent engineering** includes all the main life-cycle activities such as marketing, design, manufacturing, distribution, sales, operation, maintenance, disposal and re-cycling. Participants in the product life-cycle can interact in parallel. **Collaborative engineering** (or design) transcends the above approaches: it emphasizes interaction instead of iteration, makes the conflicts between different stakeholders in the design process explicit and strives to achieve acceptable trade-offs via negotiation [60] [61] [62] [63]. Negotiation is extended towards customers in a framework presented by [64] where design specifications of customized products are developed in the course of an iterative give-and-take process.

The above developments posed new requirements for the computational support of design. Decomposition and parallel execution in collaborative design, naturally, lend themselves to an agent-based approach. Beyond the usual advantages of having a modular system structure, additional merits are as follows: the agents need not be co-located, and they can form wrappers around and provide interface for existing legacy systems (e.g., analytical tools, simulators, CAD systems from various fields of engineering). Agents embodying knowledge and encapsulating tools of different engineering domains can communicate and work together if they have a set of shared concepts and terminology, a common language for expressing this knowledge, and a communication and control protocol for requesting information and services. An early infrastructure like this was demonstrated through the PACT system [65] which was built upon the early results of **ontology** design and **knowledge interchange**, and set a quasi-standard for a knowledge query and manipulation language. Since then, KQML, a typed message language with application-independent semantics and protocol has become widely popular in other MAS applications [66]. Ontologies also provided the basis for a knowledge-intensive design approach where relationships of different models were handled by a so-called **metamodel** mechanism [67] [68]. For instance, when designing mechatronic artifacts, different physical phenomena related to geometry and kinematics, force and distortion, electricity, friction, sound and heat have been represented by different, but re-usable and sharable aspect models. Interrelations of various kinds (causality, abstraction, approximation, aggregation, etc.) among the models were maintained by the metamodel. Recent approaches suggest solutions for situations where a priori models do not have sufficient representation power [69]. E.g., in [70], an agent system supporting the design of intelligent machining centres is proposed that concerns also the thermal behaviour of machines. Notwithstanding the general assumptions and layered architecture, it was very difficult to transfer this custom-tailored system to another application.

All the agent-based approaches to design are similar in that they represent partial, dynamically evolving design objects (beyond the well-proven object-oriented representation) means of **constraints**. Constraints are declarative, non-directional, additive, mutually dependent and can express partial information. Hence, they state clearly what has to be satisfied without stating how. Constraints are suitable for maintaining distributed, locally incomplete representations of design objects as well. That is the reason why constraints provide a key technology for multi-agent design.

A couple of design specialists may work in a fixed organization [65], or, alternatively, even thousands of agents, each responsible for a particular constraint, may delimit the design space of the feasible solutions. In the wide spectrum of possible organizations, knowledge-intensive multi-agent design systems tend to apply the organization pattern of fixed hierarchy or standardization of work. For some notable examples, see the descendants of PACT (SHADE [71], First-Link [72]), as well as the Agent-Based Design Concurrent Design Environment (ABCDE) [73], RAPPID [74], and Facilitator [75].

The RAPPID (Responsible Agents for Product Process Integrated Design) system relies on three strategic mechanisms: autonomous agents as a way to distribute decision-making among a community of human beings and computers, market-based control as a mechanism for coordinating distributed decision-making, and set-based design as a means of making decisions in parallel, regarding partial designs. [75] presented a MAS with a facilitator agent, a console agent and some service agents. The facilitator is responsible for the decomposition and dispatching of tasks, and for resolving conflicts of poor designs. The console agent acts as an interface between designers and the system. Each service agent is used for modeling different product development phases.

One expects that multi-agent design cannot be easier than single-agent design. In principle, it may be so, however, in a MAS design problems and the available knowledge can be structured in novel ways. Collaborative and life-cycle approaches to design are successful in particular because they are utterly based on **interaction**. Interaction helps to harness external knowledge that could not have been captured and internally represented. Furthermore, collaborative engineering (CE) makes explicit the disparate goals, objectives, priorities and concerns – in short, interests – behind the various activities related to a product's life-cycle. These interests manifest themselves as conflicts just in the early phase of design when decisions with far-reaching effects are made. The well-known merits of CE are due to the early detection and negotiation-based resolution of such conflicts. Negotiation over conflicts can drive the design process towards innovative solutions [62]. From a wider perspective, one can see CE as trading incomplete knowledge against interest. A trade like this can be very fruitful: rational, interest-seeking behavior on the part of autonomous fagents can result in successful overall performance in cases when the agents have limited capabilities (knowledge and/or resources). At the same time, collaboration rests upon interaction, which is still the key to creative design.

5.2 Process planning

Process planning is aimed at creating plans for discrete manufacturing operations that are executable in resource-constrained production environments and produce the designed artifacts. Hence, computer-aided process planning (CAPP) incorporates both design and production related concepts: geometry, tolerances, surface quality, material properties, manufacturing processes, machines, tools and holding equipment (fixtures, grippers and robots). Useful domain knowledge varies also with the actual technologies such as machining, sheet metal bending, inspection, or (dis)assembly. There are two usual ways of handing the complexity of CAPP problems:

- De-structuring its world into manageable micro-worlds – these are the so-called features.
- Decomposing the planning problem into sub-problems such as process and resource selection, setup planning, sequencing, path planning, and NC programming.

The idea of a planning engine that could synthesize the results of several domain-specific agents appeared in [65] and [76]. This process planner worked with a kind of hierarchical task network following the standardization-of-work coordination pattern. A similar solution strategy with sophisticated geometric reasoning appeared in [77].

Human interaction is emphasized in [78] where specific agents, working on the exact geometrical representation of the part, available machines and tools, build up the space of the process plans. Then, an optimization engine extracts solutions from this space. In fact, full-fledged CAPP systems are concurrent engineering systems (see the above section) with a special emphasis on process modeling. However, making a consistent synthesis out of results generated on partial models proved to be problematic. The crux of all but the simplest CAPP problems is to reconcile the logical and optimization aspects of planning and to handle inconsistent pieces of design and technology related knowledge. The resolution of conflicts like this calls for **negotiation** – no wonder that these issues of CAPP and concurrent engineering prompted a novel socio-technical concept that considers **engineering** as **collaborative negotiation** (ECN) [61] [79] [62]. Alternatively, in this ill-defined area one can rely on methodologies of **emergent synthesis** [28] [80]. The sharing of tasks in multi-agent process planners followed some traditional decomposition of the CAPP problem, with sub-problems calling for knowledge-intensive solutions. Hence, their characteristic organization patterns are fixed hierarchy and standardization of work.

In recent works, CAPP has been extended towards the execution of plans: planning is aware of available resources and takes actual lot-sizes or due dates into consideration. On the other hand, process planning knowledge is used for short-term scheduling decisions at the shop floor. Multi-agent system concepts are particularly well suited to this integration [80] [81]. For further details, see Section 5.6.

Finally, note that plans are blueprints of behavior composed of pre-defined actions. Hence, process planning – as a large body of planning problems in general – can be considered a problem of **configuration**. Configurational design and CAPP tend to apply similar representation and solution techniques that enable representing, solving and relaxing **distributed constraint systems**.

5.3 Production planning and resource allocation

Production planning is the process of selecting and sequencing activities so that they should achieve one or more goals of an enterprise and satisfy a set of domain constraints [82]. At the strategic and long-term level, top managers try to allocate available resources based on their experience, intuition and computer support, if available.

Resource-aware aggregate planning is addressed in [80] where a library of generic resource classes is used for describing the enterprise's resources. Resource-awareness means the creation of a dynamic inter-relationship between the planning entities and the enterprise resources, human beings and machines in a distributed manufacturing enterprise.

A market-based negotiation mechanism, called precedence cost tâtonement (P-TÂTO), was reported on in [114]. The system is composed of a project manager agent, task agents, resource manager agents, resource agents, and coordinator agents. The project manager agent maintains the project milestones, the project activity network and each task's resource allocation information. A task agent is in charge of its own single task. A resource manager agent is in charge of monitoring and coordinating a set of resources. A coordinator agent is responsible for coordinating multiple resource allocation markets in the virtual market model. The simulation results based on P-TÂTO indicated high levels of solution quality and computational efficiency.

Sequential allocation of resources can be viewed as a digraph where each vertex represents resources and the arcs represent the allocation [83]. In this method agents are associated with actors, resources and groups of resources. The method has been applied to an automatic-guided vehicle (AGV) system. The resources identified for this system are the segments and points that form the free space world model utilized by the AGVs for navigating. Agents are utilized to manage these resources (and groups of resources) to maintain the flow of AGVs.

The exploitation of agent-based technology in production planning was addressed in the ExPlanTech project [84]. There, one of the 5 intra-enterprise agents introduced is the production planning agent which is in charge of constructing an exhaustive, partially ordered set of tasks to be carried out in order to accomplish the given project.

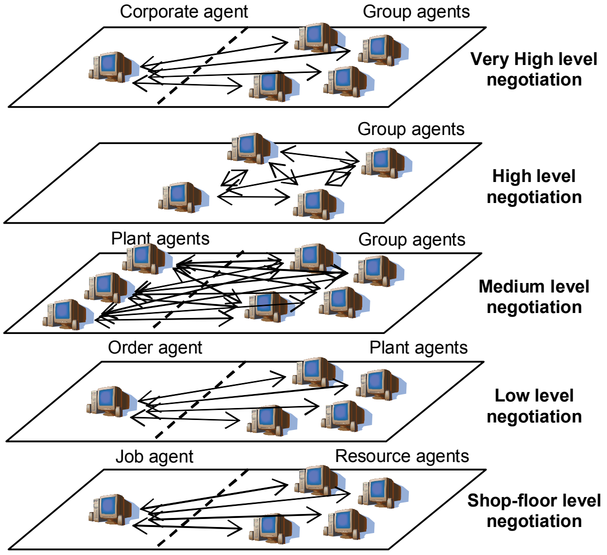

Figure 6: Negotiation models for different planning levels of the enterprise [86].

Five levels of production planning in a reconfigurable enterprise (RE) are distinguished in [85] [86] (see

Figure 6). The authors claim that the successful tools for operation management in RE need to be based on the decentralization of the decisions where each entity in charge of specific planning decisions makes its own decision autonomously, while global planning decisions are achieved by means of coordination and negotiation among them. The various levels differ from each other on the planning horizons, the planning issues, producers and mechanisms. The negotiation model was compared with a centralized solution, showing the benefits of the agent-based approach [86].

Facility layout planning by using self-organizing principles is described in [55] where the planning proceeds by local interactions between machines and AGVs, without global control. Self-organization is exploited in [87] where a decomposition of manufacturing objectives and the allocation of tasks to work systems is arranged. The allocation is arranged by a market mechanism that also makes the optimization of the manufacturing system design possible, by evaluating and selecting from among alternative, competitive work systems.

5.4 Production scheduling and control

Scheduling is the process of selecting from among alternative plans and assigning resources and times to the activities in the plan. These assignments must obey a set of rules or constraints that reflect the temporal relationships between activities, the production technology and, the capacity limitations of shared resources [88]. Manufacturing control relates to strategies and algorithms for operating a manufacturing plant, taking both the present and past observed states of the manufacturing plant, as well as the demand from the market into account.

The manufacturing control problem can be considered at two levels. At the low-level, the individual manufacturing resources are to be controlled to perform actual processes expected by the high-level control functions. High-level manufacturing control is concerned with coordinating the available manufacturing resources, in order to make the products required. In agent-based manufacturing systems, agent technology is usually applied to high-level manufacturing control, but can also be applied in the lower level [89].

There exist some survey papers on agent-based production scheduling and control [90] [5], therefore, here we concentrate on the most important aspects.

Collaborative coordination control (CCC) for a multi-machine workstation is proposed in [91]. Shorter lead times and associated higher production rate, reduction in waiting time of parts for service and lower variable costs are reported on. Agents dedicated to work centers select dynamically the most suitable dispatching rules in an agent-based dynamic scheduling system [92].

An agent-based collaborative production control framework capable of conducting scheduling and dispatching functions among production entities is reported on in [93]. The system was tested on the simulation model of a real-world multi-line elevator manufacturing line.

A multi-agent software system RIDER (Real-Time Decision Making in Manufacturing) has been developed for a cable producing company and for a carpet manufacturer [94]. Upon the occurrence of an event (machine breakdown, new orders, etc.), the agents use a mechanism for generating local alternatives and follow a message exchange procedure to build decision trees, which are traversed and evaluated via user defined cost-based objective functions. Each agent is programmed to perform a specific set of complex actions: as an example, the agent supervising the weaving department of the carpet manufacturer [95] may address both nesting and scheduling optimization problems in the weaving process and then inform the upstream agents about the alternative nesting schedules it has generated. The real-time information required for monitoring the system status and for generating valid alternatives is obtained through a data exchange mechanism, incorporated in the RIDER system, in order to communicate with other information systems and applications.

Centralized/hierarchical job shop schedulers tend to be complex and hardly applicable in real-time, dynamically changing circumstances. On the contrary, heterarchical architectures have a number of inherent advantages, such as modularity, reconfigurability, adaptability, fault tolerance, and extensibility [10] [12] [27] [96] [97]. However, the elements of architectures like this are distributed, usually have no access to global information and, therefore, optima cannot be guaranteed. In addition, such systems can become unstable from the computational point of view.

Hybrid frameworks such as described in [98] have attempted to combine features of both hierarchical and heterarchical frameworks. In [99] a hybrid, agent-based scheduling and control system architecture is presented for solving the order picking problem in an industrial warehouse. The goods are stored at multiple locations and the pick location of goods can be selected dynamically in near real-time. The higher level optimizer agent, with a global perspective, generates a balanced and synchronized order-tray sequence and efficiently assigns resources to each order tray, using a genetic algorithm (GA). The middle-level guide agent takes the resource assignment decision from the higher level agent and guides the lower level agents to achieve improved system performance. The lower level agents make their decisions based on real-time conditions, and suggest the alteration of predetermined resource assignments but have to obtain permission from the middle-level agent.

Holonic control structures

Holonic systems, as one of the new paradigms of manufacturing, consist of autonomous, intelligent, flexible, distributed, cooperative agents or holons [31]. One of the most promising features of the holonic approach is that it represents a transition between fully hierarchical and heterarchical systems [31] [98]. Holonic architectures can also include both temporal and permanent hierarchies [98].

Three types of basic holons, namely, resource holons, product holons and order holons, together with the main information flows among them are defined in the **PROSA reference architecture** [32] (Figure 7).

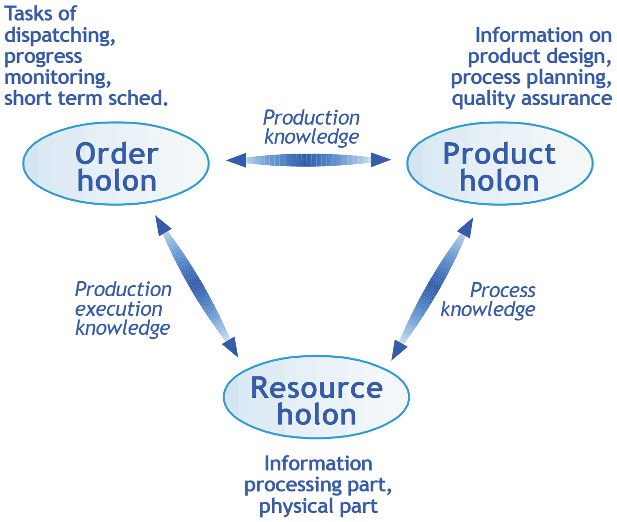
Tasks of dispatching, progress monitoring, short term sched.
Information on product design, process planning, quality assurance
Production knowledge
Order holon
Product holon
Production execution knowledge
Process knowledge
Resource holon
Information processing part, physical part

Figure 7: The PROSA reference architecture [32].

These basic entities are structured using object-oriented concepts such as aggregation, specialization. Staff holons are also foreseen to assist the basic holons in performing their work. The ADACOR architecture defines a similar model, with a supervisor holon that introduces coordination, group formation and global optimization into the decentralized control. The primary objective is to increase the agility and re-configurability of the production system [100]. Other authors refer only to two types of basic building blocks, such as order and machine agents in [101], job and resource agents in [102], and order and machine (resource) holons in [103]. A common feature of these approaches is that the functions of the order and product holons are integrated in one basic type.Based on the PROSA architecture, an agile holonic multi-cell control system (HoMuCS) was described in [104]. The holonic behavior of machine tools' CNC was highlighted in [105].

Market-based approaches

Cooperation and conflict resolution are the main issues in agent-based scheduling systems. For solving these problems, the idea of **negotiated factory scheduling** appeared some time ago. Early attempts implemented dispatching mechanisms [107] [108] but did not concern advance scheduling. Conversely, other works realized predictive schedulers with no reactive capabilities. For negotiation, versions of the contract net protocol [109] were used. Market-oriented programming based on general equilibrium theory has been first applied to resource allocation problems with no temporal aspect [109]. Dynamic reconfigurability, which asserts itself again in the holonic concept, appeared first in [108] that suggested iterated bidding for selecting cells that complete a job. This approach was developed further into random manufacturing [34] where temporal coalitions of machines competed for incoming orders.

Nowadays, negotiation based algorithms have been used almost without exception. In this way, schedule generation is a recursive, iterative process with announce-bid-award cycles where market mechanisms [34] [102] [106] can be advantageously used. Order (or part) driven and resource (machine, cell) driven techniques may be distinguished, based on who makes the announcements. The extent of coordination can be limited to given (e.g. pairs of) agents. Tasks are, usually, announced one by one, mostly after the preceding operation has been completed which causes **decision myopia**, a common drawback of distributed approaches. More advanced systems support look ahead scheduling with a longer, sometimes varying horizon.

In **multi-agent scheduling**, agents manipulate resource and/or order variables under their own authority. Potential job interactions are handled by agents telling each other their aggregate workloads and free capacities [110] [111]. Since global consistency can hardly be guaranteed, constraint checking or simulation is needed. Since constraint relaxation and backtracking may require enormous computational overhead, their use is limited. As an alternative to backtracking, heuristic tinkering of almost conflict-free schedules can be applied.

In [106] and [112] a market mechanism was proposed which made cooperation in the control of distributed manufacturing systems possible. Particularly, an integrated production planning and distributed scheduling problem was investigated. The system included self-interested, autonomous agents whose goal was to maximize their own profit. While the production planning problem was handled by a central management agent, machine agents were responsible for building up and executing local schedules at machines with different technological capabilities. Congestion control (via order selection) was the responsibility of the management alone, but scheduling decisions were reached by a partially committing bidding mechanism between the management agent and the machine agents. The bidding protocol is presented in Figure 8.

A market mechanism has been applied to solve dynamic multi-project resource-constrained scheduling problems: [113] [114] handle the decentralized scheduling of resources, which are shared by multiple projects.

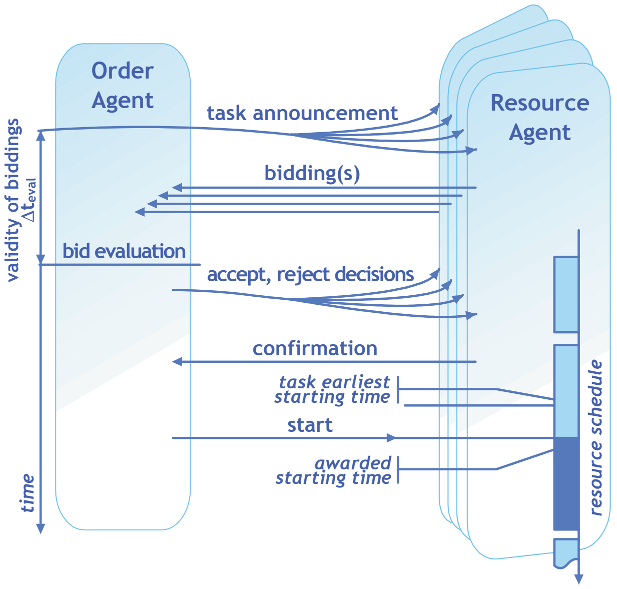

Figure 8: Agents and their relations according to [106].

In a virtual economy where agents act as buyers and sellers of resources, "resource-time slots" (i.e., availability over a given time) are traded as goods. Due to the dynamic and distributed nature of the economy, the approach achieved higher levels of flexibility and scalability. In the AARIA (Autonomous Agents for Rock Island Arsenal) system unit process, resource, manager, part, customer, and supplier are identified as agent classes [115]. A collaborative control system for mass customization manufacturing is described in [102].

A market mechanism has, however, some drawbacks: since it is a nondeterministic, utility-based method, the system's worst case behavior and even the avoidance of extreme situations are hard to guarantee. No more than the average behavior can be predicted; the significance of the results should be based on careful simulation experiments. Moreover, it is doubtful whether small artificial markets with a low number of participants and limited number of encounters could produce the effects we meet in real markets of large size. It is hard to define where the borderlines of a model like this should lay. Note that in the real world, production elements with monetary terms do have counterparts in consumption. Money that can buy nothing is good for nothing.

Stigmergy-based coordination and control

A relative novel approach for coordination in multi-agent systems is **stigmergy** which belongs to mechanisms which mimic animal-animal interactions [45]. Stigmergy is an indirect coordination tool within an insect society where parts of global information is made available locally by pheromones, e.g., as in the case of ant colonies. This way, individual agents are not exposed to the complexity and dynamics of the situation, and the communication burden in the computer realization is significantly lower than in, e.g., market-based solutions.

A holonic manufacturing execution system (MES) is presented in [45] that preserves the advantages of heterarchical designs and predicts the near future, while accounting for changes and disturbances. In order to achieve this, the software agents reflect the underlying manufacturing system (e.g., order agents reflect tasks) and delegate mobile (ant) agents consistently. For instance, exploring ant agents query, the associated resource agent about processing times; they do not have their own model of the resource and, therefore, make no assumptions that can be faulty. The forget-and-refresh mechanism of the stigmergy infrastructure ensures that information remains up-to-date.

As for the realization of stigmergy-based systems, virtual ants can be realized by mobile software agents. A message-based realization can be conceived, as well. Market- and stigmergy-based approaches do not represent two totally different ways of multi-agent coordination and control: they can be nicely combined in complex societies.

Adaptation and learning in multi-agent production control

Early investigations implied that the efficiency and quality of multi-agent scheduling depended on variable, adaptive commitment strategies. This is especially so in dynamic domains where commitment may hinder an agent from responding to new contingencies. The main issues are when, what and to what a degree to commit.

In [116] an agent-based system is described which is used for resource allocation and production control. Agents assigned to all orders and resources are adaptively conditioned according to the logistic situation. Simulation experiments proved that the approach delivered nearly the same logistic results as established ERP methods did. However, the proposed agent-based system reacted on disturbances and unexpected events better than other ERP systems.

Learning and other forms of adaptation are essential in multi-agent systems [25] [117] and can be categorized as follows:

- **Centralized learning** (or isolated learning) refers to learning approaches which are entirely executed by single agents, with no regard to interaction with other agents.
- **Decentralized learning** (or interactive learning) involves several agents which require a joint and coordinated interaction among them.

The adaptation procedure described in [117] and [118] is a centralized approach in which each resource agent locally adapts its behavior in order to achieve a more profitable position in the agent society. The feedbacks are represented by changes in local utilization parameters and bid awarding and/or rejection reactions issued by the order agent. Each resource agent incorporates a rule-base by which it can locally decide on the cost factor to be applied in a task announcement. The preconditions of these rules are the utilization of the resource and the ratio between the won and lost bids which are stored locally for each agent in the table of machine abilities and history.

Simulation results demonstrate that the major advantage of the above solution is a more balanced usage of resources. Moreover, several performance measures, such as maximum tardiness and makespan proved to be better with cost factor adaptation [117].

In [119] a market-based distributed production control system with learning and cooperative agents is described. The adaptive scheduling is done by a triple-level learning mechanism. The top level of learning consists of a simulated annealing algorithm, the middle (and the most important) level contains a reinforcement learning (RL) [120] system, while the bottom level is done by numerical function approximators, such as artificial neural networks [121]. The proposed neurodynamics-based system can be used for solving the general dynamic job-shop scheduling problem in a distributed, iterative and robust way. The time and space complexities of the solution are compared with those of classical approaches.

Finally, note that adaptive agents represent one of the most promising approaches towards Class II and Class III problems of emergent synthesis, i.e., problems with incomplete environment description [56] [59] [122].

5.5 Process control, monitoring and diagnosis

Process control, monitoring and diagnosis are closely related, partly overlapping fields. Monitoring involves observing, recording, and processing signals, and detecting abnormal conditions of the controlled process. Diagnosis is the process of generating plausible hypotheses on the causes that led to the current (abnormal) state of a system. Prognosis refers to predicting when a particular state will occur next or what state the system will reach at a specified future time.

Monitoring, diagnosis and prognosis apply both to physical processes (e.g., at the machine level) and business processes (e.g., material and work flows). Signals in the physical process correspond to process parameters such as force, vibration, temperature, pressure. For business processes, material movements (e.g., from one location to another one), process completion times, and other transactional data associated with information or material flow serve as the signals. The frequency and range of physical process signals are much higher. While automatic (feedback) control is quite common for physical processes, business processes will likely require human intervention. Despite these seemingly wide differences, the scientific principles and even the techniques underlying monitoring, diagnosis and prognosis are similar in both the physical and business settings.

The application of pattern recognition (PR) techniques, expert systems (ESs), artificial neural networks (ANNs), fuzzy systems (FSs) and nowadays hybrid artificial intelligence (AI) techniques in manufacturing can be regarded as some consecutive elements of a process started more than two decades ago [123]. Extensive background on monitoring can be found, e.g., in [124] [125] [126] [127] and [128].

In the CIRP-survey on developments and trends in control and monitoring of machining processes, the importance of **sensor integration/fusion**, sophisticated models, multimodel systems, and learning ability was also outlined [129]. The development of user-programmable, multipurpose, modular monitoring systems in manufacturing started in the early eighties [130] [131]. In these systems the main goals of modularity were to accomplish sensor integration and achieve appropriate computing power.

Modularity can be regarded as a step towards agent-based systems. In [132] a multi-agent approach was presented for the selection of grinding conditions. The agentification was made according to the different reasoning approaches used, i.e., case-based reasoning for selecting the combinations of the grinding wheel and values of control parameters, rule-based reasoning for cases where no data were available in the case-base, and neural networks for selecting the grinding wheel if required. The final decision was made by the operator.

In [133] an approach for the modeling, monitoring and optimization of manufacturing processes and process chains was introduced. A significant feature of the proposed technique was that it could handle models of different types at the same time.

By taking an agent-based approach, sub-systems and sub-system components are mapped to agents and agent organizations; interaction between sub-systems and sub-system components are mapped to cooperation, coordination and negotiation mechanisms; and relations between them are mapped to explicit mechanisms for representing organizational relationships [134].

The distributed architecture of the MAGIC multi-agent diagnostic system is based on the Multi-Agents-Multi-Level (MAML) concept as depicted in Figure 9. The idea is that the tasks of the complex embedded system's diagnosis and operator support are distributed over a number of intelligent agents which perform their individual tasks nearly autonomously and communicate via the MAGIC architecture [135]. An architecture like this can be readily distributed on and adapted to existing monitoring and control systems of large-scale plants.

The holonic control system described in [136] specifies a multi-robot control for surface treatment, especially for shot-blasting. The planning activities of the robots are supported by product models. Task-sharing for the robots proceeds through blackboard or contract net based negotiation, including task allocation, detailed motion planning (in time and space). The robots support each other by providing sensor information to locate work pieces and by sharing the workpiece to gradually generate the motion plans for the shot blasting paths.

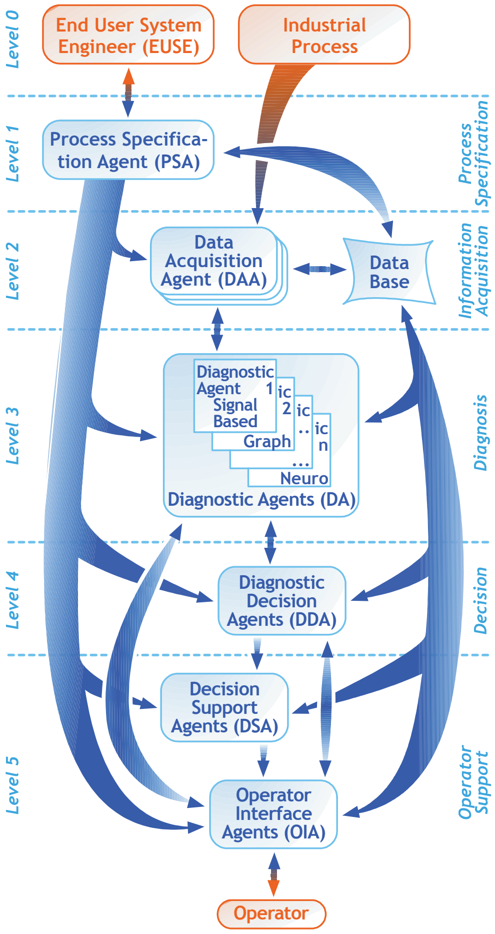

Figure 9: Concept of the MAGIC multi-agent-based diagnostic system [135].

An agent-based diagnostic system was developed for a PLC-controlled low-volume assembly line for an Australian automotive manufacturer [137]. The line consists of three assembly stations linked by a transfer line. The holonic, model-based diagnostic system represents a holarchy of four diagnostic holons representing the line and the three stations, respectively. The on-line test demonstrated the applicability of the approach, by giving 95% fault coverage and 20-40 s/fault execution time.

IT systems (“digital factory tools”) on the operating levels, usually, are not yet integrated and thus, they support separate tasks such as production order control, production monitoring, sequence planning, vehicle identification, quality management, maintenance management, material control and others [138]. An agent-based production monitoring system (ProVis.Agent) was developed for a leading German car factory for monitoring and controlling the body, paint and assembly shops, either from a central control room, or from decentralized control panels on the shop floor [138]. System modules were wrapped and enabled to communicate via a standard agent platform. Later, it was connected to an automatic object identification and localization system, closing the gap between production monitoring and sequence setup [139].

The Multi-Agent Tool Management System (MATMS) for automatic tool procurement in a supply network was introduced in [140]. MATMS operates in the framework of a negotiation-based, multiple-supplier network where a turbine blade producer requires dressing jobs on worn-out grinding wheels from external tool manufacturers. An important element of MATMS is the Dressing Time Prediction Agent for predicting grinding wheel dressing cycle times founded on historical data, under both supplier independent and dependent bases.

Modern communication technologies have made dramatic impacts on remote monitoring and maintenance. The e-maintenance platform named POMAESS (Problem-Oriented Multi-Agent-based E-Service System) interconnects separated (reactive and cognitive) agents for collaborating in an open and time-constrained environment [141]. The platform supports the cooperation of different problem solving experts (human or autonomous machine) such as production management expert, maintenance expert and control expert. An important feature of the developed system is the integration of case-based reasoning. A so-called watchdog agent is described in [142] which - in a remote monitoring and e-maintenance setting - provides continuous monitoring and prognostics of asset degradation and also evaluates asset performance.

5.6 Enterprise organization and integration

Enterprise integration is aimed at providing an information technology infrastructure for all business, engineering, operational and administrative functions of an enterprise that can be used for information exchange, decision making, coordination and collaboration. The integration efforts, usually, come together with organizational redesign and the re-engineering of business processes both within and between the main functional entities. A summary of practical and research issues is given in [143]. It is a general observation that collaboration among the entities of an enterprise could help increasing the system’s overall flexibility, reliability and performance.

Key ideas related to enterprise organization and integration are enterprise modeling, distributed planning and control, and information system modeling. Primarily due to globalization, the nature of businesses tends to be distributed making modeling, monitoring and control of business processes critical. Applying decentralized agents in enterprise integration permits the local parts of an enterprise to continue operation during temporary lapses in connectivity. Modeling formalisms include Petri nets, finite state machines, holons, and software agents. The most important questions here are:

- how to achieve the appropriate representation of process models,
- how to model various constraints within and between business functions such as marketing, design, planning, manufacturing, … and material supply, and how to use them in order to find the best-of-practice process, and
- how to maintain interdependencies within the network of organizational entities.

Furthermore, distributed modeling and control of extended enterprises need to consider scaling and temporal issues, as well.

In an early work [144] a multi-agent framework was proposed for coordination among the different components of the manufacturing system on system level, process level and decision level. Each agent has been modeled as consisting of three main parts: a knowledge base, a control unit and a functional component. Agents represented both physical and decision making entities: hence lot-sizing decisions, production scheduling and resource assignment could be coordinated.

A number of current approaches for agent-based enterprise organization and integration can be found in the literature. For instance, the principles of the ADACOR adaptive holonic production control architecture are applied in the context of a global value network [145]. The ExPlanTech framework [146] covers the functional dimension: it provides specialized modules to support complex planning, resource allocation and scheduling problems (so-called heavy duty planners). The system has its own ontology and, being implemented on the top of JADE, runs on various platforms. A knowledge-intensive multi-agent cooperative framework for concurrent intelligent design and assembly planning, utilizing a Petri net as a core representation system, is proposed in [147]. The Collaborative Agent System Architecture (CASA) system for multi-plant production planning in Internet-based collaborative enterprises is presented in [148]. This enterprise model includes the related functional components, communication infrastructure and coordination mechanisms. The multi-agent cooperative scheduling (MACS) system maintains a dynamic "envelope" composed of capacity and temporal constraints for delimiting the solution space for various agents of an extended enterprise [149].

A recent work [150] emphasizes the communication and temporal aspects of enterprise integration and, for real-time information exchange and decision making proposes, a web-based multi-agent language is used that is grounded in the Extensible Markup Language (XML) and Java.

One specific and recurrent concern of today’s enterprises is the separation between planning and executing activities, which results in loss of time and information [151]. **Concurrent process planning and scheduling** by means of intelligent software agents are addressed in [152] [153]. There, a layered system architecture is proposed, which is based on the application of cooperative agents for optimizing information logistics between process planning and production control. Product- and production capacity-related information are taken into consideration on the level of centralized process planning. On the other hand, process planning knowledge is used for short-term decentralized scheduling decisions on the shop floor level [153]. Therefore, the planning and scheduling functions on the different levels can be carried out in relative independence. Problems which result from time-delayed return of manufacturing knowledge and capacity data, or other lacks of information flows (e.g., from the use of static process plans) are eliminated (see Figure 10).

Finally, an important practical question is how to integrate agent-based solutions into existing systems. In [154] approaches to agentification of resources or even whole manufacturing systems were introduced partly as an extension of the Virtual Manufacturing (VM) concept [16]. The ability of dealing with so-called legacy software packages in a complex system is of high importance. Agents can provide a **wrapper** both for legacy systems and human participants, as shown in [101].

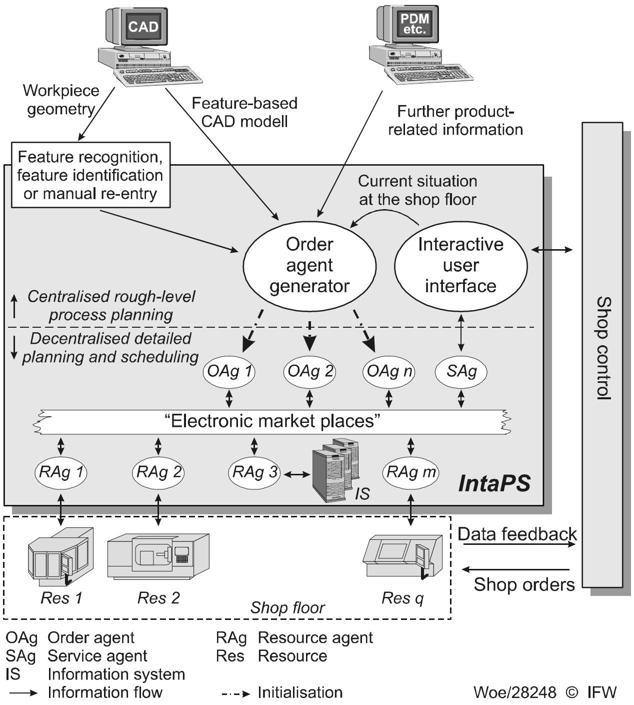

Figure 10: Electronic marketplaces for cooperative agents [153].

5.7 Production in networks

A **production** (or **supply**) **network** is a net of suppliers, factories, warehouses, distribution centers and retailers through which raw materials are acquired, transformed and delivered to customers. In a supply network, the traditional boundaries of firms are dissolved: decisions on the use of resources should concern both internal and external capacities, and the internal flow of materials should be synchronized with the incoming and outgoing flows. There exist a number of **Supply Chain Management** (SCM) systems for integrating data of all major business functions at the nodes of a supply network, but these systems are rather transactional: they provide technology for information storing, retrieval and sharing, but do not really support decision making [26].

SCM is an approach satisfying the demands of customers for products and services via the integrated management of the flow of materials, information and financial assets between autonomous business partners. Since supply networks are unique and complex, local planning at the nodes is done in many different ways: there is no “one-size-fits-all” solution. Instead, there is a need for a portfolio of coordination mechanisms where relations between the partners can be represented on a range of “temperatures”: from cold (competitive auctions, single business relations), through warm (cooperative planning), to hot (full integration). Basically, there are two types of research in applying multi-agents to supply chain management:

- The general approach handles SCM as a problem of designing and operating a multi-agent system.
- Specific problems in the supply chain management, such as collaborative inventory management [155], bidding decision [156], and material handling and inventory planning in warehouses [157] are solved by borrowing some MAS concepts and methods.

The majority of the literature has been focusing on the general application of agent-based supply chain management [96] [158] [159] [160] [161] [162] [163].

In [157] the AWAS (Agent-based model for Warehouse system) systems was proposed. AWAS consists of three sub-systems: an agent-based communication system (ACS), an agent-based material handling system (AMATH), and an agent-based inventory planning and control system (AIPCON). These sub-systems are designed to cooperate together to facilitate just-in-time exchange of orders and materials. Under these subsystems, seven kinds of basic agents are defined including customer, supplier, order, inventory, product, supplier-order, and automatic-guided vehicle (AGV) agents.

In [96] the problems based on a representation of obliged and forbidden behavior among agents were analyzed in an organizational framework, together with an inference method that decides which obligations to break in conflicting situations. These are integrated into an operational, practically useful agent development language. The major strength of the approach is the way it supports coordination by exchanging constraints about obliged and forbidden behavior among agents.

Issues and solutions for the construction of an agent-oriented software architecture for managing the supply chain on the tactical and operational level are presented in [158]. Among other things, a communication protocol for coordination between supply chain agents is proposed. The approach relies on the use of an agent building shell (ABS), providing generic, reusable, and guaranteed behavior components and services for speech act based communication, conversational coordination, and role-based organization modeling. The ABS is a collection of reusable software components and interfaces that support application-independent agent services. Using these services, developers can build on a high-level infrastructure, whose abstractions provide a conceptual framework that helps in designing and understanding agent systems, eliminate work duplication, and offer guarantees about the services provided by the tool. For illustration, six types of agents are used: order acquisition, logistics, transportation, scheduling, resource, as well as dispatching agents.

A framework for computer support of supply chain management in chemical and process industries by using JATLite (Java agent template, lite) and the standard knowledge query message language (KQML) was presented in [159]. The architecture of the multi-agent supply chain support system is shown in Figure 11. The system is defined through retailer, logistics, warehouse, plant, purchasing and raw material supplier agents.

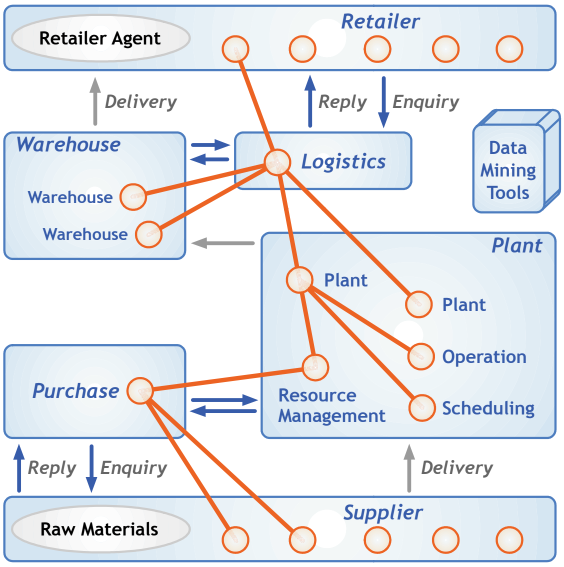

Figure 11: The multi-agent-based supply chain [159].

The integration of partners who use heterogeneous information systems was discussed in [164]. To support the communication of different partners and enable the information flow within the value added chain, XML neutral data format was used. Data are exchanged among the companies via a multi-agent software system that handles the execution of the business process. It is combined with a generic hierarchical planning model, to improve the performance of the supply chain utilizing the production resources, while in parallel it enables the user to customize the resource allocation method via the selection from a set of dispatching algorithms.

Additionally, the application of web-based software technologies for supporting the planning and monitoring a ship repair contract was presented in [165]. The discussion was based on the transactions among the ship-owner, the shipyard, the material suppliers and the shipyard's subcontractors in the maritime supply chain. Each company was represented by a set of software modules implementing an 'agent-like' behavior and controlling the execution of critical supply chain functions. A business model was implemented in a software package, representing a cost effective, simple to configure and use, platform independent, integrated environment. The system also allows small enterprises with moderate information technology infrastructure to participate in the supply network. The adaptation of an approach like this is feasible and reduces communication efforts, while it improves the communication among the cooperating companies in a supply chain contract.

A Multi-Agent Tool Management System (MATMS) for automatic tool procurement in a supply network was introduced in [166]. MATMS has been developed for a multiple-supplier network where a turbine blade producer (customer) requires from external tool manufacturers (suppliers) the purchase of new grinding wheels and the performance of dressing operations on worn-out grinding wheels for turbine blade fabrication. The system has a layered architecture [167]: the Supplier Network Level which comprises of supplier type agents is responsible for carrying out the actual dressing jobs. The Enterprise Level coordinates MATMS activities to achieve the best possible results in terms of on-time delivery, minimal inventories and costs. Agents on the Plant Level represent production lines of the customer factory.

The unified framework for supply chain decision support system proposed in [160] integrates the various elements of a supply chain such as enterprises, their production processes, and their associated business data and knowledge and represents them in a unified, object-oriented way. Supply chain elements are classified as entities, flows and relationships. Software agents are used to emulate the entities i.e., various enterprises and their internal departments. Flows of material and information are modeled as objects.

A **virtual market** solves the allocation of resources in a dynamic environment as described in [161]. Efficient coordination among agents is very important for realizing the **agility** of supply chains. Typically, there are two types of agents in the architecture of multi-agent-based agile supply chains: **functional** agents and **mediator** agents. Functional agents perform planning and/or controlling activities in the supply chains, while mediators play a system coordinator role by prompting cooperation among agents and providing message transfer services [162]. This organization is realized according to the standardization of work or skill pattern (see Section 4.3).

Negotiation via constraint evaluation is defined to support supplier selection in a hybrid supply chain [168]. Strategies are defined via constraint logic programming predicates for proposal request, generation, evaluation, and counter-proposal generation, as well as for order negotiation. The approach has been implemented as a set of negotiating processes operating across the internet.

The use of **mobile agents** in a Brazilian automobile supply network was presented by [163]. The main motivation of this work was to design and implement an environment conductive for flexibility, adaptiveness and efficiency. The system uses the Kernel Language for Agent Interaction and Mobility (KLAIN), a language that realizes a programming paradigm where processes can be transported across different computing environments.

Members of a supply network may take part in **cooperative planning** [169]. In this case, partners have a definite incentive and commitment to cooperate and share both their risks and benefits. The main driver for cooperation is uncertainty, which has its roots in market demand, manufacturing and supply. Uncertainty can be managed only if powerful planner systems fill in the various planner roles locally. Plans which are really executable and cost efficient make the future - even market demand - more predictable, and the actual production more profitable. Novel information channels have to be established between the partners so as to share the results of local planning, from detailed production schedules up to demand plans, on all the horizons and levels.

Finally, in a broader setting, the phenomena of emergence of products and their underlying supply networks are demonstrated in [170] by simulation, showing the potential of emergent synthesis in this complex field.

One of the recurrent issues related to supply chains is how to handle complexity. There has been a renewed interest in this problem, and now, multi-agent modeling and analysis is providing new approaches. We discuss related issues in the strategic directions section.

5.8 Assembly and life-cycle management

Assembly, usually, represents the last technical step in the product creation process; however, according to the up-to-date organization principles, the ready-made products are to be followed throughout their life cycles. Hierarchical, heterarchical and holonic control structures for an assembly cell are compared in [31]. Holonic systems were found to deliver better performance in a wider range of situations than their more conventional counterparts. For instance, the holonic concept demonstrated improvements in the robustness and volume flexibility in an engine assembly system in the automotive industry [172].

The concept of “plug & produce” was introduced in [173] as analogy of “plug & play” in computer world. The viability of the approach was demonstrated by using a holonic assembly cell installed by the authors.

The value-adding chain of a modern enterprise is a worldwide network of suppliers, transport agencies, manufacturers, retailers, distributors and recyclers. In [171] an agent-based approach is presented which integrates the supplier and transportation agency into assembly control. The objective of agent-based control is to avoid large lead times and capital-intensive stock, and to make the assembly process as refractory as possible to disturbances.

The agent society, organized on three levels, is shown in Figure 12. The first level incorporates the negotiation agents: Assembly Scheduler (AS), Assembly Transportation Scheduler (ATS), Transportation Scheduler (TS), the Supplier Scheduler (SS) and the Master Assembly Scheduler (MAS). The example given has only one MAS, ATS and TS, four SS and AS. The second level belongs to the Carrier Agent (C) and the Assembly Agent (A). The latter tries to ensure assembling of the right parts. The Carrier Agent supervises the complete tour of delivery. On the third level the stock agents represented by the Supplier Buffer (SB) and the Assembly Buffer (AB) pass information on the actual inventory levels to agents requesting them [171]. The authors outline that by using the proposed architecture the network became more transparent and can be operated with smaller inventory.

The life of the product holons is not ended with the accomplishing of the manufacturing process, rather, it exists until the product is disposed of. According to this idea, the product holon proposed in [174], besides containing data concerning the information on the product life cycle, user requirements, design, process plans, bill of materials, quality assurance procedures, and the process and product knowledge, also contains the information related to its end-of-life.

A framework based on autonomous, cooperative agents for life-cycle oriented data support is presented in [175]. The framework identifies internal and external data sources for product optimization. Depending on their types, the data are stored and permanently updated in different decentralized locations (web, company and machine). Initiated by the product manufacturer, the databases will be permanently updated with changes or experiences from current operations. The data can be acquired from machine control, Auto-ID or other sources.

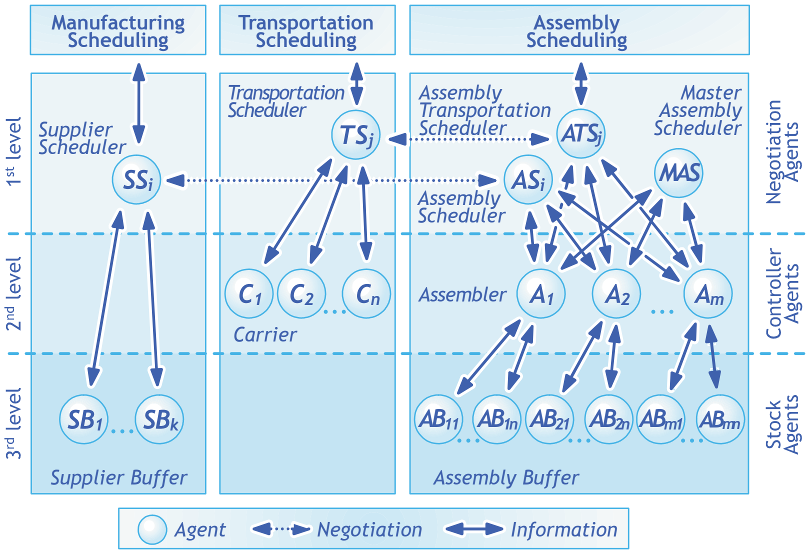

Figure 12: Society of agents for assembly control [171].

6 USE OF AGENTS

Agent technology and multi-agent systems have become prevalent in the past decade, enabled by a wide spectrum of information and control technology (such as networking, software engineering, distributed and concurrent systems, mobile technology, electronics commerce, interfaces, semantic web). As reviewed above, developments like this found their way to application and deployment in all fields of manufacturing. How could the idea of agents become so powerful? Below, also taking the perspectives of [1] into consideration, some explanations are given.

6.1 Agents as a design metaphor

As computer systems become ever more complex, we also need more powerful abstractions and metaphors to design and explain their operation. The concepts of agents enables us to structure our knowledge (and system design, accordingly) around components that have autonomy and capability to communicate. This decomposition may happen both along functional or physical dimensions. Objects that earlier had complex properties can now be personified. We can demarcate agents from the environment, and then attach intentional notions such as goal, objective, belief, intention and plan to them. This so-called **intentional stance** is an abstraction that helps us in a useful and familiar way to describe, explain, and predict the behavior of complex systems [176]. Note that most important developments in computing have been based on new abstractions, such as procedures, abstract data types, objects. Now, agents as intentional systems represent a further and increasingly powerful abstraction.

Any kind of intelligence requires the handling of **conflicts** rooted in disparate interests. Efficient operation, adaptability, and occasionally even the survival of a complex system hinges on whether it is able to resolve conflict situations. The conflicts and their resolutions constitute the base of the so-called **social knowledge** – a kind of knowledge which emerges from the collective action of the individuals and, therefore, belongs to the system as a whole. Instead of trying to eliminate conflicts (which is impossible), conflicts are rather to be exploited. Conflicts become explicit if the system is modeled like a community of self-interested, rational agents. Hence, the agent-based approach forces the system's designer to find ways for managing conflicts.

6.2 Agents as source of software technology

As far as software technology is concerned, agents and MAS provide a wide array of models, techniques, formal modeling approaches and development methodologies that all together shape the general-purpose paradigm of **agent-oriented software engineering** (AOSE) [41] [177]. There have been developed several programming languages and software development environments which not only support MAS programming, but also implement key concepts of MAS in a unified framework. Now consolidated FIPA **standards** are available which are intended to promote the interoperation of heterogeneous agents and the services they represent. The FIPA specifications include Agent Communication Language messages, message exchange interaction protocols, speech act theory-based communicative acts and content language representations [48]. The agent unified modeling language (AUML) [178] and software **platforms** such as the IBM's Java Agent DEvelopment Framework (JADE) [179], IAI's CyebelePro [180], the BDI agent frameworks Jadex or JACK [181] [182] the Zeus Agent Toolkit, and the Cougaar Agent Architecture [183] are all available for the specification and realization of multi-agent systems.

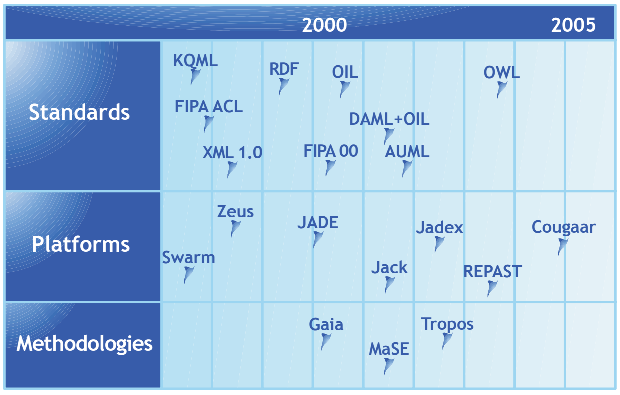

Figure 13: Current standards, platforms and methodologies of agent-oriented software engineering.

These platforms reduce the developmental time of agent-based systems by providing the decomposition and communication infrastructure. For instance, JADE and Cyebele are general-purpose, while Couggar is readily applicable to logistics applications. Once the designer defines the agents and their interconnections, all of these platforms allow the designer to build the agent system with relative ease. However, it must be noted that building agent functionalities is left to the designer. All the above platforms give application protocol interface facilities, making the integration of other programs and systems relatively less cumbersome. The platforms' communication is based on KQML structure and are FIPA compatible. Furthermore, they allow for agent mobility and since they are based on Java, they are independent of the actual hardware/software platforms. There are frameworks directly developed to support simulation, such as SWARM or its successor, the Recursive Porous Agent Simulation Toolkit (REPAST) [184] .

Recent developments have provided **infrastructure** for open agent communities. The DARPA Agent Markup Language (DAML) project was aimed at developing a language and appropriate tools to facilitate semantic interoperability of agents [185], just as parallel efforts of the W3C consortium have resulted in the standards of Extensible Markup Language (XML) [186], and the so-called Resource Description Format (RDF) [187]. These representation technologies set the base for the recently developed Web Ontology Language (OWL) [188].

However, it is a wide-spread view that the design and implementation of agent-based software systems needs not only standards and platforms, but novel **methodologies** that cover the whole life-cycle of system development [189]. There are quite a number of special-purpose AOSE methodologies and some generic ones such as Gaia, Tropos, and Multiagent Systems Engineering (MaSE) [177]. Figure 13 provides a sampling of important developments from the past decade.

6.3 Agent-based simulation

Agent-based modeling is especially suitable for simulating the behavior of complex systems operating in dynamic environments. In contrast to traditional, top-down approaches, the emphasis is on capturing the individual, together with all its limitations (both cognitive and computational) and the interactions of individuals [190]. The question is whether and how local interactions can produce observable - and useful - patterns of global behavior. Hence, **agent-based simulation** (ABS) became an accepted methodology for developing plausible explanations for emergent phenomena. Alternatively, it was used for verifying multi-agent system design in a number of fields from social to natural sciences.

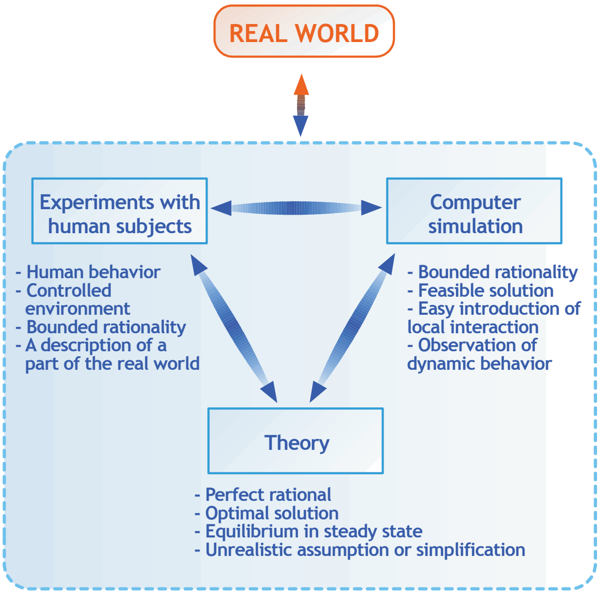
REAL WORLD
Experiments with human subjects
Computer simulation
- Human behavior
- Controlled environment
- Bounded rationality
- A description of a part of the real world
- Bounded rationality
- Feasible solution
- Easy introduction of local interaction
- Observation of dynamic behavior
Theory
- Perfect rational
- Optimal solution
- Equilibrium in steady state
- Unrealistic assumption or simplification

Figure 14: Comparing results of theory, experiments with human subjects and agent-based simulation [192].

Agent-based simulation provides important means for the engineering of social systems by assisting the design and implementation of social policies. Computational and human decision makers alike can participate in ABS experiments. E.g., a study of the institutional design of a recycling society is presented in [191] [192].

6.4 Deployed multi-agent systems in manufacturing

Section 5 provided a review on the applications of agent theory and multi-agent systems in all domains of manufacturing, from engineering design to managing production networks. The rich variety of multi-agent approaches clearly shows the application potential of agent technology. By now, there is a common understanding that various requirements of manufacturing can really be met by autonomous, embodied, communicative and eventually cooperative agents operating in a society.

Still, the number of **deployed** multi-agent systems that are already running in real industrial environments are surprisingly small. For instance, Table 1 summarizes the successful industrial projects (taken from our review) that resulted in deployed, operational systems. Other recent surveys also conclude with the observation that no significant advancement has been made yet in transferring agent technology to industry [6] [193].

This, relatively slow transfer has manifold reasons. First, the introduction of agents, in principle, does not reduce the complexity of problems. Next, interoperability is expensive. Just due to the increased communication overhead, the performance of a MAS can degrade and especially rough-grained systems (consisting of sophisticated agents) can hardly be scaled up [146] [193]. Although the agent metaphor is useful in system design, and there also are several methodologies to support agent-oriented software engineering, industrial-strength support is still missing. Similarly, there are no methods to make the wrapping of legacy systems by agents into an automatic affair [194]. Hence, actual industrial deployments are risky and expensive software engineering endeavors.

In the behavior of a MAS there is always an element of emergence which can be a serious barrier to the practical acceptance of agent-based solutions. Industry needs safeguards against unpredictable behavior and guarantees regarding reliability and operational performance. It is still open how to demonstrate required system properties via robust simulation experiments, and how to complement simulation with formal analysis.

Against all the above difficulties, there is a consensus concerning some application areas where the industrial take-up of agent technology is to be expected even on the short-term [1] [36] [193]:

- Where neither access to information nor decision rights can be centralized. This is the case in managing supply networks, including transportation and material handling.
- In complex operations management (such as resource allocation, planning and scheduling) where the problems can be decomposed along district, typically conflicting goals and performance objectives.
- In industrial monitoring and control where robustness and fast reconfigurability are essential requirements in a distributed setting.

Application	**refs.**
Production planning and resource allocation	
ExPlanTech, planning mass production of car engine manufacturing	[146] [194]
Facility layout planning by using self-organization	[55]
Production scheduling and control	
AARIA (Autonomous Agents for Rock Island Arsenal): scheduling of an army manufacturing facility	[115]
Hot steel rolling agent: dynamic scheduling of steel production	[195]
Warehouse scheduling and control	[99]
Process control, monitoring and diagnosis	
Holonic diagnosis of an automotive assembly line	[137]
Holonic shot-blasting by robot cooperation	[136]
ProVis.Agent: Production monitoring architecture	[138]
Production monitoring connected to object identification / localization	[139]
Production in networks	
Brazilian automobile supply chain network.	[163]
MAS applied for petroleum refinery supply chain	[160]
Supply chain management in computer industry	[158]
Replenishment	[196]
Assembly and life-cycle management	
Holonic control of an engine assembly system	[172]

Table 1 : Agent systems deployed in industry.

7 STRATEGIC DIRECTIONS

Below we discuss research directions of agent-based computing that are of strategic relevance for manufacturing. Note that a comprehensive roadmap for research and application of agent technology is presented in [1]. Some of the basic ideas came from [197] that explores the notion of service-oriented computing.

7.1 Coordination and cooperation

Coordination and cooperation are still the most ubiquitous issues when designing and running a multi-agent system. Expecting that an agent is both autonomous and cooperative is quite common - but is it reasonable and realistic? The pursuit of common goals, however desirable, is hard to realize if we have autonomous agents and allow them to behave rationally. How can partial, distorted, even conflicting interests be reconciled in the service of overall goals? Can the willingness to cooperate be an intrinsic property of the individuals, or rather it should ensue from the social laws or rules that govern the overall system? Can collaborative work and cooperation be guaranteed at all, and if so, under what conditions? How can one ensure that a community of agents can respond to dynamic changes, both external and internal, technical, economic or environmental, in a swift but smooth way [111] [112]? Finally, how can timely response be guaranteed if computing resources are distributed and limited?

By design, the primary concern of autonomous agents is to do the best for themselves. Hence, cooperation requires giving up individual rationality – at least in the short-term. However, the extent to which individual goals can be fulfilled may depend on the behavior of the entire community. High-level overall performance may be beneficial also for individual agents – but the MAS must have an **incentive system** that provides this leverage to individual decision making. Incomplete information and bounded computational resources can be turned into an advantage. In a multi-agent setting, agents aware of these shortcomings have a direct motivation to ask for missing or inaccessible information and, at the same time, also to answer queries of the others. Hence, what normally are considered barriers of realizing intelligent systems can provide foundations for cooperation.

As we have shown earlier, in certain domains the dynamic nature of manufacturing requires that the agents have not only proactive but also reactive capabilities. This requirement is especially strong in production control (see Section 5.4.) Responsiveness can only be achieved if the agents are allowed to change their commitments dynamically. After that, the net of related commitments has to be repaired. Consequently, collaboration is efficient only if it is based on strong mutual commitments of the partners, and if the partners do not take part in too many joint activities simultaneously. The faithfulness to own intentions and to those of the others has to be sacrificed in the service of responsiveness, redundancy and reactivity. That is why, although they offer strong descriptive power, rough-grained logic-based models of agents (like BDI) are not widely applied in dynamic domains where both reactivity and cooperation are key issues. As an alternative, one can regard a MAS as a **sphere of commitments** which encapsulates the promises and obligations the agents may have toward each other [198].

Coordination, collaboration and cooperation (3C) have **special aspects in manufacturing**. Reflecting on the market-product-manufacturing triplet as a complex, multi-level heterogeneous system,

- 3C have to be realized throughout the enterprise's functional organization, from the highest level business functions (demand planning, master planning, capacity planning, etc.) down to shop floor control and monitoring responsible for executing details plans.
- There is also a need for 3C along the value chain, involving all partners including customers, as well suppliers in a production network.
- The life-cycle axis of products also provides a good opportunity for 3C where all partners – customers included – may have an effect on the product or service.

Throughout the previous sections we referred to many systems that support coordination – and eventually, even cooperation – along some of the above dimensions. However, as for now there is no complete multi-agent model for manufacturing that would embrace all of them.

7.2 Supply networks as complex adaptive systems

The successful integration of the entire production network depends heavily on the availability of accurate and timely information that can be shared by all network members. Information technology with its capability of setting up dynamic information exchange network has been a key enabling factor in the design of production networks to meet such requirements. A major obstacle remains, however, in the deployment of coordination and decision technologies to achieve complex, adaptive, and flexible collective behavior in the network. This is due to the lack of our understanding of organizational, functional and evolutionary aspects in production networks. A key to tackling this problem is to realize that production networks should not just be treated as a "system", but as a **complex adaptive system** (CAS). The study of CAS augments systems theory and provides a rich set of tools and techniques to model and analyze the complexity arising in systems encompassing technology. A similar viewpoint has been emphasized in [199] where the focus was to demonstrate how supply networks should be managed if we recognize them as CAS. The concept of CAS allows one to understand how supply networks as living systems co-evolve with the rugged and dynamic environment in which they exist and identify patterns that arise in such an evolution. The authors conjecture various propositions stating how the patterns of behavior of individual agents in a supply network can be related to the emergent dynamics of the network. One of the important deductions made is that when managing supply networks, managers must appropriately balance how much to control and how much to let emerge. While there are approaches for treating supply chains as CAS [200], no concrete framework has been suggested yet where such conjectures can be verified and generalized.

7.3 Design for emergence

In the previous sections we have reviewed a number of agent-based systems that were all in close correspondence with some manufacturing domains. The individual agents were, however, realized in a wide spectrum: from so-called **coarse-grained** agents (like BDI) with sophisticated communication and cognitive reasoning, as well as learning faculties to **fine-grained** agents with a very limited operational repertoire, but high connectivity and intensive interactions.

It is a general observation that fine-grained but complex systems may display patterns of behavior and develop certain functional properties that cannot be understood and explained solely on the basis of the control of the individuals. In the eye of the observer, these emergent features are unexpected, novel and show the traces of a stable order [28]. Following a widely acknowledged hypothesis, this is due to the feedback between the levels of a hierarchic dynamic system: local dynamics determine global order, which in turn, constrains local dynamics. Accordingly, implicit global complexity emerges from explicit local simplicity [201]. This rule is considered **universal** and characteristic to all systems rich in interacting components: studies of statistical physics, living organisms, animal societies, ecosystems, markets and/or cultural groupings also show some evidence of this [202].

When designing an agent-based system, one has to decide where along this spectrum individual agents should fall. Realizing coarse-grained agents is a classical knowledge-engineering task whose main barriers are the incompleteness and/or inconsistency of available knowledge. On the contrary, approaching the fine-grained end, the key issue is how we can **engineer** multi-agent systems that exhibit **purposive**, goal-directed oriented behavior at the system level by relying on their emergent nature. Generally, how can we design for emergence? Emergence comes with the risk of uncontrolled and unpredictable behavior of the overall system. In particular situations, the behavior of a system may even exhibit the features of chaos [203]. Is there any way to guarantee that the system avoids undesirable paths? How can we specify safeguards if we have to anticipate the unexpected? Computational simulation provides some help – in fact, at the moment the only help – in answering questions like this, but in any case, it can give only partial and incomplete view of the performance of a system, especially if it is complex.

When appealing for emergent functionalities in complex systems, we seem to go back to the old traditions of engineering. Long ago, the words like “machine”, “mechanical”, or “engineer” did not refer to rationality but rather to trickery, artifice and machinery [204]. When dealing with complex systems in this way, we try to elicit effects which are beyond the limits of our actual knowledge. Getting back to what is now considered principled engineering, we need further research in characterizing interactions that may or may not produce emergent phenomena, explore its root causes (such as the dimensionality and connectivity of agents, the flow of information among them, and the propagation of constraints) and develop predictive theories.

7.4 Ontologies

As we emphasized earlier, ontologies have to be developed to support the exchange of meaningful information across autonomous entities that can organize and use information heterogeneously. In manufacturing domains, this issue asserts itself primarily in collaborative engineering (design and process planning included) and in production networks. In these domains, autonomous interacting entities – may they be human or machine agents – use and communicate information and knowledge coming from most heterogeneous sources. Note that the first steps towards knowledge interchange in general were made by the developers of a pioneering collaborative engineering system [65]. Furthermore, ontologies may set the conceptual basis for enterprise integration.

By definition, agents do have a capability to observe the world surrounding them (see Section 3). An ontology commits an observer of the world to see it in terms of individuals and their relationships. Individuals - or things - may be anything that can be referred to: as e.g., in the world of collaborative engineering from the most concrete object such as a particular workpiece up to abstract concepts of machining processes. However, what the things and relationships are depend as much upon the world as upon the observer. Different observers, or the same observer in different roles or at different times, may divide up the world in different ways. More generally, there is a basic tension between the local, situated character of knowledge and action and the ways in which knowledge and action necessarily implicate multiple contexts when used out of the boundaries of an agent, especially in an open system.

Provided there is a common representation language (interlingua) and an accepted interchange format, the case is still controversial when it comes to reasoning over everyday objects and events – i.e., commonsense reasoning. One may formalize context, or alternatively, building up a large knowledge-base by handcrafting microtheories of everyday domains and concepts. Recent developments of ontologies tend to take the latter direction. However, real-world applications of commonsense reasoning are still missing: one cannot but hope that engineering and manufacturing are the fields where its strength will be shown and justified [67]. Commonsense may evade formalization attempts until the senses (visual and tactile perception, balancing, etc.) are taken into account. This, in turn, is certainly beyond the power of symbolic representation and reasoning methods and requires the synthesizing power of the agents concept.

7.5 Embodied and ambient intelligence

In most of the above frameworks, intelligence was a property that emerged as agents interacted with each other and their environment, relying also on individual faculties such as problem solving, decision making, reasoning, dialogue with humans, etc. By the term **embodiment** we refer to the fact that intelligence requires also a body, and all the implications of this hypothesis.

In recent years, various research communities have started to realize the importance of a physical body (and brain-body interaction) for understanding intelligence and its central role in a wide range of processes including perception, object manipulation, movement, locomotion, but also reasoning and learning. These works go beyond the so-called symbol system hypothesis (“intelligence equals symbolic computation”) [40] and point out that physical instantiation and materials play an essential role in intelligent behavior. Also through growth, biological organisms can form highly complex morphological structures. In this respect there are promising starting points for manufacturing: e.g., self-assembling materials and modular robots. One cannot expect equal biological capabilities in the near future, but some aspects of physical growth can certainly be studied through advanced principles of autonomous modular systems.

Shape and materials (e.g., sensors, muscles) perform important functions for an agent in real time. **Morphological computation** [205] opens a novel direction for computation where processes are based on shape (such as molecules, DNA, modules of a modular robot) and material properties (e.g., of the muscle-tendon system). The challenge for manufacturing is to apply morphological concepts in theoretical and practical explorations of embodied agents.

7.6 Interactive computing

For a MAS that consists of agents that “personify” various objects, the inclusion of human persons into an agent-based system has fewer barriers than before. Human decision makers – provided they get sufficiently concise information – can help in resolving conflict situations or solve problems that are beyond the capabilities of local problem solvers.

In fact, agent technology provides an excellent basis for realizing **mixed-initiative problem solving** that supports an ongoing, dynamic interleaving of contributions from human users and computational agents. This is a collaborative activity aimed at converging to some solution where goals and commitments may come from either party. This way of problem solving relies heavily on **interaction**; in fact, it is close to how engineers get to the bottom of problems. Interaction helps to harness external knowledge that could not have been captured and represented internally. Hence, it makes possible to use deficient knowledge in an efficient manner: this has been regarded as a crux of **engineering problem solving** for a long time [206]. According to some theoreticians, interaction provides a more powerful paradigm for computation than the traditional algorithmic models [207].

Human-computer interaction may have new channels for handling massive sensory input, and, on the other hand, for projecting human’s state of mind via some media.

The difficulties with solving artifactual problems we face today are closely related to the difficulty of handling **self-reference** in the environment of the artifacts. Emergent synthesis is an approach to the self-reference problem in the environment of artifacts [28]. Accordingly, “Interactive

Engineering" deals with the self-referential effects of the dynamic interactions among artifacts, producers and consumers.

8 CONCLUSIONS

In the paper we have introduced the agent paradigm of computing and discussed the properties of software agents and the operation of multi-agent systems. After reviewing a number of agent applications, we can conclude that various agent technologies are attractive in all main domains of manufacturing because they offer help in realizing important properties as autonomy, responsiveness, modularity and openness. Multi-agent systems working in a decentralized way are able to use distributed and incomplete sources of information and knowledge. Still, uncertainties and eventual conflicts can be resolved via communication, collaboration and cooperation.

Though the agent-based approach allows for an open-ended design and implementation of complex systems, the problems themselves cannot be solved by less effort, and scalability, safety and traditional software quality are serious bottlenecks. We have summarized the main barriers for the industrial take-up of agent technologies, such as the risk of consistent global operation, the appearance of inevitable conflicts between self-interested entities, and the extra burden of communication. Until recently, the industrial acceptance of MAS in manufacturing has been relative low, partly because of the above issues, and partly because of the difficulties in their stepwise integration with existing legacy systems.

Developments in various agent technologies are still extremely dynamic, innovative and ramifying. At the same time, there is also a strong commitment to convergence with current industrial software technologies [1]. Hence, we can expect that agent-based computing in its deployment phase will overcome most of the difficulties enumerated above.

According to the ManuFuture initiative, the ultimate goal of manufacturing is the "general transformation of all resources to meet human needs" [29] [208] [209]. Four main directions are emphasized: adaptive, digital, knowledge-based, and networked manufacturing. However, these approaches can be considered in a holistic way only under the concept of cooperative manufacturing. The support of various agent technologies is essential, in fact, indispensable, in all these directions.

The further evolution of multi-agent systems and manufacturing will probably proceed hand in hand: the former can receive real challenges from the latter, which, in turn, will have more and more benefits in applying agent technologies, presumably together with well-established or emerging approaches of other disciplines. The authors' intention was to contribute to this fruitful tendency.

9 ACKNOWLEDGMENTS

The authors wish to thank all the contributors for having sent material and/or helped to improve earlier versions of the paper: L. Alting, G. Chryssolouris, B. Denkena, N. Duffie, G. Halevi, J. Jedrzejewski, Z. Katz, J. Lee, P. Leitao, R. Kegg, L. Kops, B. Köppen-Seliger, L. Nemes, D.H. Norrie, G. Perrone, V. Portman, W.B. Rowe, O. Sauer, B. Scholz-Reiter, M.P. Singh, R. Teti, T. Tolio, H.K. Tönshoff, M. Tseng, P. Valckenaers, H. Van Brussel, H.H. Wiendahl, H.P. Wiendahl. The participation of the Hungarian authors in this work was, partially, supported by the NKFP grant No. 2/010/2004, as well as the OTKA grants No. T046509 and No. T049481.

10 REFERENCES

[1] Luck, M., McBurney, P., Shehory, O., et al., 2005, Agent Technology: Computing as Interaction. A Roadmap for Agent-Based Computing, AgentLink III, http://www.agentlink.org/roadmap/al3rm.pdf

[2] Wooldridge, P., 2000, An Introduction to Multiagent Systems, Addison-Wesley, Reading, MA.

[3] Weiss, G., 1999, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. The MIT Press.

[4] Shen, W., Norrie, D.H., 1999, Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey, Knowledge and Information Systems, 1/2: 129-156.

[5] Babiceanu, R.F., Chen, F.F., 2006, Development and Applications of Holonic Manufacturing Systems: A Survey, Journal of Intelligent Manufacturing, 17: 111-131.

[6] Shen, W.; Hao, Q.; Yoon, J.; Norrie, D.H., 2006, Applications of Agent Systems in Intelligent Manufacturing: An Updated Review, Int. J. of Manufacturing Technology and Management, to appear.

[7] Kops, L., 1980, The Factory of the Future - Technology of Management, Towards the Factory of the Future, ASME, PED-1: 109-115.

[8] Kops, L., 1982, Towards Strategy Oriented Scientific Management, ASME, PED-7: 177-187.

[9] Hatvany, J., 1984, Optimizing the Chances of Survival, Annals of the CIRP, 33/11: 345-346.

[10] Hatvany, J., 1985, Intelligence and Cooperation in Heterarchic Manufacturing Systems, Robotics and Computer-Integrated Manufacturing, 2/2: 101-104.

[11] Duffie, N., 1982, An Approach to the Design of Distributed Machinery Control Systems, IEEE Trans. on Industry Applications, IA-18/4: 435-442.

[12] Duffie, N., 1990, Synthesis of Heterarchical Manufacturing Systems, Computers In Industry, 14: 167-174.

[13] Monostori, L., Váncza, J., Márkus, A., et al., 2005, Real-Time, Cooperative Enterprises: Management of Changes and Disturbances in Different Levels of Production, Proc. of the 38th CIRP Int. Seminar on Manufacturing Systems, Florianopolis, Brazil, (CD Version is available).

[14] Wiendahl, H-P., Scholtissek, P., 1994, Management and Control of Complexity in Manufacturing, Annals of the CIRP, 43/2: 533-540.

[15] Merchant, M.E., 1998, An Interpretive Look at 20th Century Research on Modelling of Machining, Inaugural Address, Proc. of the CIRP International Workshop on Modelling of Machining Operations, Atlanta, Georgia, USA, May 19: 27-31.

[16] Onosato, M., Iwata, K., 1993, Development of a Virtual Manufacturing System by Integrating Product Models and Factory Models, Annals of the CIRP, 42/1: 475-478.

[17] Maropoulos, P.G., 2002, Digital Enterprise Technology - Defining Perspectives and Research Priorities, Proc. of the 1st CIRP (UK) Seminar on Digital Enterprise Technology, DET02, Sept. 16-17, Durham, UK, Part V: 3-12.

[18] Monostori, L.; Váncza, J.; Márkus, A.; Kádár, B.; Viharos Zs.J., 2003, Towards the Realization of Digital Enterprises, 36th CIRP Int. Seminar on Manufacturing Systems, June 3-5, Saarbrücken, Germany: 99-106.

[19] Alting, L., Legarth, J.B., 1995, Life Cycle Engineering and Design, Annals of the CIRP, 44/2: 569-580.

[20] Westkämper, E., Alting, L., Arndt, G., 2000, Life Cycle Management and Assessment: Approaches and Visions towards Sustainable Manufacturing, Annals of the CIRP, 49/2: 501-522.

[21] Takata, S., et al., 2004, Maintenance Changing role in Life Cycle Management, Annals of the CIRP, 53/2: 643-655.

[22] Hatvany, J., 1986, Available and Missing AI-Tools, Annals of the CIRP, 35/2: 433-435.

[23] Sohlenius, G.; Kjellberg, T., 1986, Artificial Intelligence and its Potential Use in the Manufacturing System, Annals of the CIRP, 35/2: 425-432.

[24] Teti, R., Kumara, S.R.T., 1997, Intelligent Computing Methods for Manufacturing Systems, Annals of the CIRP, 46/2: 629–652.

[25] Monostori, L., Márkus, A., Van Brussel, H., Westkämper, E., 1996, Machine Learning Approaches to Manufacturing, Annals of the CIRP, 45/2: 675-712.

[26] Wiendahl, H.-P., Lutz, S., 2002, Production in Networks, Annals of the CIRP, 51/2: 1-14.

[27] Vámos, T., 1983, Co-operative Systems - An Evolutionary Perspective, IEEE Continuous Systems Magazine, 3/2: 9-14.

[28] Ueda, K., Márkus, A., Monostori, L., Kals, H.J.J., Arai, T., 2001, Emergent Synthesis Methodologies for Manufacturing, Annals of the CIRP, 50/2: 535-551.

[29] Jovane, F., Koren, Y., Boer, C.R., 2003, Present and Future of Flexible Automation: Towards New Paradigms, Annals of the CIRP, 52/2: 543-560.

[30] Tharumarajah, A., Wells, J., Nemes, L, 1996, Comparison of the Bionic, Fractal and Holonic Manufacturing Systems Concepts, Int. J. on Computer Integrated Manufacturing, 9/3: 217-226.

[31] Valckenaers, P., Bonneville, F., Van Brussel, H., Bongaerts, L., Wyns, J., 1994, Results of the Holonic System Benchmark at KULeuven, Proc. of the Fourth Int. Conf. on CIM and Automation Techn., Oct. 10-12, New York: 128-133.

[32] Van Brussel, H. et al., 1998, Reference Architecture for Holonic Manufacturing Systems: PROSA, Computers in Industry, 37/3: 255-276.

[33] Warnecke, H.J., 1993, Revolution der Unternehmenskultur - Das fraktale Unternehmen, Springer Verlag, Berlin, Germany.

[34] Iwata, K., Onosato, M., Koike, M., 1994, Random Manufacturing Systems: A New Concept of Manufacturing Systems for Production to Order, Annals of the CIRP, 43/1: 379-383.

[35] Ueda, K., Vaario, J., Ohkura, K.H., 1997, Modelling of Biological Manufacturing Systems for Dynamic Reconfiguration, Annals of the CIRP, 46/1: 343-346.

[36] Bussmann, S., Jennings, N.R., Wooldridge, M., 2004, Multiagent Systems for Manufacturing Control: A Design Methodology, Springer.

[37] Bond, A.H., Gasser, L. (Eds.), 1988, Readings in Distributed Artificial Intelligence, Morgan-Kaufmann.

[38] Doyle, J., et al., 1996, Strategic Directions in Artificial Intelligence, ACM Computing Surveys, 28/4: 651-670.

[39] Luck, M., et al., 2003, Agent Technology: Enabling Next Generation Computing. A Roadmap for Agent-based Computing, V1.0, AgentLink. http://www.agentlink.org/roadmap/ al2/roadmap.pdf

[40] Russel, S., Norvig, P., 1995, Artificial Intelligence: A Modern Approach, Prentice Hall.

[41] Jennings, N.R., 2001, An Agent-Based Approach for Building Complex Software Systems, Communications of the ACM, 44/4: 35-41.

[42] Rao, A.S., Georgeff, M., 1992, An Abstract Architecture for Rational Agents, Proc. of the 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann: 439-449.

[43] Dastani, M., Hulstjin, J., van der Torre, L., 2005, How to Decide What to Do? European Journal of Operational Research, 160: 762-784.

[44] Ghallab, M., Nau, D., Traverso, P., 2004, Automated Planning: Theory and Practice. Morgan Kaufmann.

[45] Valckenaers, P., Van Brussel, H., 2005, Holonic Manufacturing Execution Systems, Annals of the CIRP, 54/1: 427-432.

[46] Smith, R.G., 1980, The Contract Net Protocol: High-Level Communication and Control in Distributed Problem Solving, IEEE Trans. on Computers, C-29/12: 1104-1113.

[47] Dorigo, M., Maniezzo, V., Colorni, A., 1996, Ant System: Optimization by a Colony of Cooperating Agents Systems, IEEE Trans. SMC, 26/1: 29-41.

[48] Foundation for Intelligent Physical Agents, www.fipa.org

[49] Searle, J. R., 1970, Speech Acts: An Essay in the Philosophy of Language. Cambridge U. Press.

[50] Mintzberg, H., 1993, Structures in Fives: Designing Effective Organizations, Prentice Hall.

[51] van Aart, Ch., 2005, Organizational Principles for Multi-Agent Architectures. Birkhauser Verlag, Basel.

[52] Patel, M., Honavar, V., Balakrishnan, K., 2001, Advances in the Evolutionary Synthesis of Intelligent Agents, MIT Press, Cambridge, MA.

[53] Hadeli, Valckenaers, P., Kollingbaum, M., Van Brussel, H., 2004, Multi-Agent Coordination and Control Using Stigmergy, Computers in Industry, 53: 75-96.

[54] Ueda, K., Hatono, I., Fujii, N., Vaario, J., 2001, Line-Less Production System Using Self-Organization: A Case Study for BMS, Annals of the CIRP, 50/1: 319-322.

[55] Ueda, K, Fujii, N., Hatono, I., Kobayashi, M., 2002, Facility Layout Planning Using Self-Organization Method, Annals of the CIRP, 51/1: 399-402.

[56] Koestler, A., 1967, The Ghost in the Machine, PAN Book, London.

[57] Walker, S.S., Brennan, R.W., Norrie, D.H., 2005. Holonic Job Shop Scheduling Using a Multiagent System, IEEE Intelligent Systems, 20/1: 50-57.

[58] Ueda, K, Vaario, J., Fujii, N., 1998, Interactive Manufacturing: Human Aspects for Biological Manufacturing Systems, Annals of the CIRP, 47/1: 389-392.

[59] Ueda, K., Hatono, I., Fujii, N., Vaario, J, 2000, Reinforcement Learning Approaches to Biological Manufacturing Systems, Annals of the CIRP, 49/1: 343-346.

[60] Tseng, M., Kjellberg, T., Lu, S.C-Y., 2003, Design in the New e-Commerce Era, Annals of the CIRP, 52/2: 509-519.

[61] Jin, Y., Lu, S.C.-Y., 1998, An Agent-Supported Approach to Collaborative Design, Annals of the CIRP, 47/1: 107-110.

[62] Jin, Y., Lu, S.C.-Y., 2004, Agent-based Negotiation for Collaborative Design Decision Making, Annals of the CIRP, 53/1: 121-124.

[63] Brissaud, D., Zwolinski, P., 2004, End-of-Life-Based Negotiation Throughout the Design Process, Annals of the CIRP, 53/1: 155-158.

[64] Chen, S., Tseng, M., 2005, Defining Specifications for Customer Products: A Multi-Attribute Negotiation Approach, Annals of the CIRP, 54/1: 159-162.

[65] Cutkosky, M., Englemore R., 1993, PACT: An Experiment in Integrating Concurrent Engineering Systems, IEEE Computer, 26/1: 28-37.

[66] Finin, T, 2001, Specification of the KQML Agent-Communication Language - Plus Example Agent Policies and Architectures, http://www.cs.umbc.edu/kqml/kqmlspec/spec.html

[67] Yoshikawa, H., 1993, Systematization of Design Knowledge, Annals of the CIRP, 42/1: 131-134.

[68] Yoshikawa, H., Tomiyama, T., Kiriyama, T., Umeda, Y., 1994, An Integrated Modelling Environment Using the Metamodel, Annals of the CIRP, 43/1: 121-124.

[69] Noel, F., Tichkiewitch, S., 2004, Shared Dynamic Entities Technology to Support Distant Coordination in Design Activity, Annals of the CIRP, 53/1: 163-166.

[70] Jedrzejewski, J., Lorek, F., 2000, Agent-System-Based Design of Intelligent Machining Centres. Proc. of the 2nd CIRP Int. Sem. on Intell. Comput. in Man. Eng., June 21-23, Capri, Italy: 447-451.

[71] McGuire, J., et al., 1993, SHADE: Technology for Knowledge-Based Collaborative Engineering, Concurrent Engineering: Research and Applications, 1: 1-17.

[72] Park, K., Cutkosky, M., Conru, A., Lee, S., 1994, An Agent-Based Approach to Concurrent Cable Harness Design, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 8: 45-61.

[73] Balasubramanian, S., Maturana, F., Norrie, D., 1996, Multiagent Planning and Coordination for Distributed Concurrent Engineering, Int. J. of Cooperative Information Systems, 5/2-3: 153-179.

[74] Parunak, H., Sauter, J., Fleischer. M., 1999, The RAPPID Project: Symbiosis between Industrial Requirements and MAS Research, Autonomous Agents and Multiagent Systems, 2: 111-140.

[75] Sun, J., Zhang, Y., Nee, A., 2001, A Distributed Multiagent Environment for Product Design and Manufacturing Planning, International Journal of Production Research, 39/4: 625-645.

[76] Kambhampati, S., et al., 1993, Integrating General Purpose Planners and Specialized Reasoners: Case Study of a Hybrid Planning Architecture. IEEE Trans. on SMC, 23/6: 1503–1518.

[77] Gupta, S.K., Nau, D.S., Regli, W.C., 1998, IMACS: A Case Study in Real-World Planning, IEEE Intelligent Systems, 13/3: 49-60.

[78] Amara, H., Dépincé, P., Hascoët, J-Y., 2004, A Human-Centered Architecture for Process Planning, CIRP J. of Manufacturing Systems, 33/4.

[79] Lu, S.C-Y., Cai, J., Burkett, W, Udwadia, F., 2000, A Methodology for Collaborative Design Process and Conflict Analysis, Annals of the CIRP, 49/1: 69-73.

[80] Maropoulos, P.G., McKay, K.R., Bramall, D.G., 2002, Resource-Aware Aggregate Planning for the Distributed Manufacturing Enterprise, Annals of the CIRP, 51/1: 363-366.

[81] Maropoulos, P.G., et al., 2003, A Novel Digital Enterprise Technology Framework for the Distributed Development and Validation of Complex Products, Annals of the CIRP, 52/1: 389-392.

[82] Halevi, G., 2004, Production Management – Simple or Complex, International Journal of Innovation and Technology Management, 1/4: 369-372.

[83] Wallace, A., 2003, Sequential Resource Allocation Utilizing Agents, International Journal of Production Research, 41/11: 2481-2499.

[84] Riha, A., Pechoucek, M., Vokrinek, J., Marik, V., 2002, ExPlanTech: Exploitation of Agent-Based Technology in Production Planning, MASA, LNAI 2322, Springer: 308-322.

[85] Bruccoleri, M., Lo Nigro, G., Federico, F., Noto La Diega, S., Perrone, G., 2003, Negotiation Mechanisms for Capacity Allocation in Distributed Enterprises, Annals of the CIRP, 52/1: 397-400.

[86] Bruccoleri, M., Lo Nigro, G., Perrone, G., Renna, P., Noto La Diega, S., 2005, Production Planning in Reconfigurable Enterprises and Reconfigurable Production Systems, Annals of the CIRP, 54/1: 433-436.

[87] Sluga, A., Butala, P., Peklenik, J., 2001, Self-Organization in a Distributed Manufacturing System Based on Constraint Logic Programming, Annals of the CIRP, 50/1:323-326.

[88] Zweben, M., Fox, M. S., 1994, Intelligent Scheduling, Morgan Kaufman Publ., San Francisco, CA.

[89] Wang, L., Balasubramanian, S., Norrie, D.H., Brennan, R.W., 1998, Agent-Based Control System for Next Generation Manufacturing, Proceedings of IEEE Int. Symp. on Intell. Control: 78-83.

[90] Caridi, M., Cavalieri, S., 2004, Multiagent Systems in Production Planning and Control: An Overview, Production Planning and Control, 15/2: 106-118.

[91] Nof, S.Y., Weill, R., 1992, Collaborative Coordination Control (CCC) of Distributed Multimachine Manufacturing, Annals of the CIRP, 41/1: 441-444.

[92] Koussis, K., Pierreval, H., Mebarki, N., 1997, Using Multi-Agent Architecture in FMS for Dynamic Scheduling, J. of Intelligent Manufact, 8/1: 41-47.

[93] Lu, T., Yih, Y., 2001, An Agent-Based Production Control Framework for Multiple-Line Collaborative Manufacturing, International Journal of Production Research, 39/10: 2155-2176.

[94] Papakostas, N., Mourtzis, D., Bechrakis, K., Chryssolouris, G., Doukas, D., Doyle, R., 1999, A Flexible Agent Based Framework for Manufacturing Decision-Making, Proc. of the 9th Flexible Autom. and Intell. Manufact. Conf.: 789-800.

[95] Chryssolouris, G., Papakostas, N., Mourtzis, D., 2000, A Decision-Making Approach for Nesting Scheduling: A Textile Case, International Journal of Production Research, 38/17: 4555-4564.

[96] Prabhu, V.V., Duffie, N.A., 1995, Modelling and Analysis of Nonlinear Dynamics in Autonomous Heterarchical Manufacturing System Control, Annals of the CIRP, 44/1: 425-428.

[97] Prabhu, V.V., Duffie, N.A., 1996, Modelling and Analysis of Heterarchical Manufacturing Systems using Discontinuous Differential Equations, Annals of the CIRP, 45/1: 445-448.

[98] Bongaerts, L., Monostori, L., McFarlane, D., Kádár, B., 2000, Hierarchy in Distributed Shop Floor Control, Computers in Industry, 43/2: 123-137.

[99] Kim, B.I., Heragu, S.S., Graves, R.J., St Onge, A., 2003, A Hybrid Scheduling and Control System Architecture for Warehouse Management, IEEE Trans. on Robotics and Autom., 19/6: 991-1001.

[100] Leitao, P., Restivo, F.J., 2006, ADACOR: A Holonic Architecture for Agile and Adaptive Manufacturing Control, Computers in Industry, 57/2: 121-130.

[101] Wiendahl, H.-P., Ahrens, V., 1997, Agent-Based Control of Self-Organised Production Systems, Annals of the CIRP, 46/1: 365-368.

[102] Tseng, M.M, Lei, M., Su, C., 1997, A Collaborative Control System for Mass Customisation Manufacturing, Annals of the CIRP, 46/1: 373-376.

[103] Kádár, B., Monostori, L., Szelke, E., 1998, An Object Oriented Framework for Developing Distributed Manufacturing Architectures, Journal of Intelligent Manufacturing, 9/2: 173-179.

[104] Langer, G., Alting, L., 2001, Trends and Perspectives - An Architecture for Agile Shop Floor Control Systems, J. of Manufact. Systems, 19/4: 267-281.

[105] Bengoa, A., Gluch, S., Jacobs, H.-J., 1996, An Approach to Holonic Components in Control of Machine Tools, Annals of the CIRP, 45/1: 437-440.

[106] Márkus, A., Kis, T., Váncza, J., Monostori, L., 1996, A Market Approach to Holonic Manufacturing, Annals of the CIRP, 45/1: 433-436.

[107] Maley, J.G., 1988, Managing the Flow of Intelligent Parts, Robotics and Computer-Integrated Manufacturing, 4/3-4:525-530.

[108] Shaw, M.J., 1988, Dynamic Scheduling in Cellular Manufacturing Systems: A Framework for Networked Decision Making, Journal of Manufacturing Systems, 7/2: 83-94.

[109] Wellmann, M.P., 1993, A Market Oriented Programming Environment and its Application to Distributed Multicommodity Flow Problems, Journal of Artificial Intelligence Research, 1: 1-23.

[110] Duffie, N.A., Prabhu, V.V., 1994, Real-Time Distributed Scheduling of Heterarchical Manufacturing Systems, J. of Manufact. Systems, 13/2: 94-107.

[111] Duffie, N.A., Kaltjob, P.O., 1998, Dynamics of Real-Time Distributed Scheduling in Multiple-Machine Heterarchical Manufacturing Systems, Annals of the CIRP, 47/1: 415-418.

[112] Váncza, J., Márkus, A., 2000, An Agent Model for Incentive-based Production Scheduling, Computers in Industry, 43/2: 173–187.

[113] Kumara, S.R.T., Lee, Y.-H., Chatterjee, K., 2002, Distributed Multiproject Resource Control: A Market-Based Approach, Annals of the CIRP, 51/1: 367-370.

[114] Lee, Y., Kumara, S., Chatterjee, K., 2003, Multi-agent-based Dynamic Resource Scheduling for Distributed Multiple Projects Using a Market Mechanism, J. of Intell. Manufact., 14/5: 471-484.

[115] Parunak, H., Baker, A., Clark, S., 2001, AARIA Agent Architecture: from Manufacturing Requirements to Agent-based System Design, Integrated Computer-Aided Engineering, 8/1: 45-58.

[116] Dammann, M., Ouali, K., Wiendahl, H.-P., 2005, Adaptable Agent-based Production Control, 3rd Int. Conf. on Reconfigurable Manufacturing, May 10-12 2005, Ann Arbor, Michigan, USA.

[117] Kádár, B.; Monostori, L.; Csáji, B., 2003, Adaptive Approaches to Increase the Performance of Production Control Systems, Proc. of 36th CIRP ISMS, June 3-5, Saarbrücken, Germany:305-312.

[118] Monostori, L., Csáji, B.Cs., Kádár, B., 2004, Adaptation and Learning in Distributed Production Control, Annals of the CIRP, 53/1: 349-352.

[119] Csáji, B.Cs., Monostori, L., Kádár, B., 2004, Learning and Cooperation in a Distributed Market-Based Production Control System, Proc. of the 5th Intern. Workshop on Emergent Synthesis, IWES'04, May 24-25, Budapest, Hungary: 109-117.

[120] Sutton, R.S., Barto, A. G., 1998, Reinforcement Learning. The MIT Press.

[121] Bertsekas, D.P., Tsitsiklis J.N., 1996, Neuro-Dynamic Programming, Athena Scientific.

[122] Ueda, K., Lengyel, A., Hatono, I., 2004, Emergent Synthesis Approaches to Control and Planning in Make to Order Manufacturing Environments, Annals of the CIRP, 53/1: 385-388.

[123] Monostori, L., 2003, AI and Machine Learning Techniques for Managing Complexity, Changes and Uncertainties in Manufacturing, Engineering Applications of AI, 16/4:277-291.

[124] Rangwala, S.S., Dornfeld, D.A., 1989, Learning and Optimisation of Machining Operations Using Computing Abilities of Neural Networks, IEEE Trans. on SMC, 19/2: 299-314.

[125] Monostori, L., 1993, A Step Towards Intelligent Manufacturing: Modelling and Monitoring of Manufacturing Processes through Artificial Neural Networks, Annals of the CIRP, 42/1: 485-488.

[126] Kamarthi, S.V., Kumara, S.R.T., Cohen, P.H., 2000, Flank Wear Estimation in Turning through Wavelet Representation of Acoustic Emission Signals, J. of Manufact. Sci. and Eng., 122/1:12-19.

[127] Rowe, W.B., Li, Y., Inasaki, I., Malkin, S., 1994, Applications of Artificial Intelligence in Grinding, Annals of the CIRP, 43/2: 521-532.

[128] Bukkapatnam, S., Kumara, S.R.T., Lakhtakia, A., 2000, Fractal Estimation of Flank Wear in Turning. J. of Dyn. Systems, Meas. and Control, 122: 89-94.

[129] Tönshoff, H.K., Wulsberg, J.P., Kals, H.J.J., König, W., Van Luttervelt, C.A., 1988, Developments and Trends in Monitoring and Control of Machining Processes, Annals of the CIRP, 37/2: 611-622.

[130] Weck, M., Monostori, L., Kühne,L., 1984, Universelles System zur Prozess- und Anlagenüberwachung, Vortrag und Berichtsband der VDI/VDE-GMR Tagung Verfahren und Systeme zur technischen Fehlerdiagnose, Langen, FRG: 139-154.

[131] Monostori, L., Barschdorff, D., 1992, Artificial Neural Networks in Intelligent Manufacturing, Robotics and CIM, 9/6: 421-437.

[132] Rowe, W.B., Li, Y., Chen, X., Mills, B., 1997, An Intelligent Multi-Agent Approach for Selection of Grinding Conditions, Annals of the CIRP, 46/1: 233-238.

[133] Monostori, L., Viharos, Zs.J., 2001, Hybrid, AI- and Simulation-Supported Optimisation of Process Chains and Production Plants, Annals of the CIRP, 50/1: 353-356.

[134] Wagner, T., 2003, An Agent-Oriented Approach to Industrial Automation Systems, LNAI2592, Springer: 314-328.

[135] Köppen-Seliger, et al., 2003, MAGIC: An Integrated Approach for Diagnostic Data Management and Operator Support, Proc. of SAFE-PROCESS, June 9-11, Washington D.C., USA.

[136] Heikkila, T., Rannanjarvi, L., Sallinen, M., Rintala, M, 2003, A Holonic Shot-Blasting System, in: Deen, S.M. (ed.), Agent-Based Manufacturing: Advances in the Holonic Approach, Springer: 255-302.

[137] Jarvis, D.H., Jarvis, J.H., 2003, Holonic Diagnosis for an Automotive Assembly Line, in: Deen, S.M. (ed.), Agent-Based Manufacturing: Advances in the Holonic Approach, Springer: 193-206.

[138] Sauer, O., 2004, Modern Production Monitoring in Automotive Plants., Proc of the FISITA World Automotive Congress, May 23-27, Barcelona.

[139] Sauer, O, 2006, Production Monitoring Connected to Object Identification / Localization – A Step towards Real-Time Manufacturing in Automotive Plants, January Assembly of the CIRP, Paris, oral presentation

[140] Teti, R.; D'Addona, Segreto, T., 2004, Flexible Tool Management in a Multi-Agent Modelled Supply Network, Proc. of the 37th CIRP Int. Seminar on Manufacturing Systems, May 19-21, Budapest, Hungary: 351-356.

[141] Yu, R., Iung B. Panetto, H., 2003, A Multi-Agents Based E-maintenance System with Case-Based Reasoning Decision Support, Engineering Applications of Artificial Intelligence, 16: 321-333.

[142] Chen, Z., Lee, J., Qiu, H., 2004, Intelligent Infotronics System Platform for Remote Monitoring and E-Maintenance, International Journal of Agile Manufacturing, 7/2.

[143] Balakriahnan, A., Kumara, S.R.T., Sundaresan, S., 1999, Manufacturing in the Digital Age: Exploiting Information Technologies for Product Realization, Information Systems, 1/1: 25-50.

[144] Sikora, R., Shaw, M., 1997, Coordination Mechanisms for Multiagent Manufacturing Systems: Applications to Integrated Manufacturing Scheduling. IEEE Trans. on Eng. Management, 44/2: 175-187.

[145] Leitao, P. Colombo, A.W., Restivo, F.J., 2005, ADACOR: A Collaborative Production Automation and Control Architecture. IEEE Intelligent Systems, 20/1: 58-66.

[146] Pechoucek, M., Vokrinek, J., Becvar, P., 2005, ExPlanTech: A Multiagent Support for Manufacturing Decision Making, IEEE Intelligent Systems, 20/1: 67-74.

[147] Zha, X.F., 2002, A Knowledge Intensive Multi-Agent Framework for Cooperative / Collaborative Design Modeling and Decision Support of Assemblies, Knowledge-Based Systems, 15: 493-506.

[148] Shen, W., Kremer, R., Ulieru, M., Norrie, D., 2003, A Collaborative Agent-Based Infrastructure for Internet-Enabled Collaborative Enterprises, Intern. J. of Production Research, 41/8: 1621-1638.

[149] Reis, J., Mamede, N., O'Neill, H., 2001, Locally Perceiving Hard Global Constraints in Multi-Agent Scheduling, J. of Intell. Manufact., 12/2: 223-236.

[150] Prabhu, V., Kumara, S.R.T., Kamath, M. 2003, Scalable Enterprise Systems: An Introduction to Recent Advances, Kluwer Academic Publishers.

[151] Denkena, B., Battino, A., Woelk, P.-O., 2005, Intelligent Software Agents as a Basis for Collaborative Manufacturing Systems. Proc. First I*PROMS Virtual Conference, Elsevier Ltd, Oxford, UK: 17-22.

[152] Denkena, B., Woelk, P.-O., Herzog, O., Scholz, T., 2003, Integration of Process Planning and Scheduling using Intelligent Software Agents; The 36th CIRP Int. Seminar on Manufacturing Systems; Saarbrücken, June 3-5: 537-544.

[153] Denkena, B., Herzog, O., Woelk, P-O., Scholz, T., 2003, Application of Intelligent Agents for Concurrent Process Planning and Production Control, Annals of the German Academic Soc. for Production Engineering (WGP), Prod. Eng., 10/1: 15-18.

[154] Kádár B., Monostori, L., 1998, Agent-Based Control of Novel and Traditional Production Systems, Proc. of ICME98, CIRP Int. Sem. on Intell. Comput. in Manufact. Eng., July 1-3, Capri, Italy: 31-38.

[155] Fu, Y., Piplani, R., 2000, Multiagent Enabled Modeling and Simulation towards Collaborative Inventory Management in Supply Chains, Proc. of the 2000 Winter Simulation Conference: 1763-1771.

[156] Qinghe, H., Kumar, A., Shuang, Z., 2001, A Bidding Decision Model in Multiagent Supply Chain Planning, Int. J. of Prod. Res., 39/15: 3291-3301.

[157] Ito, T., Abadi, S., 2002, Agent-Based Material Handling and Inventory Planning in Warehouse, Journal of Intelligent Manufacturing. 13/3: 201-210.

[158] Fox, M., Barbuceanu, M., Teigen, R., 2000, Agent-Oriented Supply-chain Management, Int. J. of Flexible Manufacturing Systems, 12: 165-188.

[159] Garcia-Flores, R., Wang, X.Z., Goltz, G.E., 2000, Agent-Based Information Flow for Process Industries' Supply Chain Modeling, Computers and Chemical Engineering, 24: 1135-1141.

[160] Julka, N., Karimi, I., Srinivasan, R., 2002, Agent-Based Supply Chain Management: Framework and a Refinery Application, Computers and Chemical Engineering, 26: 1755-1769.

[161] Kaihara, T., 2003, Multi-Agent-based Supply Chain Modeling with Dynamic Environment, International Journal of Production Economics, 85: 263-269.

[162] Lou, P., Zhou, Z.D., Chen, Y.P., Ai, W., 2004, Study on Multi-Agent-Based Agile Supply Chain Management, Int. J. of Adv. Manuf. Techn., 23: 197-203.

[163] Mondal, S., Tiwari, M., 2003, Formulation of Mobile Agents for Integration of Supply Chain Using the KLAIM Concept, Int. J. of Prod. Res., 41/1: 97-119.

[164] Chryssolouris, G., Makris, S., Xanthakis, V., Mourtzis, D., 2004, Towards the Internet-Based Supply Chain Management for the Ship Repair Industry, Int. J. of CIM, 17/1: 45–57.

[165] Makris, S., Mourtzis, D., Papakostas, N., Chryssolouris, G., 2005, e-Collaboration for Ship repair Supply Chain Management, 10th IEEE Intern. Conf. on Emerging Techn. and Factory Autom.: 713-718.

[166] Teti, R., D'Addona, D., Segreto, T., 2005, Flexible Tool Management in Multi-Agent Modelled Supply Network, CIRP J. of Mftg. Systems, 34/3: 203-218.

[167] D'Addona, D., Teti, R., 2005, Intelligent Tool Management in a Multiple Supplier Network, Annals of the CIRP, 54/1: 459-462.

[168] Wilhelm, R., Chu, B., Sun, R., 2005, Electronic Commerce Negotiation in a Supply Chain via Constraint Evaluation, Annals of the CIRP, 54/1: 423-426.

[169] Egri P., Váncza, J., 2005, Cooperative Planning in the Supply Network - A Multiagent Organization Model, Multi-Agent Systems and Applications IV, Springer LNAI 3690: 346-356.

[170] Ueda, K., Vario, J., Takeshita, T., Hatano, I., 1999, An Emergent Synthetic Approach to Supply Network, Annals of the CIRP, 48/1: 377-380.

[171] Seliger, G, Krützfeldt, D., 1999, Agent-Based Approach for Assembly Control, Annals of the CIRP, 48/1: 21-24.

[172] Bussmann, S., Sieverding, J., 2001, Holonic Control of an Engine Assembly Plant – An Industrial Evaluation, Proc. of the 2001 IEEE SMC Conf., Tuscon, Arizona: 169-174.

[173] Arai, T., Aiyama, Y., Maeda, Y., Ota, J., 2000, Agile Assembly System by "Plug & Produce", Annals of the CIRP, 49/1: 1-4.

[174] Monostori, L., Ilie-Zudor, E., 2000, Environmental and Life Cycle Issues in Holonic Manufacturing, Proc. of the 33rd CIRP Int. Sem. on Manufacturing Systems, June 5-7, Stockholm, Sweden: 176-181.

[175] Niemann, J., Ilie-Zudor, E., Monostori, L., Westkämper, E., 2004, Agent-Based Product Life Cycle Data Support, IFAC MIM'04 Symposium, October 21-22, Athens, Greece: 103-108.

[176] Dennett, D. C., 1987, The Intentional Stance. MIT Press.

[177] Bergenti, F., Gleizes, M-P., Zambonelli, F., 2004, Methodologies and Software Engineering for Agent Systems, Kluwer.

[178] Agent UML, www.auml.org

[179] JADE – Java Agent Development Framework, http://jade.tilab.com/

[180] CybelePro, http://www.cybelepro.com/

[181] Bordini, R.H, Dastani, M., Dix, J., El Fallah Seghrouchni, A., (eds.), 2005, Multi-Agent Programming: Languages, Platforms and Applications, Springer.

[182] Howden, N., Rönnquist, R., Hodgson, A., Lucas, A., 2001, JACK - Summary of an Agent Infrastructure, 5th Int. Conf. on Autonomous Agents.

[183] Cognitive Agent Architecture (Cougaar), http://cougaar.org/

[184] Recursive Porous Agent Simulation Toolkit REPAST, http://repast.sourceforge.net/.

[185] Berners-Lee, J., Hendler, O., Lassila, E., 2001, The Semantic Web, Scientific American, 284/5: 34-43.

[186] XML.org, www.xml.org

[187] World Wide Web Consortium, www.w3c.org

[188] McGuinness, D.L., Harmelen, F.V., 2004, "OWL Web Ontology Language Overview," http://www.w3.org/TR/2004/REC-owl-features-20040210/, The World Wide Web Consortium - Recommendation.

[189] Zambonelli, F., Jennings, N.R., Wooldridge, M., 2003, Developing Multiagent Systems: The GAIA Methodology, ACM Trans. on Software Engineering and Methodology, 12/3: 317–370.

[190] Scholz-Reiter, B., Höhns, H., Hamann, T., 2004, Adaptive Control of Supply Chains: Building Blocks and Tools of an Agent-Based Simulation Framework, Annals of the CIRP, 53/1: 353-356.

[191] Ueda, K., Nishino, N., Oda, S.H., 2003, Integration of Economics into Engineering with an Application to the Recycling Market, Annals of the CIRP, 52/1: 33-36.

[192] Ueda, K, Nishino, N., Nakayama, H., Oda, S.H., 2005, Decision Making and Institutional Design for Product Lifecycle Management, Annals of the CIRP, 54/1: 407-410.

[193] Marik, V., McFarlane, D., 2005, Industrial Adoption of Agent-Based Technologies, IEEE Intelligent Systems, 20/1: 27-35.

[194] Pechoucek, M., Rehák, M., Marik, V., 2005, Expectations and Deployment of Agent Technology in Manufacturing and Defence: Case Studies, Proc. of AAMAS'05, July 25-29, Utrecht: 100-106.

[195] Cowling, P., Ouelhadj, D., Petrovic, S., 2003, A Multiagent Architecture for Dynamic Scheduling of Steel Hot Rolling, J. of Intell. Manufact., 14: 457-470.

[196] Anthes, G.H., 2003, Agents of Change, Computerworld, January 27.

[197] Singh, M.P., Huhns, M.N., 2005, Service Oriented Computing - Semantics, Processes, Agents, John Wiley.

[198] Jain, A.K, Aparicio M.,Singh, MP, 1999, Agents for Process Coherence in Virtual Enterprises, Communications of the ACM, 42/3: 62-69.

[199] Choi, T.Y., Dooley, K.J., Ruangtusanathan, M., 2001, Supply Networks and Complex Adaptive Systems: Control Versus Emergence, Journal of Operations Management, 19/3: 351-366.

[200] Surana, A., Kumara, S.R.T., Greaves, M., Raghavan, U.N., 2005, Supply-Chain Networks: A Complex Adaptive Systems Perspective, Int. Journal of Production Research, 43/20: 4235–4265.

[201] Langton, C., 1989, Artificial Life, in: Langton, C., (ed.) Artificial Life, Addison-Wesley:1-47.

[202] Parunak, H. van Dyke, Bruecker, S., Savit, R., 2004, Universality in Multi-Agent Systems, Proc. of AAMAS'04, July 2004, New York, USA: 928-935.

[203] Kumara, S.R.T., Ranjan, P., Surana, A., Narayanan, V., 2003, Decision Making in Logistics: A Chaos Theory Based Analysis, Annals of the CIRP, 52/1: 381-384.

[204] Prigogine. I., Stengers, I, 1986, La Nouvelle Alliance, Métamorphose de la Science, Gallimard, Paris.

[205] Hara, F., Pfeifer, R., 2003, (eds.), Morpho-functional Machines: The New Species (Designing Embodied Intelligence), Springer.

[206] Hatvany, J., Lettner, F., 1983, The Efficient Use of Deficient Knowledge, Annals of the CIRP, 32/1:423-426.

[207] Wegner, P., 1997, Why Interaction is More Powerful than Algorithms, Communications of the ACM, 40/5: 80-91.

[208] N.N., 2003, Working Document for the Manu*future* 2003 Conference, Manufuture 2003: European Manufacturing of the Future: Role of Research and Education for European Leadership, December 1-2, 2003, Milano, Italy: 1-51.

[209] N.N., 2004, Manu*future*: A Vision for 2020, Assuring the Future of Manufacturing in Europe, Report of the High-Level Group, EC, November: 1-20.