How Intelligent Manufacturing Holons Configure Themselves
Fletcher, M.; Brennan, R.W.; Xu, Y.

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC-CNRC

*How Intelligent Manufacturing Holons Configure Themselves **

Fletcher, M., Brennan, R.W., and Xu, Y.
November 2001

* published in Proceedings of the IASTED International Conference on Intelligent Systems and Control, Clearwater, Florida. November 19-22, 2001. pp. 37-42. NRC 44931.

Canada

HOW INTELLIGENT MANUFACTURING HOLONS CONFIGURE THEMSELVES

M. FLETCHER[1], R.W. BRENNAN[2], Y. XU[2]
[1]Institute for Manufacturing, University of Cambridge
Mill Lane, Cambridge CB2 1RX, United Kingdom
Email: mf283@eng.cam.ac.uk
[2]Department of Mechanical and Manufacturing Engineering,
University of Calgary,
2500 University Drive NW, Calgary, Alberta T2N 1N4, Canada.
Email: brennan@enme.ucalgary.ca

Abstract: Due to customers' requirements for customized products that meet their unique needs, manufacturing companies have to be more agile with how their shop-floors are configured. One paradigm for ensuring such agility is that of *Holonic Manufacturing Systems (HMS)*. In a HMS, artificially intelligent entities (called *holons*) act both autonomously and cooperatively to manage the shop-floor in a distributed fashion. This paper discusses the technical challenges and business issues associated with how holons configure themselves.

Keywords: holonic manufacturing, configuration control, genetic algorithm.

1 INTRODUCTION

The arrival of advanced manufacturing control technology, high-speed communication networks and the ability of people to buy highly-customized products over the Internet have provided the imputes for manufacturing companies to be more agile in their production.

Holonic Manufacturing Systems (HMS) is a philosophy for introducing sophisticated control techniques onto a shop-floor via the aggregation of autonomous cooperative components that can be managed in a decentralized manner [1][2]. These components, called *holons* [3], enable the company to produce a wider variety of products and deliver them to market in shorter time-scales than at present. Holons use artificial intelligence (AI) techniques to configure themselves, on an order-by-order basis, for the manufacture of the desired product. By configuration, we mean a combination of: (i) the schedule of tasks to be performed at each holon, and (ii) the assignment of function blocks across holons to provide the necessary skills to process information and/or physical workpieces. This philosophy contrasts with traditional hierarchy-based management metaphors and enables the HMS to be sufficiently flexible, self-organizing, fault-tolerant and reactive to meet the commercial challenges that lie ahead in the 21st Century.

This paper outlines some of the key research issues, commercial barriers and social implications of introducing the HMS philosophy. The paper is arranged as follows: Section 2 narrates the HMS infrastructure to support holons' autonomous and cooperative activities in a robust fashion. Section 3 discusses how holons manage their configurations, within the scope of the infrastructure, to achieve agility. Section 4 identifies some of the challenges yet to be overcome. Section 5 concludes the paper.

2 HMS Infrastructure

In this section, we describe the infrastructure that supports persistence, recovery and transaction processing within a holonic manufacturing system. This infrastructure should be viewed as part of some generic holonic system reference architecture [4].

2.1 Scope of Infrastructure

The application of an infrastructure is to facilitate *coherence* via collaboration, coordination, negotiation, cooperation and competition among autonomous decentralized holons. In this context, coherence [5] is a property of the HMS as a whole, so that when it is viewed as a blackbox it appears to be operating like a centrally controlled system with no duplication of skills and activities being executed in optimal order/time. It also enables us to adopt a more pragmatic approach with respect to specifying and solving cooperative manufacturing activities by utilizing well-defined database principles and DBMS functionality. This approach contrasts with the traditional AI oriented approaches to modeling coordinated activities based on negotiation, multi-agent planning etc. Using AI models in isolation has been shown to be *ad hoc* and lack crucial criteria, such as termination etc, within real-world manufacturing scenarios [6]. Our approach (Figure 1) is based on two practical assumptions: (1) the manufacturing holons participating in a cooperative activity are autonomous systems. These systems are based on *agents* that have intelligence [7], mobility [8], software skills [9] and/or cooperation techniques [10], and (2) the execution of an activity can be coordinated via a combination of database technology and artificial intelligence to satisfy the requirements of this non-traditional manufacturing scenario.

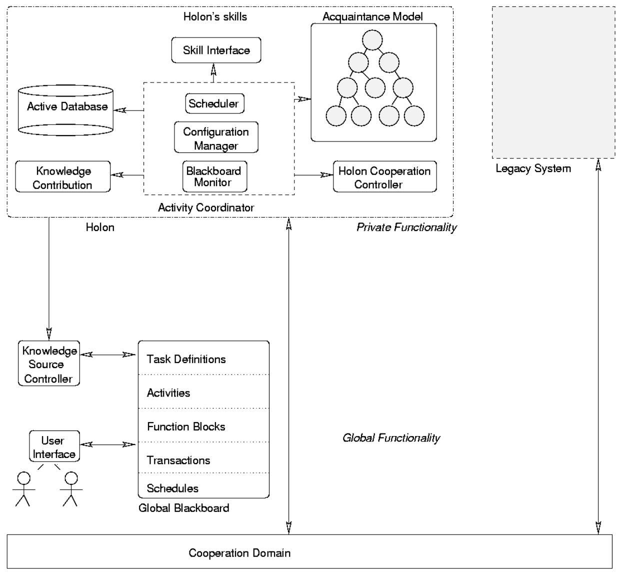

Figure 1: Infrastructure for Holonic Coherence.

Figure 1 will be explained in more detail in the remainder of this section. A conjunction of *active databases* with *blackboards* is proposed as the data architecture to achieve cohesive collaboration among problem solving nodes (i.e. active holons). The global active blackboard acts as a system repository by providing a central data structure divided into panels. A collection of independent knowledge sources (i.e. holons) can read and write to one or more panels under the supervision of a holon activation controller. The controller facilitates global scheduling, concurrency control and system integration. The global active database provides asynchronous processing (via data, events, action requests and conditions being passed as situation-action rules) between the holons involved in the joint problem solving through a conceptual inter-holon structure called a *cooperation domain*. This cooperation domain is a logical space that facilitates cooperation, negotiation etc among holons. Data, condition-action rules (modeling the state of cooperative entities manipulated by the cooperation domain), schedules and function block assignments are stored in the database and managed via conventional transaction processing techniques. The main features of a cooperation domain are:

- Collaboration and information management. This can be handled by a coordinator holon to administrate a joint task, and retain/disseminate knowledge.
- Logical framework for holon inter-connection. We model this property using a *cooperation block (CB)*, i.e. a temporary alliance between a coordinator holon and one or more cohort holons that execute the manufacturing activities.
- Physical communication platform. Holons pass messages using a reliable transport mechanism.

Holons can join a domain, query attributes associated with a domain, exchange information amongst one another through the domain, and depart the domain when their tasks are completed. As part of this information exchange, the coordinator holon can distribute tasks among the cohort holons using protocols such as the Contract Net Protocol (CNP) [11]. We now elaborate on the key elements within the holonic system infrastructure and how the cooperation domain supports them.

2.2 Key Elements of Infrastructure

Loosely coupled problem solving nodes (holons, users and legacy systems) are distributed over a local area network. The functionality of these problem-solving nodes can be replicated at distinct devices. Any omissions mean that the HMS will not be able to manufacture a product with certain features or recover from particular failures etc. The elements in this holonic system infrastructure are: the set of problem solving nodes, the group of activity coordinators (one per holon), the active databases in each holon, and the global blackboard. We briefly outline the functionality of these elements to provide decentralized configuration management.

2.2.1 Problem Solving Nodes

Problem solving nodes perform sub-activities and atomic actions as part of a manufacturing activity to produce customer-specific goods. Human user or legacy non-holonic equipment can adopt the role of a node with a particular set of knowledge, responsibilities, roles, skills, quality characteristics and constraints. A node has resources (e.g. software, computing/manufacturing hardware and communication facilities) and information (e.g. data, condition-action rules, dictionaries, meta-data about itself and knowledge of other nodes' capabilities) to execute various activities.

Typically the knowledge and functionality associated with a node remains reasonably static over time. However, facilities are required to let the nodes evolve and be replaced. This evolution is implemented as a Kaizen (i.e. incremental) migration policy to improve the system's performance and flexibility. These nodes are autonomous and therefore their knowledge may not be directly communicated due to security or differing formats etc. Thus nodes perform their activity processing with minimal knowledge required to coordinate with other nodes, and have very little or no knowledge of local processing done by other nodes. The holon's activity coordinator controls any information that is exchanged to coordinate execution. An *activity* is decomposed into a set of sub-activities, until the leaves of this recursive division are atomic actions that can be performed by the nodes. To complete an activity, a subset of the HMS's resources must be allocated and their utilization locked. Moreover, a multi-holon plan must be established to coordinate nodes' activities. A coordinated activity will be: specified in a declarative fashion by a holon, translated into high-level data structures and condition-action rules for execution by holons, and have suitable knowledge relating to its execution be exchanged via the cooperation domain.

An activity can be modeled as a project network. This network is represented as a directed graph where vertices represent decision points of holons and/or function blocks, while edges correspond to the flow of data. An activity is dynamic, distributed and is managed via cooperation among holons' activity coordinators.

2.2.2 A Holon's Activity Coordinator

The activity coordinator is part of the holon's control system (i.e. a software module) and is responsible for the execution of the holon's activities. These activities include: (i) performing atomic actions on information and/or physical workpieces, (ii) decomposing the manufacturing activity further, and (iii) organizing interaction with other holons, users and legacy shop-floor systems. The activity coordinator has both a local view and a partial-global (participant) view of the processing being performed at various nodes in the problem-solving network. The activity coordinator has access to both private and public data structures that model: (1) the requirements of each activity in terms of decomposition strategies, resources demanded etc., and (2) the interaction between these activities with respect to synchronization rules and so forth.

During its task allocation strategy (possibly done via a CNP) the activity coordinator tries to match the activity's requirements with the available resources in the HMS and which holons manage these resources. The activity coordinator interacts with the schedulers and blackboards of various holons to track the progress of (sub)activities and atomic actions distributed across the holonic system. Figure 2 shows the elements of this interaction using a UML collaboration diagram. The operations needed to perform (sub)activities and atomic actions are stored as resource information in the holon's active database.

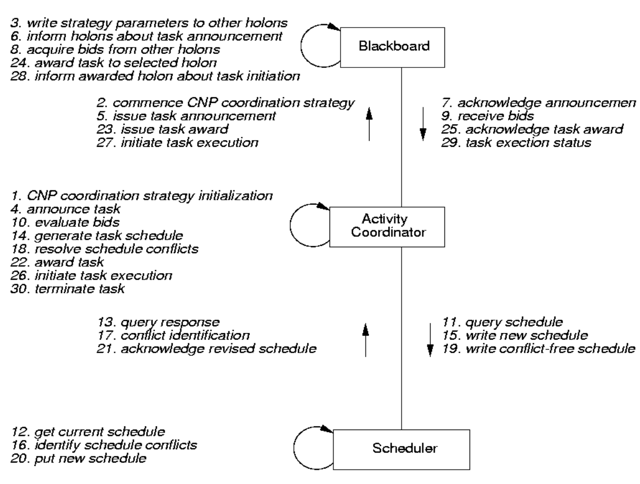

Figure 2: Activity Coordinator Interactions for CNP.

These operations are then performed by the various function blocks within the responsible holon once an allocation has been done. When a plan is constructed, skills and their timings are explicitly specified in the activity plan. This plan describes the interaction between problem solving nodes, the interaction between sub-activities and atomic actions within the scope of the activity plan, the assignment of work to nodes, and their scheduled time of execution.

The plan also specifies the protocol and ontology used by each holon's activity coordinator for interaction. Users have transparent access to the plan via a high-level specification language. We will not address that issue here due to paper economy. This access enables the users to define details of inter-holon coordination, cooperation, negotiation and so forth, and specify the mapping of sub-activities to holons etc. One mechanism to facilitate the aforementioned coherence in the system is by means of condition-action rules. Both fire these rules:

- The active database within the holon. This invocation causes messages to be passed between the respective holons' control systems.
- The IEC 61499 function blocks [12][13] that perform real-time atomic actions and interface to the manufacturing shop-floor to read sensors and set actuators' values. This causes data and events to be passed between function blocks inside holons.

The activity coordinator mediates knowledge exchange, passes partial/complete solutions between nodes and initiates/terminates local sub-activities and atomic actions. Hence the only assumption our holonic system infrastructure makes with respect to information exchange is that the problem-solving node can pass knowledge to and from its corresponding activity coordinator. For legacy systems, the activity coordinator is a software wrapper to encapsulate the functionality of the system and enable it to interact with other nodes in the HMS. For users, the activity coordinator is a cognitive decision process in the human mind. We do not investigate this issue further here. A more appropriate vehicle for studying these decisions is within research on the holon/human interface.

2.2.3 A Holon's Active Database

The definition of sub-activities, atomic actions and their interactions determines the structure and operations of the active database. The structure is also affected by the requirements to support the plan for executing and managing the set of activities in the holonic environment. The active database system supports a collection of rules in the form of events, conditions and actions. The order of these condition-action rules is as follows:

- An event occurs. These events include:
    1. The insertion of a tuple into the database.
    2. Receipt of a message.
    3. Acquiring sensor data from physical environment.
- A condition is evaluated (possibly including retrieval from relations). This condition evaluates to a Boolean (i.e. values are either TRUE or FALSE).
- If this Boolean condition evaluates to true then the associated action is executed.

Such scenarios draw upon classic database technology to provide robustness. Each holon is responsible for monitoring the state of its active database and for performing the triggered actions. In addition to its other characteristics, the activity plan specifies a group of scenarios that could arise during execution of a given activity. The plan also defines the set of actions to be performed in predefined cases (e.g. remedies upon certain fault states). These states and compensating actions are encoded as condition-action rules. Such rules are modeled in terms of the available functionality offered by the holon's active database. The role of the active database in enabling the holon to coordinate its activities with other nodes in the system is critical. Hence a methodology for implementing such model-based and rule-based approaches in a holon's active database appears a key requirement. Without this methodology, all the functionality of the holon would have to be implemented in the activity coordinator. This coupling has some consequences: (i) it increases complexity, (ii) it reduces software cohesion, and (iii) it creates a bottleneck.

2.2.4 Global Blackboard

The blackboard acts as a centralized data structure upon which holons read and write the following knowledge: assignments of function blocks to holons, task schedules to denote by what deadlines elements of activity plans are to be accomplished, partial solutions created during an activity, data used to coordinate (sub)activities, and advertisements of services offered by holons.

We use the first two types of knowledge for distributed configuration management (see next section). The blackboard is particularly useful for synchronizing the multi-holon sub-activities and atomic actions. Blackboards have been extensively studied in various AI application domains. However their use in manufacturing and initiation/monitoring of actions (as a consequence of writing to the blackboard) are novel features of our approach. The use of condition-action rules on the blackboard enables holons to share information and execute sub-activities/atomic actions in a synchronous fashion. These elements are critical to the flexibility and independence demanded by holons. The active elements of the blackboard will provide the necessary support for coherence over the constituent holons and the activities they are performing. Any lack of artificial intelligence at individual holons could be compensated for by a combination of the blackboard and the cooperation domain. The key assumption made in our approach is that each holon is able to communicate data using a suitable agent communication language [11].

Each blackboard is accessible for reading and writing by all holons or users. Where needed these blackboards can be made consistent to ensure a global accessible space for posting solutions and have constraints verified. The configuration management for an activity (including its planning, assignment of appropriate function blocks to accomplish the activity and scheduling) has to be managed by the activity coordinators as they are responsible for data processing, dissemination and control.

3 Configuration Management

We represent the configuration of task schedules and function block assignments to holons as a pair of chromosomes. These can be manipulated, as a genetic algorithm, via suitable operators at each holon. The *task schedule* and *FB assignment* chromosomes are modeled using the blackboard so that distributed holons can read and write revised configurations in a persistent and consistent manner.

Note that such a genetic algorithm can quickly realize fast convergence by assigning the best chromosome to be a child during each generation. To avoid slow convergence and stop the algorithm entering into cycles, a random chromosome can be introduced at an appropriate juncture to 'kick-start' the generation process. Figure 3 shows how the pair of chromosomes are composed and represented through the blackboard.

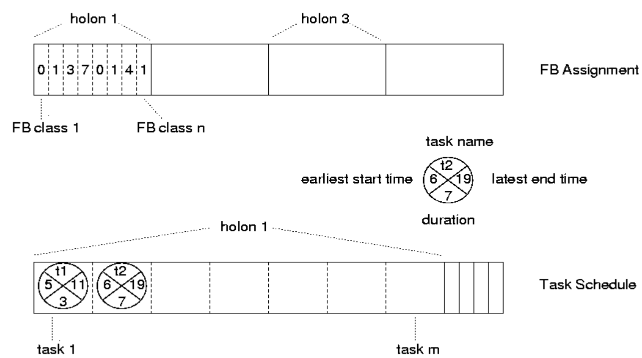

Figure 3: Composition of Chromosome in Blackboard.

The *FB assignment* chromosome is divided into regions; one region per holon *h* in the HMS. Each cell of every region in the *FB assignment* chromosome represents a function block class and contains an integer value of zero or more to denote how many instances of that class are to be resident within the jurisdiction of *h*. Similar holon regions exist in the *task schedule* chromosome; cells denote the tasks allocated to *h*, their earliest start time, latest end time and duration. The fitness of the overall solution is calculated after being converted into the equivalent configuration. The flow of the configuration management process is illustrated in Figure 4. The effectiveness of the proposed method can then be judged. We are in the process of developing a simulation environment to experiment with how the proposed method could work in practice. We intend to publish results as they become available. This is just one barrier to full-scale deployment of holonics in industry.

4 Deployment Challenges

In order to automate and deploy (re)configuration management with holon technologies, in addition to gaining a thorough theoretical comprehension of the dominant strategies (if they exist) under which different configuration formats and rules can be applied, there are two major research challenges to be overcome. First, a uniform platform must be constructed to support various configuration management policies, mechanisms, rules and structures. Also the platform must cater for a variety of bespoke and general-purpose holon configuration management strategies. Second, a mechanism for capturing and analyzing business preferences and strategies must be established. These preferences relate to specification of holons' private utility functions for judging the merits of a configuration, real-time sensitivity in changing the configuration and the options when selecting appropriate operators to manipulate the chromosome during reconfiguration. The strategies cover aspects like how should the real-time constraints be treated as a function of the messages from other holons.

The mechanics of HMS configuration management vary dramatically between different implementations. Crucial variables consist of open or restricted sets of possible function blocks to be incorporated into the configuration, unilateral or multi-lateral decision making by holons (hence mechanisms for task allocation and synchronization), dictatorial or collaborative means of setting conflicts, and so forth.

A generic inter-holon cooperation protocol is demanded to: (i) operate within the scope of the cooperation domain, (ii) encapsulate all these varieties and strategies, and (iii) be extensible. In configuration management, the coordinator holon has the responsibility to select the rules and mechanisms for the (re)configuration strategy, which may not belong to any previously established strategy. Hence, designing an abstract platform and protocol capable of facilitating a wide array of configuration issues is one of the most challenging aspects for deploying this new breed of HMS. Holons cannot be assumed to use strategies all designed at a central location, but instead will endeavor to apply policies that maximize their individual utility function during the reconfiguration process. Therefore the platform and the protocols to enable configuration management must allow the following:

- Externally developed holons (e.g. say, from a company that regularly receives outsourced manufacturing tasks).
- Diverse strategies from out-of-company sources.

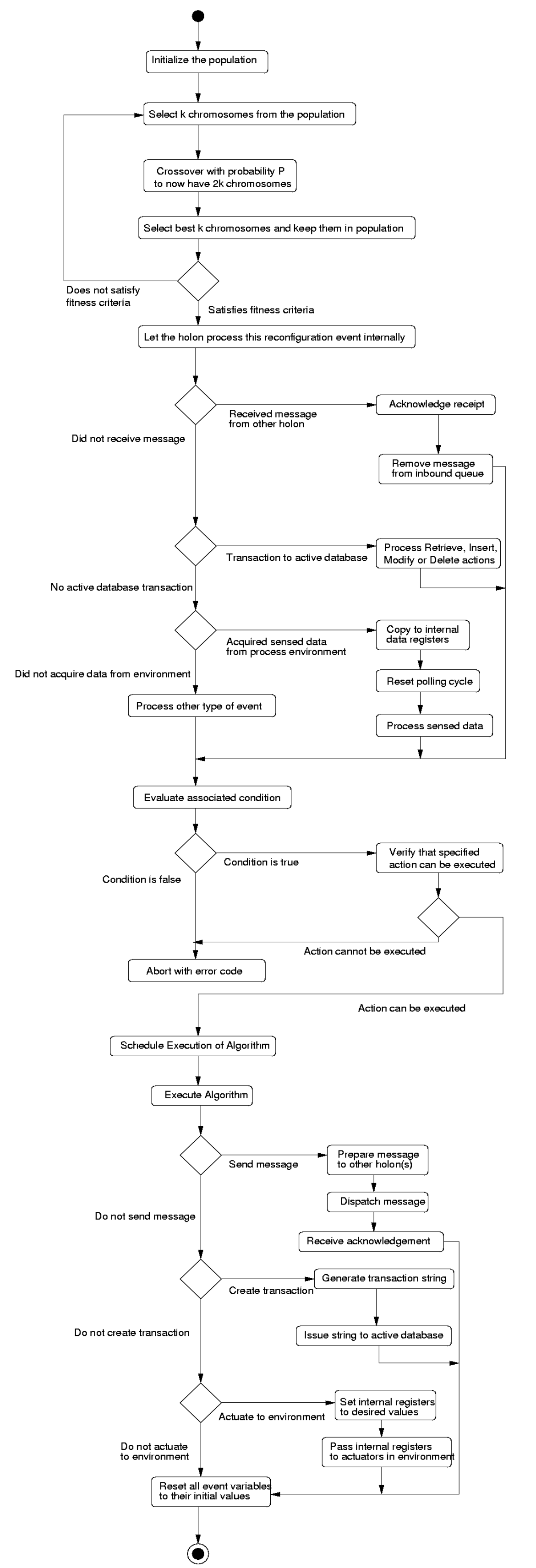

Figure 4: Configuration Management Flow via GA.

These elements must be incorporated seamlessly into the HMS and participate, in a coherent manner, during any changeover periods. Technically, this is much more complicated than just having an interoperable standard and a methodology to apply such foreign holons/strategies in an open manner. As most manufacturing businesses are not interested in the design of holons *per se*, but only in their reasoning and the services they offer, the problem becomes one of 'how can new strategies, in different languages, formats and using varied ontologies be integrated into the HMS at runtime'. To facilitate inter-holon configuration management, mechanisms to capture business' strategies (at a conceptual level) and dynamically apply them within holons (at an implementation level) are a necessary characteristic. Regretfully, this approach may prove to be technically and cognitively too complex for application in real-world manufacturing; at least from a short-term perspective.

To make this approach a reality, the holons' decisions relating to what function blocks reside where, what tasks are assigned to which holon and what deadline constraints must be satisfied must be represented using game theory in terms of moves in a well-formed game. The key to success could be to represent a small set of one holon's decisions based on its autonomous utility function and its cooperative strategy, and be able to extrapolate them to predicate probable decisions.

It will be even more complicated if risk characteristics for each holon are introduced to reflect that fact that compensatory payments have to be made to other holons (possibly in different companies as part of a supply chain) for breaching a contract. For instance, for not performing a manufacturing task on time and so delivering a partially assembled product late. Hence company *a* may have to reduce its price for the product being delivered to company *b* by 10% per day for every 24 hours it is delayed. These risk characteristics could be a function of messages received from other holons, data from the current (re)configuration process, or the prices from the inter-company market in which the business is participating. Complex as it might be, correlation and affiliation between ongoing reconfiguration processes may be crucial if holons are to engage in multiple reconfiguration processes simultaneously. This is especially so if the configuration management processes are mutually dependent or holons are subject to dynamic constraints on their resources, budgets and so on.

5 Conclusion

We have presented a model of holon configuration management and discussed a number of issues associated with deploying holons on manufacturing shop-floors. We have also raised certain challenges that must be overcome through further research. As companies increasingly shift their emphasis towards high-variety low-volume production, artificially intelligent holons and distributed control are believed to be the logical consequence.

Note that the above discussion assumes that the user can describe their product's features in a suitable high-level specification language and that the holons can understand how this specification affects their configuration. We envisage that in the long term, Internet-based systems (used to assist users in specifying their customized products etc) and legacy shop-floor production, assembly and fixturing machines will be more tightly integrated. By applying a HMS infrastructure, such coupling and inter-holon cooperation can be managed in a cohesive fashion. This paper is one step towards that goal.

Clearly there is incentive for businesses to introduce holonic ideas onto their shop-floors in order to satisfy the ever-growing demand for customer-specific products. It is only a matter of time for holons to become one of the mainstream tools for providing agility and reconfiguration on the shop-floor. Moreover competition between businesses to manufacture such profitable customized products and the integration of holons with supply chain management will make HMS's arrival imminent.

REFERENCES

[1] J.H. Christensen, HMS: Initial Architecture and Standards Directions, In *the 1st European Conference on Holonic Manufacturing Systems*, 1994.

[2] Overview of Holonic Manufacturing System Project, 2001, http://hms.ifw.uni-hannover.de/public/overview.html

[3] A. Koestler, *The Ghost in the Machine* (London: Arkana 1967).

[4] M. Fletcher *et al*, An Open Architecture for Holonic Cooperation and Autonomy, In *the 11th conference on Database and Expert System Applications*, IEEE, 2000.

[5] A. Bond and L. Gasser, *An Analysis of Problems and Research in DAI* (Morgan Kaufman 1988).

[6] H. Van Brussel *et al*, PROSA: A Reference Architecture for Holonic Manufacturing Systems, *Computers in Industry,* 37(3), 1998.

[7] N. Jennings, *Coordination Techniques for Distributed Artificial Intelligence*, (John Wiley and Sons, 1995).

[8] V. Roth and M. Jalali-Sohi, Concepts and Architecture of a Security-centric Mobile Agent Server, In *the 5th International Symposium on Autonomous Decentralized Systems*, published by IEEE Computer Society, 2001.

[9] E. Durfee *et al*, Coherent Cooperation Among Communicating Problem Solvers, In *Readings in Distributed Artificial Intelligence*, Morgan Kaufman, 1988.

[10] A. Haddadi *et al*, *Communication and Cooperation in Agent Systems: A Pragmatic Theory* (Springer, 1996).

[11] Agent Communication Language, the non-profit Foundation for Intelligent Physical Agents (FIPA), Technical report, http://www.fipa.org, 1997.

[12] M. Fletcher, D.H. Norrie, and J.H. Christensen, A Foundation for Realtime Holonic Control Systems, *Journal of Applied System Sciences,* 2001.

[13] International Electro-technical Commission, Function Block Architecture (numbered IEC 61499), Technical report, http://www.holobloc.com, 2001.