

A Review of Holonic Manufacturing Systems Literature

Authored by Amro M. Farid

Supervisor: Dr. Duncan McFarlane

Center for Distributed Automation & Control

Institute for Manufacturing

Engineering Department

University of Cambridge

June 9, 2004

Abstract

This reports gives an extensive review of the holonic manufacturing systems literature in 2004. After a brief introduction of holonic principles and concepts, a full rationale of holonic systems in manufacturing is given. In particular, an argument is developed that holonic manufacturing systems provide greater agility than their current industrially adopted counterparts. The body of the review highlights the most significant architectures, methodologies, protocols, algorithms and interactions. Finally, open research issues and barriers to industrial adoption are extracted from the review of the current state of the literature.

Contents

1	Introduction	7
1.1	Background to Holonic Systems	8
1.2	Holonic Systems in Manufacturing Context	9
2	HMS Rationale & Its Requirements	12
2.1	Business Drivers Rationale	12
2.2	Manufacturing System Rationale	13
2.2.1	CIM Systems	14
2.2.2	Heterarchical Systems	18
2.2.3	Holonic Systems	19
2.3	Control System	21
2.3.1	Requirements and Recommendations	21
2.3.2	Available Technologies	22
3	Developments in Holonic Manufacturing Systems	23
3.1	Architectures	24
3.1.1	Conceptual Description	24
3.1.1.1	Initial Conceptions	24
3.1.1.2	Cooperation Domains	25
3.1.1.3	PROSA – Product, Resource, Order, Staff Architecture	26
3.1.1.4	HCBA – Holonic Component Based Architecture	27
3.1.1.5	HoMuCS – Holonic Multi-Cell Control System	29
3.1.1.6	Agent Architecture Taxonomy	29
3.1.1.6.1	Reactive Agents	30
3.1.1.6.2	Deliberative Agents	31
3.1.1.6.3	Hybrid Agents	31
3.1.1.7	Multi-Agent System Taxonomy	32
3.1.1.7.1	Federation Architectures	33
3.1.1.7.2	Autonomous Agent Architectures	34
3.1.1.8	Miscellaneous Conceptions	34
3.1.2	Detailed Design and Simulated Implementations	34
3.1.2.1	Rockwell/BHP Steel Rod Mill Water Cooling System	35
3.1.2.2	VTT Automation Robot Cell	35

3.1.2.3	Holden’s Engine Operations Engine Assembly Line	38
3.1.2.4	Miscellaneous Designs	40
3.1.3	Implementations	40
3.1.3.1	Daimler Chrysler Holomobiles	40
3.1.3.2	University of Cambridge Robot Assembly Cell	42
3.1.3.3	KU Leuven	45
3.1.3.4	Miscellaneous Implementations	46
3.2	Methodologies & Strategies	46
3.2.1	Design & Modelling Tools	46
3.2.1.1	IDEF0	46
3.2.1.2	Petri-Nets	47
3.2.1.3	The Unified Modelling Language – UML	47
3.2.1.4	Agent Building Tools	49
3.2.2	Design Methodologies & Strategies	49
3.2.2.1	PROSA	49
3.2.2.2	HoMuCS Methodology	50
3.2.2.3	DACS Multi-Agent System Design Methodology	51
3.2.3	Migration Methodologies & Strategies	51
3.2.3.1	HCBA Migration Strategy	52
3.2.3.2	Daimler-Chrysler Holomobiles Migration Strategy	52
3.2.3.3	Holden’s Engine Operations Migration Strategy	53
3.3	Protocols	53
3.3.1	Contract-Net Protocol	53
3.3.1.1	The Tri-Base Acquaintance Model	54
3.4	Algorithms	55
3.4.1	Scheduling Algorithms	56
3.4.1.1	Lagrangian Relaxation Scheduling	56
3.4.1.2	Temperature Equilibrium Scheduling	57
3.4.1.3	Market-Driven Scheduling	57
3.4.2	Integrated Planning Scheduling Algorithms	58
3.4.2.1	Lagrangian Relaxation with Planning	58
3.4.2.2	Market Based Approaches	59
3.5	Interactions	59
3.5.1	IEC 61499 Function Blocks	59

3.5.2	Function Block Operating System	61
4	Open Issues & Industrial Adoption	62
4.1	HMS System Design	62
4.1.1	Availability of Proven Design Methodologies	62
4.1.2	Analysis of Performance of Holonic Manufacturing Systems	63
4.2	HMS System Implementation	63
4.2.1	Appropriate environments for implementing holonic control	63
4.2.2	Establishing suitable standards for holonic control systems	63

List of Figures

1	A Generic Architecture of a Holon [42]	9
2	A Waterfall Chart Linking Business Drivers to Control System Needs[175][44]	13
3	The Basic Functions of Manufacturing Facility [199]	16
4	A Typical Manufacturing Control Hierarchy [172]	17
5	An Abstracted Hierarchical Structure [77], [29]	17
6	An Abstracted Heterarchical Structure [28], [29]	19
7	An Abstracted Holarchy Structure [28][29]	20
8	A General Activity Model of a Holon [61]	25
9	The Critical Holon Interfaces in Holonic Manufacturing Systems [61]	25
10	Basic Buildings Blocks of PROSA Architecture [246]	26
11	Aggregation and Specialization in PROSA Workshop [245]	27
12	HCBA Holon Architecture [54]	28
13	HCBA System Architecture [58]	29
14	Outline of HoMuCS Architecture [138]	30
15	A Subsumption Agent: A Reactive Architecture [43][40]	31
16	A BDI Agent: A Deliberative Architecture [43][197]	32
17	An InterRaP Agent: A Hybrid Architecture [43][177]	32
18	A Desktop Utility Agent System: A Facilitator Architecture [65]	33
19	An Example Broker Architecture [211]	33
20	An Example Mediator Architecture [211]	34
21	A Conceptualized Autonomous Agent Architecture [211]	35
22	Water Cooling System Schematic [4]	36
23	Holon Architecture [4] [170]	36
24	The VTT Automation Holon Structure [114]	38
25	The VTT Simulated Manufacturing Robot Cell [114]	39
26	The Flat VTT Robot Cell Holarchy [114]	39
27	A Conventionally Controlled Mercedes Benz V6/V8 Engine Assembly Line [47]	41
28	A Conventionally Controlled Line with Implemented Holons [47]	41
29	A Holonically Assembled Meter Box [55][259]	42
30	Robotic Assembly Cell Meter Box Diagram [55][259]	43
31	University of Cambridge Holonic Robot Assembly Cell [55][259]	43
32	Static and Dynamic HCBA Integration [55]	44

33	Univeristy of Cambridge Robot Assembly Cell HCBA Architecture [55]	44
34	Holonic Flexible Assembly System at KU Leuven [241]	45
35	An Illustrative IDEF0 Example	47
36	An Illustrative Petri-Nets Example [199]	48
37	An Illustrative PROSA based UML example [246]	48
38	An Overview of the HoMuCS Methodology [138]	50
39	A Migration Framework for a Holonic Factory [57]	52
40	The Contract-Net Protocol [43][224]	54
41	The IEC 61499 Function Block Model [39]	60
42	IEC 61499 Function Block Data/Event Synchronization [39]	61
43	A FBOS based Holonic Manufacturing System	62

1 Introduction

The Holonic Manufacturing Systems (HMS) field was initiated in Japan by Suda [230], [231] as a response to the growing perception that Japanese manufacturing firms lacked competitiveness in a global manufacturing environment. He hypothesized that the cause of this inability to compete was rigid manufacturing practices that did not have the necessary agility¹ and responsiveness² in increasingly volatile markets. Suda noted that the robustness³, flexibility⁴, and adaptability⁵ of holonic systems would be a highly desirable characteristic in the effort to regain international competitiveness⁶.

Since that point, numerous and relevant technology contributions have been made in the field of holonic manufacturing systems. This review paper summarizes the most relevant and

1

Definition 1.0.1. Agility – The quickness with which a system adapts to modifications of the production and/or product envelop in the sense of operation flexibility [100]. It is both reactive (responsive) and proactive [208].

2

Definition 1.0.2. Responsiveness – The ability of a production system to respond to dynamic conditions (originating inside or outside the manufacturing organization) which impact upon production goals [160]. It forms the reactive part of agility [208].

3

Definition 1.0.3. Robustness – The ability of a system to not have to respond or change its behaviour because it is designed well enough so that these disturbances do not affect the output [100].

4

Definition 1.0.4. Flexibility – The facility with which a system design may be modified to meet similar product requirements [100], [67] and extended with new elements to augment the existing level of functionality [223], [67], [209].

5

Definition 1.0.5. Adaptability – The ability of a system in operation to change behaviour to maintain its desired output in the presence of external and/or internal disturbances [100].

⁶Robustness, flexibility and adaptability are necessary parts of responsive and agile systems. The first being a subset of the second [208].

effective of these contributions with the intention of identifying the features, attributes and modules that would lead most effectively to an industrial realization of a holonic manufacturing system. In doing so, it relies upon previously published reviews [44], [172], [175] and builds upon them wherever possible.

This report illustrates these points in the following sequential method of approach. This introduction continues in Section 1.1 with a description of holonic systems' origin, and later in Section 1.2 gives a very topical description of how holonic principles may be applied in manufacturing. Section 2 presents a three-part holonic manufacturing system rationale that builds from the business drivers to the manufacturing system implications and their requirements on the chosen control system. Section 3 forms the body of this review and consists of the technological developments most relevant to holonic manufacturing systems. These developments can be divided into: architectures which vary in detail from conceptual descriptions to partial implementations, design methodologies which attempt to specify systematic ways to develop detailed architectures, and finally the protocols, algorithms, and interactions which together give holonic manufacturing systems their required agility. Finally, Section 4 closes the review by identifying the open issues, specifically in HMS design and implementation that prevent their straightforward industrial adoption.

1.1 Background to Holonic Systems

Holonic systems originate from the works of the philosopher A. Koestler [134], who in 1967 proposed the term holon to describe his observations of the behaviour of biological and social systems. The word holon itself originates from the Greek word "holos" meaning "whole" and the suffix "on" meaning "part of" i.e. a neutron or proton. He found that all biological and social systems evolve, grow, and adapt to complex and changing environments by forming stable intermediate holons. More specifically, holons exhibit a dual behaviour which he called the Janus Effect [38]. On the one hand, each holon has an autonomous quality. Its development and functionality is sufficient to exist alone. On the other hand, each holon also has a cooperative quality that allows it to depend upon a social framework of holons. In such a way, they interact together to meet overall goals of the collective [134].

It was these ideas which Suda felt would be particularly beneficial in a manufacturing system [230], [231]. Soon afterwards, the Holonic Manufacturing Systems Project with its associated research consortium was formed as one of the six Intelligent Manufacturing Systems (IMS) feasibility studies [112], [207], [100], [64]. Since then, numerous conceptions (of

varying degrees of similarity) were proposed to bring holonic principles to a manufacturing context. In order to instantiate more concretely such a holon, one generally accepted conceptual architecture is introduced in Figure 1. Its composition is explained in further detail in Section 1.2.

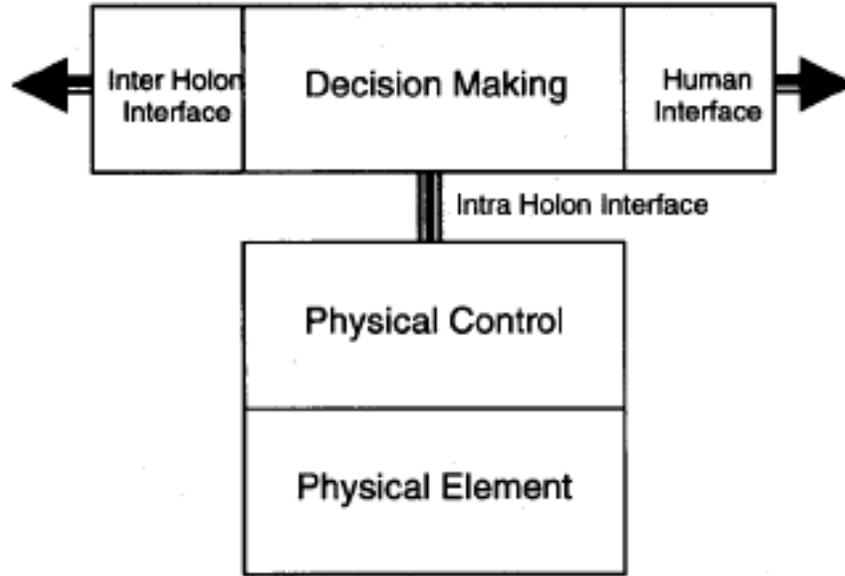


Figure 1: A Generic Architecture of a Holon [42]

1.2 Holonic Systems in Manufacturing Context

The holon concept above must adhere rigorously to holonic behaviours, properties and attributes which have been carefully defined by the HMS consortium. It is defined below.

Definition 1.2.1. Holon – An autonomous and cooperative building block of a manufacturing system for transforming, transporting, storing and/or validating information and physical objects. A holon consists of an information processing part and often a physical part. A holon can be part of another holon [100].

In addition to the properties of autonomy and cooperation, some authors [158], [175], [44] add recursivity, self-organization, and reconfigurability to the list of manufacturing holon properties. These properties, upon which the holon definition rests, are also defined below.

Definition 1.2.2. Autonomy – The capability of an entity to create and control the execution of its own plans and/or strategies [100].

Definition 1.2.3. Cooperation – A process whereby a set of entities develops mutually acceptable plans and executes these plans [100].

Definition 1.2.4. Recursivity – A similarity in the informational architecture and communications model between holons [158]

Definition 1.2.5. Self-Organization – The ability of manufacturing units to collect and arrange themselves in order to achieve a production goal [100].

Definition 1.2.6. Reconfigurability – The ability of a function of a manufacturing unit to be simply altered in a timely and cost effective manner [100].

One should note that the last three properties directly contribute to the definitions of robustness, flexibility and adaptability found on page 7. Additionally, the holonic literature often refers to holonic attributes, and for clarity, its definition is given below ⁷.

Definition 1.2.7. Holonic Attributes: Attributes of an entity that make it a holon. The minimum set is autonomy and cooperation [100].

Finally, to clarify holonic systems in a manufacturing context, the definition of a holonic manufacturing system is given with that of a holarchy of which HMS are composed.

Definition 1.2.8. Holonic Manufacturing System: A holarchy that integrates the entire range of manufacturing activities from order booking through design, production and marketing to realize the agile manufacturing enterprise [100].

Definition 1.2.9. Holarchy: A system of holons that can cooperate to achieve a goal or objective. The holarchy defines the basic rules for cooperation for the holons and thereby limits their autonomy [100].

In the previous section, an initial conception of a holon was given in Figure 1. Having defined the necessary set of behavioral properties, attention can now turn to the elements of its composition which would allow it to achieve these behaviours. A holon is necessarily composed of a physical, hardware part and a decision-making software part which are connected by an intra-holon interface [64]. Additionally, each holon has both a holon human interface and an inter-holon interface for communication directed to achieving global objectives [42].

The physical hardware part of a holon is traditionally thought of as the manufacturing resources which one may find in a plant such as a milling or stamping machines [100].

⁷Throughout this review, the author will refer to holonic attributes and properties interchangeably.

Their control might be a combination of programmable logic controllers (PLC), local area networks (LAN), and PC's [100]. However, this conception is not sufficient. Firstly, not any machine can be readily included into a holon architecture. Specifically, it needs to be sufficiently flexible ⁸ such that 1.) it may interface with the software component of the holon 2.) meet the previously specified behavioural requirements [100]. Good examples of flexible manufacturing resources are CNC milling machines and robots [100]. The above conception is also insufficient because it does not include the raw material, works in progress (WIP), and final products to which a software interface maybe added to achieve holonic behaviour [270], [174], [41]. Finally, the above conception is insufficient because it does not account for the ability of a holon to exist within another holon like cells, shop floors, or factory holons [245].

The HMS consortium has almost universally accepted that the software part of a holon and its holarchy is enabled by agents and multi-agent systems [24]. Brennan [39] explains this relationship between holons and agents in great detail and concludes that multi-agent systems are a necessary component of any HMS implementation. However, there exist many (not necessarily congruent) ideas of the definition of an agent ⁹. The author resolves the apparent disagreement with a definition formed from Huhns [122] ¹⁰ and Durfee [83] ¹¹ which is found below:

Definition 1.2.10. Multi-Agent Systems – Loosely-coupled network(s) of active and persistent software components that perceive, reason and communicate together to solve problems that are beyond their individual capabilities.

The reader accustomed to object-oriented programming may note that agents are similar to objects in their focus on data abstraction, encapsulation, modularity and inheritance [33]. However, they do differ in that they possess a greater degree of autonomy to control their state and behavior, can demonstrate reactive, proactive, and cooperative behavior and have their own thread of control [260].

⁸How flexible a manufacturing resource in the holonic context is still an open research question, but the implemented architectures in Section 3.1.3 should shed light onto the degree of required flexibility.

⁹For example, Rzevski defines agents as "a software object that mimics the role of a competent personal assistant to perform a specific task on behalf of a user intelligently or not, independently or with little guidance [200].

¹⁰Huhns states that multi-agents systems are "a group of active, persistent software components that perceive, reason, and communicate."

¹¹Durfee states that multi-agents systems are "loosely-coupled network(s) of problem solvers that work together to solve problems that are beyond their individual capabilities."

This section has briefly described the necessary behaviour and composition of a generic holon. From this platform, Section 2 can give a rationale for the behaviour of holonic manufacturing systems and Section 3 can describe the technological developments of which it is composed. This section, however, has not given sufficient attention to the specific functionality of an individual holon or holonic manufacturing system. In later sections, it will be important to identify clearly the functionality of the proposed systems ¹² This given, this review restricts its discussion strictly to discrete part manufacturing despite the clear potential use of holonic and multi-agent manufacturing systems to Human-Machine Information Management [1], [2], Robotics [8], Cognitive Psychology [268], Automated Storage and Retrieval Systems (warehousing)[143], Product Design [66], [15] [69], [187], Enterprise Integration and Supply Chain Management [97], [185], [192], and Process Control Systems [60].

2 HMS Rationale & Its Requirements

The previous section introduced and defined holonic concepts within the domain of discrete part manufacturing. It also superficially stated the rationale for its emergence. This section describes that rationale in much greater detail. It uses a three part method to highlight how the holonic vision directly links the salient business drivers to the necessary manufacturing system attributes which in turn are used to develop the implied functionality requirements of its control system. The basic outline of this rationale is summarized graphically in the waterfall chart in Figure 2 below.

2.1 Business Drivers Rationale

Manufacturing firms find themselves in a continually evolving and increasingly competitive marketplace. In addition to consistently providing high quality products at low cost, they must realize that their excess capacity causes their marketing departments to gradually shift market power to the consumer [44]. As a result, to maintain market share, firms must address consumer demands for continual innovation, low-cost customization, and improved customer service [207]. These demands are satisfied by shortened product-life cycles, reduced time-to-market, and increased product variety at little or no extra cost or quality deterioration

¹²The author believes that many sources in the literature do not explicitly contrast the observed functionality of a proposed system to the functional requirements derived from the previously stated holonic behaviours.

complexity	*	*	*	*				
responsiveness					*	*	*	*
	standardisation	minimal system structure	intuitive/transp. structure	well-def./transp. behaviour	flexibility	reconfigurability	scalability	robustness
			*			*	*	*
		*	*			*	*	*
*			*	*	*			
			*	*	*	*	*	*
			*	*				*
			*	*	*	*		
	*	*			*	*		

Figure 2: A Waterfall Chart Linking Business Drivers to Control System Needs[175][44]

[175]. Their fulfillment manifests themselves in the firm’s manufacturing operations as 1.) more complex products, 2.) rapidly evolving product lines 3.) faster product introductions 4.) volatile output (& demand) and 5.) reduced investment per unit [175]. In short, the manufacturing firms must find ways to mitigate the increased complexity and continual change in its operations [44].

2.2 Manufacturing System Rationale

The difficulty in addressing the previously mentioned operations requirements can not be well understood without a clear understanding of the nature of modern-day industrially implemented manufacturing systems. These systems may be broadly categorized under the title of computer integrated manufacturing (CIM). Their limitations and drawbacks provide the motivation from which HMS research has emerged. The defining characteristics of these CIM systems are now presented so that later in Sections 2.2.2 and 2.2.3 we may understand the motivation behind the innovations and research that have followed.

2.2.1 CIM Systems

Much like the 1990's business environment outlined in Section 2.1, the 1980's experienced its own competitive drivers. Firms were under particular pressure to reduce their time-to-market and operating costs [108]. More specifically, firms found themselves paying excessively for large quantities of inventory and many layers of management personnel [108]. Manufacturing firms sought to achieve these objectives from the state of the art management techniques of the time like Just-In-Time (JIT), Total-Quality- Management (TQM) and World-Class-Management (WCM) [108]. All of these techniques were system-wide improvements and required a great deal of coordinated and organized data collection. With the emergence of cost-effective local area networks and personal computers, CIM systems were able to gain prevalence as the need to implement JIP, TQM, and WCM with integrated IT systems grew [108].

Harrington, the first to introduce CIM, captured its function in its original definition:

Definition 2.2.1. Computer Integrated Manufacturing (CIM) – The integration of business engineering, manufacturing and management information that spans company functions from marketing to product distribution [109].

This rather broad definition of CIM led to more reflective acronyms like the Computer Integrated Enterprise (CIE) and the Computer Integrated Manufacturing Enterprise (CIME)¹³, but they were not generally accepted [108]. This was partially because the integration of manufacturing operations remained at the core of CIM functionality. Explicitly stated, CIM systems integrated shop floor processes, their manufacturing engineering planning, and the production planning and control of the shop floor and its associated materials [109].

CIM systems achieved their previously mentioned objectives in a number of ways – all tied to the organization and coordination of data through the through networked PC [108]. For the first time, extensive use of the databases managed all of the functional¹⁴, product¹⁵,

¹³The European Commission sponsored programme ESPRIT endorsed this acronym.

¹⁴

Definition 2.2.2. Functional Data – Information categorized as part of a company's knowledge base [108].

¹⁵

Definition 2.2.3. Product Data – Data on products and their products [108].

operational ¹⁶ and performance ¹⁷ data of an enterprise [108]. This data was published with word processors, manipulated with spreadsheets, and transferred instantaneously through the network [108]. CIM also automated communication and data collection within the factory; thereby improving its speed and accuracy [199]. CIM, (in many instances), even eliminated the use of paper and its associated cost [108].

This degree of coordination affected the operations and interfaces of every department of the manufacturing enterprise. Marketing, engineering, production planning, plant operations, physical distribution and business & financial management were all improved [108]. While not all of these interfaces can be easily identified, Figure 3 systematically captures the dominant interactions between the firm's constituent departments [199]. Manufacturing firms achieved further productivity gains through the use of numerous inter-departmental PC-based software applications and algorithms [108]. Most notably, Computer Aided Process Planning (CAPP) sped up the preparation of routing and operation plans and Master Production Scheduling (MPS) planned the mixture of products for manufacture through the consolidation of data on existing and forecasted order quantities [108]. Materials Requirement Planning (MRP) managed the purchasing of materials through the extensive use of accurate inventory level information and products' bill of materials [199]. Additionally, Manufacturing Resource Planning (MRPII) created production schedules based upon the available manufacturing capacity [108]. Finally, firms were able to promote simultaneous or concurrent engineering through the use of Computer Aided Design and Computer Aided Manufacturing (CAD/CAM) packages [199]. The interested reader will find a number of thorough texts on CIM systems [199], [198], [133], [205], [108]. The results of this integration and automation led to reductions of design cost of 15-30%, reductions of in-shop part time of 30-60%, quality improvement and scrap reduction of 20-50% and improved product design through an increased number of design iterations [199].

CIM systems provided many benefits with their automated integrative functions but

16

Definition 2.2.4. Operational Data – The plans and instructions required to control the operations of a company [108].

17

Definition 2.2.5. Performance Data – Data that measures the degree to which operational or product data has been achieved [108].

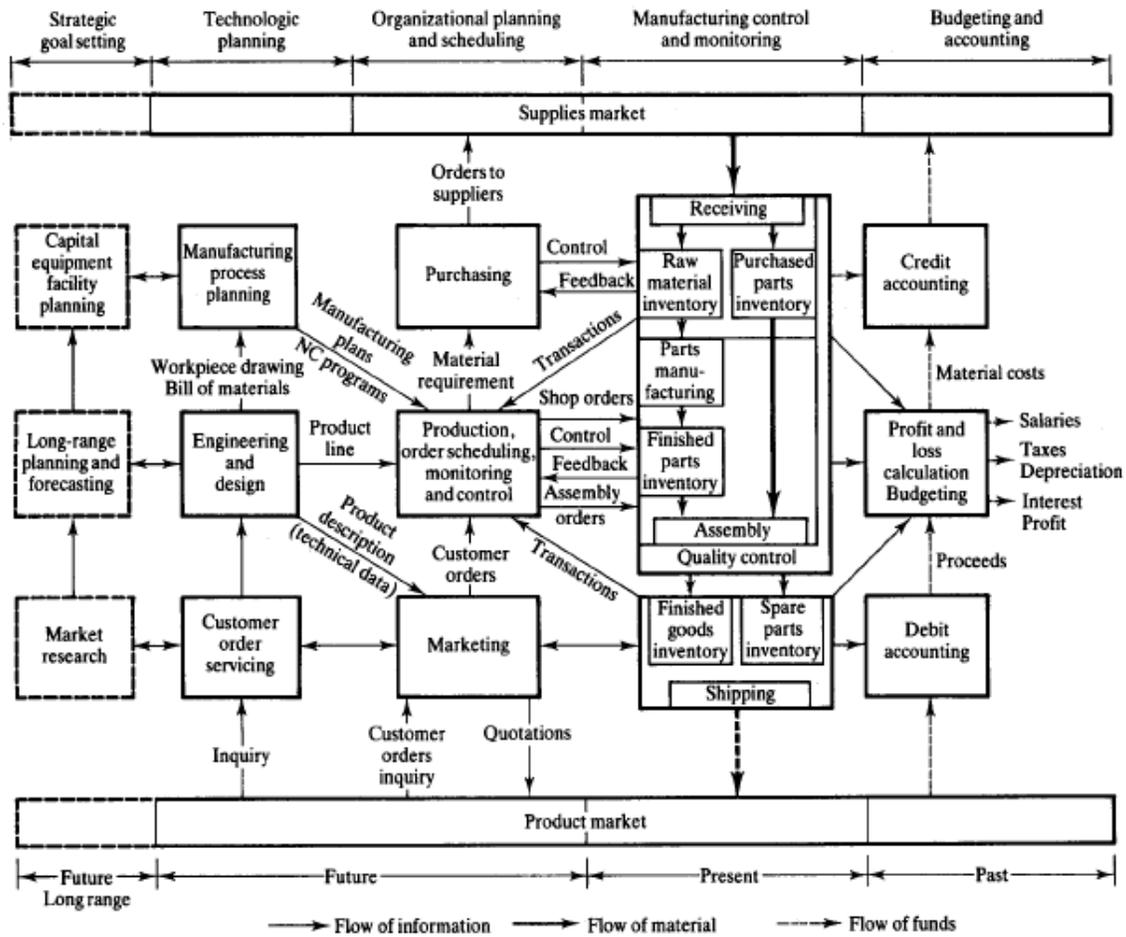


Figure 3: The Basic Functions of Manufacturing Facility [199]

proved to have poor agility [240] due to their hierarchical implementations [127]. The most well known CIM architectures proposed by IBM, the National Institutes of Sciences & Technology (NIST), Siemens, Digital Equipment, and the ESPRIT projects were all hierarchical [199], [78], [32]. A generic diagram of CIM hierarchy, seen in Figure 4, shows the generalized manufacturing tasks of planning, scheduling, execution, and control of machines & devices [172]. Each level has its own purpose and function [77]. Higher levels have increasingly complex computing structures and create the global goals and long-range strategies for all the levels (and computers) below them [77]¹⁸. Additionally, sensed information is abstract,

¹⁸Interestingly, the design community's experience and training in the use of hierarchical techniques probably aided the development of CIM as a hierarchical structure[77]. It was probably reinforced by the hierarchical structure of companies as responsibility and accountability are maximized when CIM layer corresponds directly to a personnel department [108]. Nevertheless, recent CIM developments have introduced modified

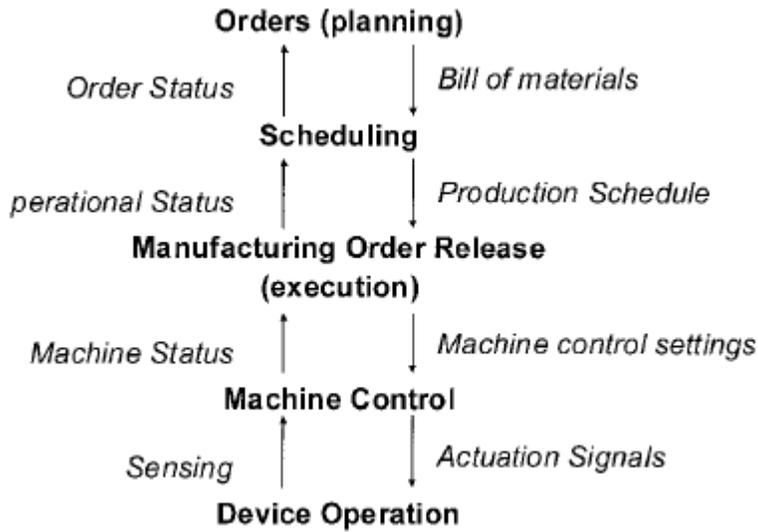


Figure 4: A Typical Manufacturing Control Hierarchy [172]

databases, time period.

Figure 5 show this hierarchical structure as an abstracted tree of parent-child nodes composed of master-slave communication relationships [28]. Commands flow purely top-

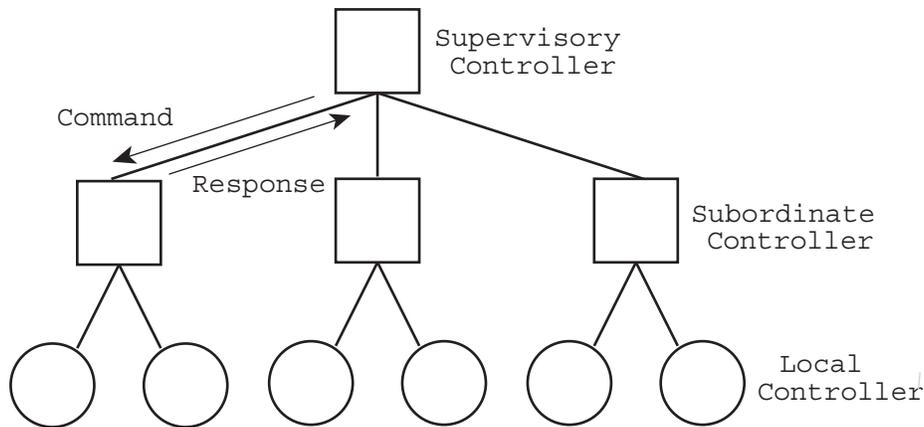


Figure 5: An Abstracted Hierarchical Structure [77], [29]

down and responses return in a static and deterministic fashion after task completion [29].

Beginning with the late 1980's, some began to think that hierarchical structures were not sufficiently flexible, and adaptable, and fault tolerant [196]¹⁹. The lack of flexibility results in hierarchical structures with increased distribution and coordination [29]. Newer texts on the subject also emphasize integration over hierarchy [108].

¹⁹

from the rigid communication relationships fixed in the early design stages of the CIM system [80]. Flexibility is further degraded when a great deal of complexity is introduced as each controller requires knowledge of its neighbours below and above to maintain these fixed relationships [80]. This complexity is amplified when explicit interrelationship exception handling is written to improve fault-tolerance. Fairley notes that a high degree of modularity and low degree of coupling facilitate implementation, debugging, testing, maintenance and complexity reduction [84]. Complexity also weakens fault-tolerance especially in the presence of disturbances and system modifications [49]. Additionally, CIM systems lack fault-tolerance because they are particularly sensitive to two failure modes. The failure of 1.) the controller or 2.) the communication link below it results in a paralysis of all the controllers to which they are connected [49], [116]. Finally, CIM systems exhibit poor adaptability because neither higher level controllers have access to detailed sensory information nor lower level controllers have the computational resources to make effective use of the data [77]. As a result, data is aggregated and thereby delayed in database formation [77]. In the presence of disturbances, the optimized global decisions of higher level controllers use obsolete or estimated data [77], [196].

This system wide lack of agility of CIM gained American political attention in the early 1990's [185] much like it had in Japan [230], [231]. As a result, numerous agile manufacturing initiatives outside of holonic manufacturing systems were begun [185]. McFarlane notes some of these initiatives to develop distributed and modular equipment: Reconfigurable Manufacturing (USA), Intelligent Assembly, Fixturing & Handling (Japan,EU) and Self-Diagnosis and Self Repair capabilities (EU/Aus)[171]. Unfortunately, it was soon realized that the implementation of flexible equipment and tooling was not sufficient and a comprehensive system wide view of agility was required [142], [216], [202].

2.2.2 Heterarchical Systems

A system wide approach to agility was proposed with the conception of heterarchical systems [28], [29]. Several researchers, [196], [80], [111], [209], [81], [79], sought to replace the rigidity of hierarchical systems with a completely flat structure where each component exhibits full local autonomy [77]. Decision making is made entirely locally at the point of information

Definition 2.2.6. Fault-Tolerance – The ability of a system to continue to function, perhaps in a degraded state, despite the occurrence of system failures [34]. It is a necessary requirement for robustness and adaptability.

gathering and measurement [81]. Furthermore, each component cooperates via a negotiation procedure in the form of temporary and flexible relationships [29]. Figure 6 shows an abstracted view of a heterarchical structure.

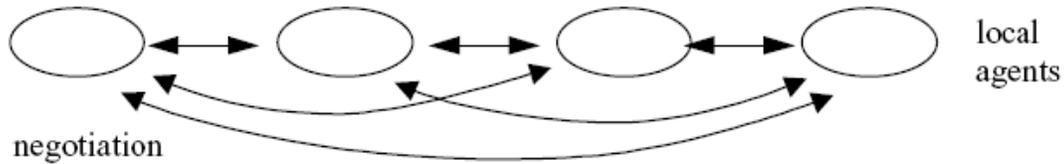


Figure 6: An Abstracted Heterarchical Structure [28], [29]

Heterarchy had numerous but ultimately insufficient advantages for industrial adoption. The full autonomy of each component in combination with the negotiation algorithm meant that failures did not propagate and the system exhibited fault tolerance [80], [224]. For similar reasons, the system could adapt readily to local disturbances [77]. These results were supported both theoretically [68], as well as experimentally [82]. Additionally, developer found that complexity, measured in lines of code, was reduced [80]. Finally, the system flexibility was excellent as the negotiation algorithm and system components could be modified, removed, and added easily [77]. Heterarchical systems, however, in their full autonomy lost the ability to create and achieve global objectives [80], [24]. Furthermore, Valckenaers showed that the system can reach unstable states where small disturbances induce large disturbances elsewhere in the system [240]. Additionally, hierarchical systems are limited by the network bandwidth as network traffic can grow quickly with the number of components [132]. Finally, heterarchical systems were never industrially adopted due to their inability to achieve a predictable result [28], [29].

2.2.3 Holonic Systems

The failure of heterarchical systems, and the need to resolve CIM systems' deficiencies opened the way for numerous new manufacturing concepts – loosely categorized as Next Generation Manufacturing Systems. In addition to holonic manufacturing systems, other concepts like bionic [183], [184], genetic [237], [238], fractal [250], random [124], and virtual manufacturing systems [129] were proposed [32], [245]. In addition to these references, the interested reader can find review and comparisons of these concepts in [244], [235], [236]. Interestingly, a number of authors believe holonic manufacturing to be among the better concepts [101],

[236], [143]. Bongaerts and Van Brussel state that holonic systems are preferable because they emphasize autonomous holons that maintain robustness and adaptability in the presence of disturbances, loose hierarchies that permit global optimization, flexible hierarchies that are reconfigurable and adaptable and migration strategies that facilitate industrial adoption [32], [245].

The qualities of autonomous holons and loose-flexible holarchy is a fusion of the advantages of hierarchical and hierarchical systems [32]. Hierarchy is introduced to not just permit global optimization but also to create stability in the system [245] as in the watchmakers parable told in [134], [222]. At the same time, these structural relationships are neither fixed nor permanent. Holons can: belong to multiple hierarchies, enter and leave them and do not rely on the proper function of their neighbours [32]. Figure 7 gives an abstracted view of a holarchy. Additionally, holonic systems have autonomous and cooperative qualities much

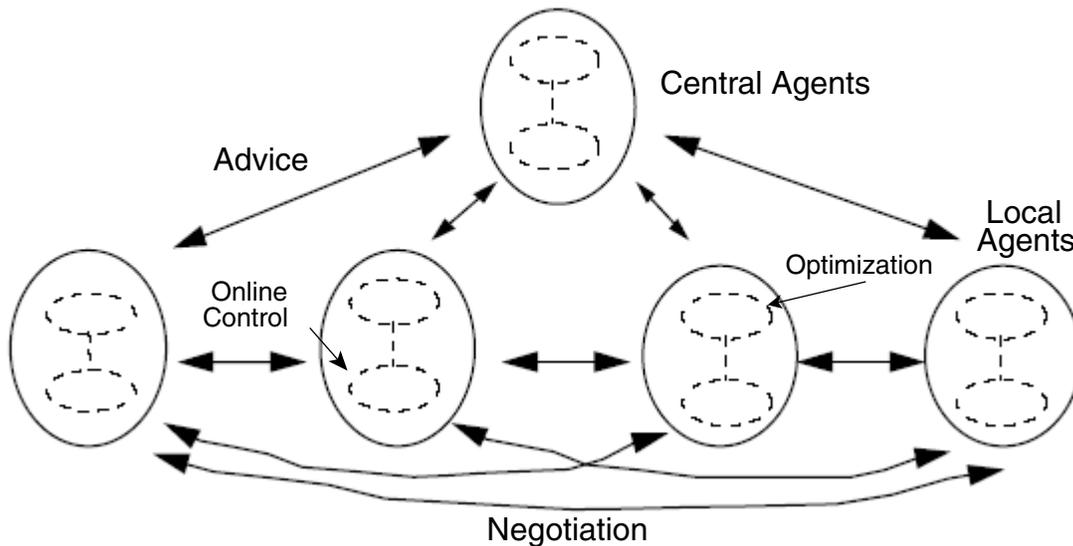


Figure 7: An Abstracted Holarchy Structure [28][29]

like heterarchical systems do[44]. They negotiate amongst each other and make local decisions, albeit with less autonomy than their heterarchical system counterparts [245]. More central holons, also have the ability to give advice or apply flexible rules to more peripheral holons [32]. In these ways, holonic manufacturing systems can mitigate most unwanted circumstances and become robust[135], flexible and adaptable [175].

2.3 Control System

Having studied the strengths and weakness of hierarchical and heterarchical systems, one can begin to specify more detailed requirements for a holarchy's behavior – its constituent control system. In this section, the author extracts a number of clearly stated requirements from Sections 2.2.1 and 2.2.2, compares them to recommendations elsewhere in the literature, and finally proposed the available technologies to achieve them.

2.3.1 Requirements and Recommendations

Section 2.2.1 showed that CIM systems were rigid due to fixed master-slave relationships and the complexity between them. From these observations, we can infer two requirements to improve flexibility and robustness:

Requirement 2.3.1. *Inter-holon communication relationships must be flexible and temporary.*

Requirement 2.3.2. *Complexity must be minimized by low-coupling, high cohesion holons.*

These requirements are expanded to a more generalized recommendation:

Recommendation 2.3.1. Control interactions should be abstract, generalized and flexible [175].

In other words, the control system must be distributed to achieve a holon's cooperative, recursive, and reconfigurable properties.

The above requirements also directly improve adaptability. Brennan explicitly states that a distributed systems can more effectively reject disturbances [39]. Adaptability, is furthered by two more requirements:

Requirement 2.3.3. *Time sensitive or time varying data must be used locally at the point of gathering or measurement. Time invariant data may be centralized.*

Requirement 2.3.4. *Time sensitive or time varying data must be available to the relevant decision maker be they local or central.*

which may be generalized to the following recommendation.

Recommendation 2.3.2. The architecture of the control should be decentralized and physically-based. [175].

Together, these requirements and recommendations may be used to support Parunak’s assertion that decentralized control must be ”emergent rather than planned, concurrent rather than sequential” [186]. Finally, Brennan encourages physically based holons and contrasts a physical decomposition to a functional one [39]. Weiss cites evidence of the former in natural systems where critical information is often organized by physical entities and managed by other physical entities that represent them [258]. Booch finds the latter to cause inconsistency and unintended relationships [33]. One sees that these arguments solidify holons’ autonomous properties within a flexible hierarchy [44].

Finally, Section 2.2.1 showed that CIM systems did not provide enough computation resources to lower level controllers. Hence,

Requirement 2.3.5. *Lower level controllers must have sufficient computational ability to make their designated decisions.*

This requirement can be expanded to two more recommendations:

Recommendation 2.3.3. The control should be both reactive and proactive [175].

Recommendation 2.3.4. The control should be self-organizing [175].

One may note that this comprises a fundamental upgrade of previously held paradigms of adaptability. Not only must holons respond to disturbances, but they must also self-organize and be proactive towards achieve their global and local objectives. In this way, one may conceive the harmonization of all of the previously mentioned holon properties.

2.3.2 Available Technologies

Section 1.2 noted that the software part of holons must necessarily be an agent. Multi-agent systems in manufacturing are not an entirely unproven technology. They have been used in the past in systems where ”data, control, expertise, or resources are distributed” or when legacy systems need to be inter-compatible [263]. Parunak explains that a primary advantage of agents is that not only can they decide locally as subroutines do but they also initiate their own (proactive) activity – thus enabling the holon properties [186]. In this way, agents give a framework for handling uncertainty, inconsistency, security and optimization [262], [261]. Parunak continues by stating that agents much like manufacturing entities have well-defined state variables that are distinct from their environment [188]. Hence, it is logical to make agents correspond to machines, tools, fixtures, products, parts, features and operations

²⁰[39]. More about agents can be found in Section 3.1.1.6 as well as on the FIPA website [87].

The intelligence of agents and the resultant agility on the manufacturing system creates new demands on the machine and device control in the shop floor. Specifically, these demands required lower level controllers to have real-time capability [16].

Definition 2.3.1. Real Time System – A systems where the its correctness depends not only on the logical result of computation but also on the time at which the results were produced [229].

Real-time systems (in order to exist) provide not just the necessary speed but also performance predictability [17]. The execution of tasks such as robot movements can be started and finished with temporal certainty. Real-time systems also implicitly demand reliability from their components to achieve this level of predictability [16]. They can also provide the necessary adaptability by sensing and responding to unexpected events [17].

The communication demands and responsiveness across a very distributed and decentralized architecture such as a holonic manufacturing system will required integral use of an internet backbone [269]. The method of utilization may vary from system to system but it is fundamental. For example, mobile agents are one innovative and effective use of the internet in HMS [90].

3 Developments in Holonic Manufacturing Systems

Having introduced holonic systems in Section 1, Section 2 gave an extended rationale for their adoption. Specifically, it showed how holonic systems aim to fulfill the faults of other manufacturing systems. It also rationalized holonic systems' requirements and outlined general directions of technology development and implementation. This section builds upon this foundation with a review and elaboration of these development paths found in the literature. Throughout the course of the review, care is taken to evaluate the degree to which the holonic rationale and its requirements are fulfilled. The survey begins with the most prominent of the proposed system architectures in increasing degree of implementation. Next, methodologies for design and development of these architectures are discussed. Finally, the protocols, algorithms, and interactions upon which these architectures and methodologies are founded are investigated.

²⁰Interestingly, much agent research has used functional agent decompositions as in [66], [167], [192], [63].

3.1 Architectures

The architecture section proceeds from a very general picture of holonic systems to one of greater detail. In other words, it investigates the most prominent conceptual descriptions and shows the manner in which realized implementations compare. Numerous architectures have been proposed as a result of the IMS feasibility program, and there is significant diversity, variation, and even disagreement in both conceptual descriptions and realized implementations. Additionally, a wide variety of architectures may be different at first glance but their differences may be resolved. To aid the review, the architectures must be differentiated into holon architectures that describe solely intra-holon components and HMS architectures that describe the component holons and their interactions. Laws notes that the literature is less than clear in making the distinction [141]. This review also attempts to highlight the repeatability, distinguishing features, and validation/evaluation processes (if any) of the mentioned architectures ²¹.

3.1.1 Conceptual Description

The first architectures published in the early and mid-1990's published as a result of the IMS feasibility program were purely conceptual in nature. They improved upon the previously stated rationale solely with a description of a holonic manufacturing system's component elements. In so doing, they formulate the system's logical boundaries, clarify its requirements and give further rationale for its implementation.

3.1.1.1 Initial Conceptions :

After the original conception of applying holonic principles to a manufacturing context [230], [231], and the creation of the HMS project [207], [112], Holonic Manufacturing Systems lacked a widely accepted concept of a holarchy structure and its individual holons. To solve this problem, Christensen proposed an initial holon architecture found in Figure 8 [62]. It may be viewed as the predecessor to the more detailed conception previously shown in Figure 1 on page 9 [61]. He also reiterated the holarchy structure found in Figure 7 on page 20 [61]. Finally, he attempted to standardize the nature of the holon interfaces that constitute a holarchy with Standards Theme Tasks (STT's), as shown in Figure 9 [61]. At this point in HMS research, architectures were still very much conceptual but Christensen's work did serve as the bedrock upon which other researchers' works could coalesce.

²¹The validation/evaluation of an HMS architecture has been found to be non-trivial [264]

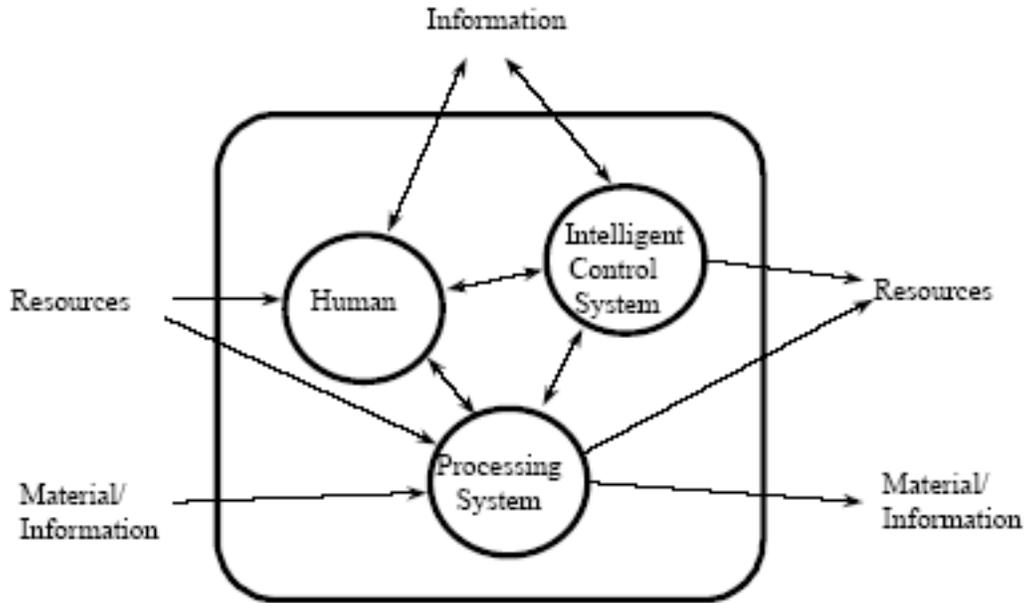


Figure 8: A General Activity Model of a Holon [61]

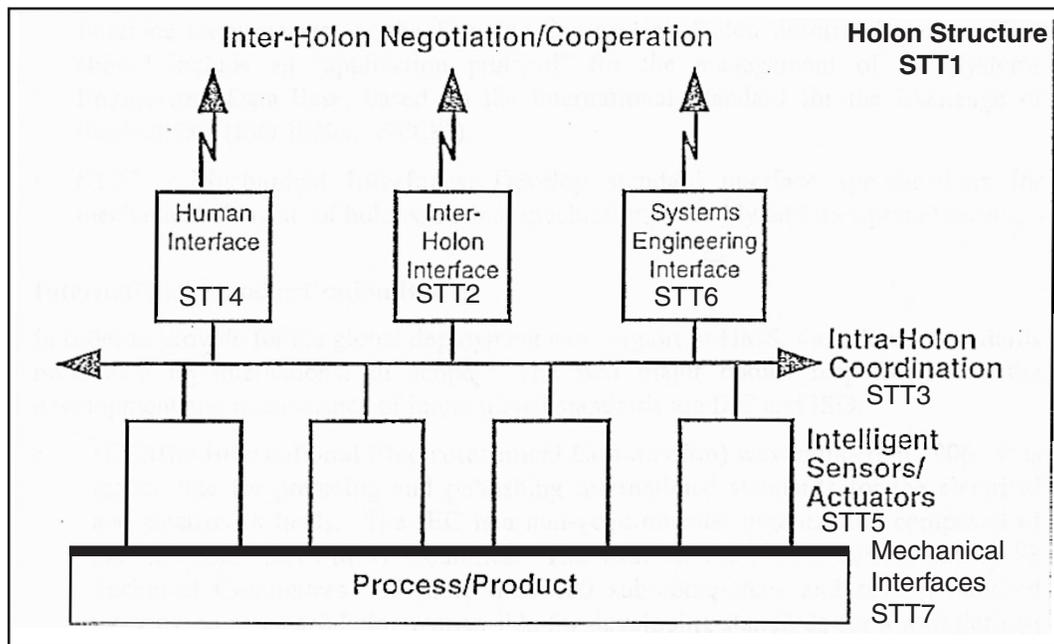


Figure 9: The Critical Holon Interfaces in Holonic Manufacturing Systems [61]

3.1.1.2 Cooperation Domains :

The cooperation domains paradigm was originally conceived soon after the beginning of

the HMS research program [72], [71], [73], [74]. Since then it has progressively developed and at times absorbed newly developing technologies [95], [91]. Nevertheless, its basic idea remains the same. Cooperation domains are where holons may location, contact, and interact with each other. In that sense, it may be viewed as a holarchy. Cooperation domains contain shared data structures and facilities for message passing. It also includes decision making and monitoring mechanisms that support the holon’s local activities. It also includes techniques and rules for treating compound holons. The research into cooperation domains solidified some of the more abstract descriptions before it and paved the way for more detailed architectures [95].

3.1.1.3 PROSA – Product, Resource, Order, Staff Architecture :

Soon afterwards, probably the most well-proliferated conceptual architecture was proposed. Called PROSA, it decomposed the shop floor into at least product, resource, and order holons as shown in Figure 10 [246]. The resource holon contains any physical resource required for

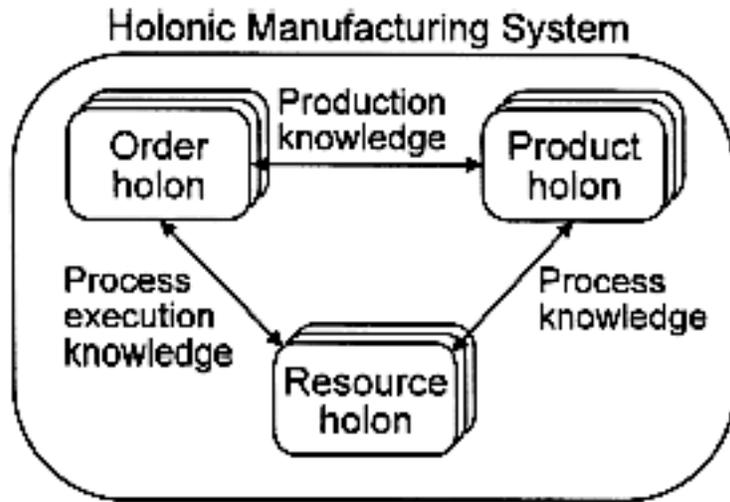


Figure 10: Basic Buildings Blocks of PROSA Architecture [246]

manufacture at any level of production, i.e. raw material, robots, furnaces, AGV’s, a whole shop floor, or even an entire factory [245]. The product holon, holds the process²² and product knowledge²³ required for the proper manufacture of a given product [266]. In other words, it contains the end products of the traditional product design, process planning

²²Process knowledge is information on how to perform a certain process on a specific resource [246].

²³Product knowledge is information on how to product a certain product using a specific resource [246].

and quality assurance departments [246]. The order holon represents a task in the holonic manufacturing system and manages the physical product proceeding through the system and maintains the process execution knowledge [267]²⁴. In addition to these holons, an optional staff holon may be added [32]. It provides the basic holons with supplementary information to make their decisions correctly [245]. It may be used to implement centralized algorithms deemed too difficult for a distributed solution [246]. PROSA also allows for specialization of the basic holons based upon their inherent characteristics, i.e. robot, CNC machine and conveyor resource holons [243]. The PROSA architecture also aggregates a large number of low-level holon in order to minimize its associate complexity [246]. This is done intuitively much like modern day plants aggregate machines into stations into cells, into shop floors. For example, Figure 11 shows how specialized product, resource and order holons may be embedded within a larger workstation resource holon.

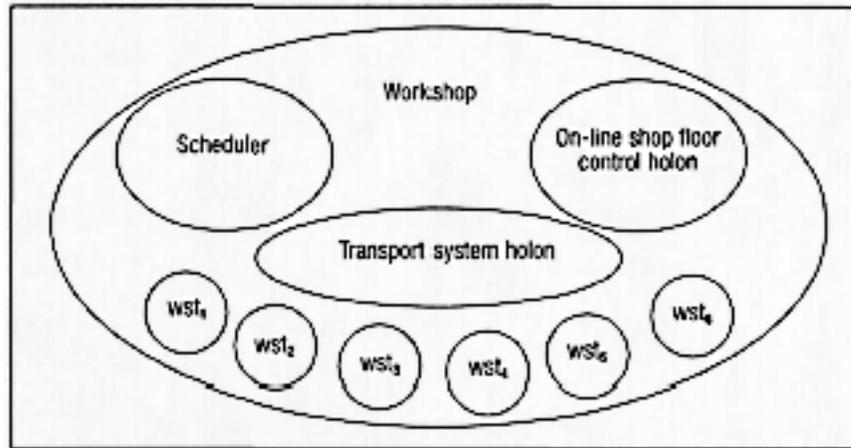


Figure 11: Aggregation and Specialization in PROSA Workshop [245]

3.1.1.4 HCBA – Holonic Component Based Architecture :

HCBA, much like PROSA, can serve as a template for numerous implementations and applications of holonic systems [58]. It also uses a product and resource holon that are similar to those of PROSA [55]. The similarity, however ends there. HCBA grew from the application of Component Based Development (CBD) to HMS [56]. CBD is a software programming paradigm that is based upon plugging reusable and reconfigurable components together [233]. These components are given holonic attributes and go onto to form HCBA's holons [58]. In

²⁴Process execution knowledge is information on the current progress of a processes execution [246].

HCBA, the intra-holon structure is well specified from the bottom up and is shown in Figure 12 [55]. Each holon has a components that corresponds to machine, cell, and factory level

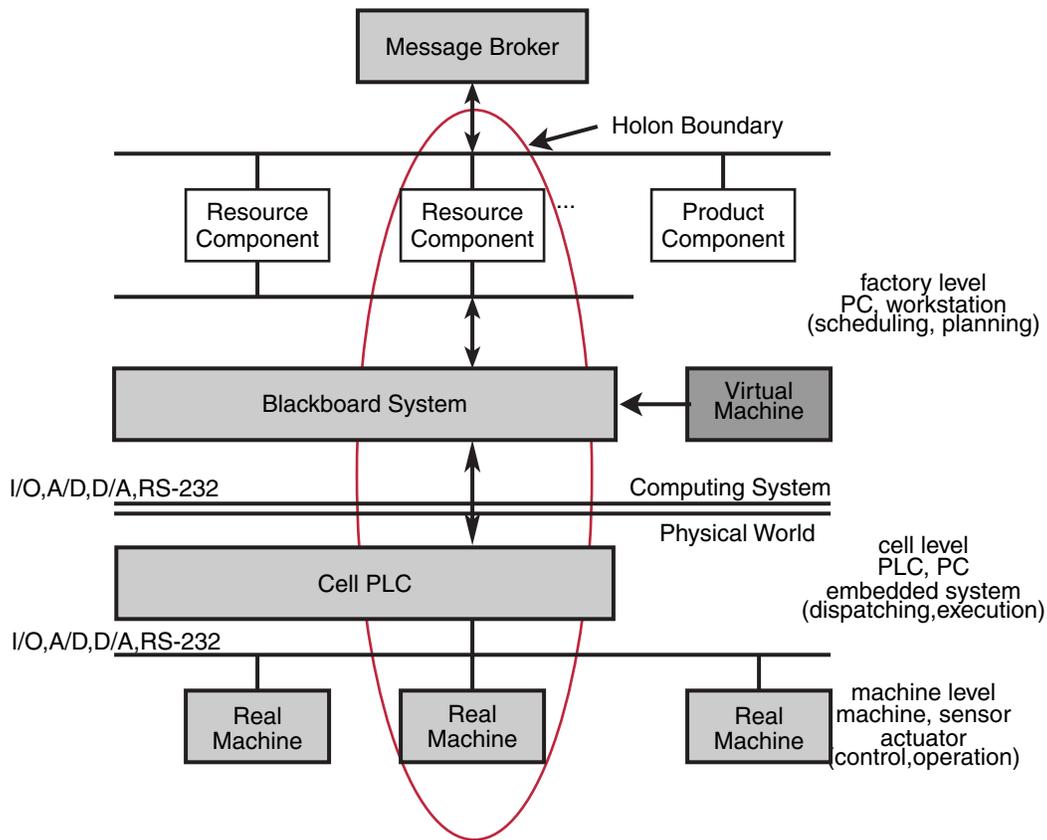


Figure 12: HCBA Holon Architecture [54]

components in conventionally controlled CIM systems [58]. Interestingly, this correlation is the foundation of a migration strategy to be introduced in section 3.2.3.1 [57]. Between the cell and machine levels, or in other words, at the hardware-software boundary, exists the BlackBoard System (BBS) [55]. The BBS is a cell-wide system that mirrors the real-time relays of the cell PLC and each holon is allocated a "slice" to control its respective hardware resource [58]. The BBS has the added advantage of providing a simulation interface for a virtual machine [55]. The message broker (MB) is responsible for inter-holon communication is one instantiation of the broker agents to be described in Section 3.1.1.7.1 [58]. The full holarchy of HCBA continues to use the CBD paradigm within a nested structure as seen in Figure 13 [55]. A manufacturing firm is decomposed into nested business, factory, cell and machine levels which double as the firms' resource holons [58]. At each of these levels, also exists at least one product holon which dynamically spawns a WIP agent which coheres to

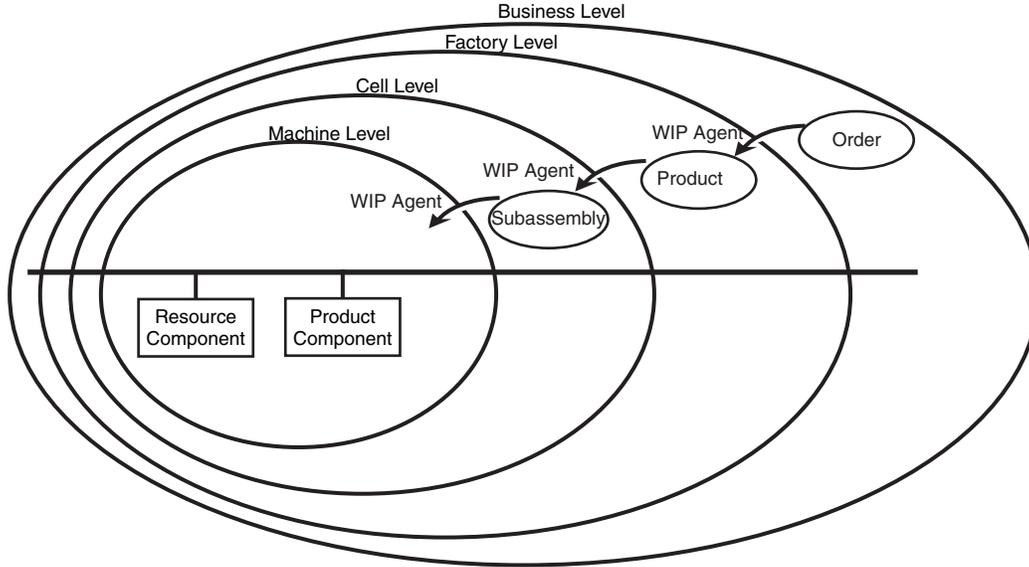


Figure 13: HCBA System Architecture [58]

and escorts the product holon at the lower level [55]. For example, at the lowest level, a subassembly creates a WIP agent for the machine level, which in turns escorts parts to and through the machines for operation. This architecture was implemented at the University of Cambridge’s Robot Assembly Cell and is further discussed in Section 3.1.3.2 [56].

3.1.1.5 HoMuCS – Holonic Multi-Cell Control System :

HoMuCS is a generalize HMS architecture developed purely to create physically implemented PROSA architectures [138]. This development process is described later the HoMuCS methodology in Section 3.2.2.2. The functional models describe the generic holon functionality as shown in Figure 14 [138]. In addition to autonomy and cooperation rules, the functional models describe a holon’s basic functionality which includes dispatching, monitoring, planning, and auxiliary functions [139]. These functional models drive the functional requirements for the object oriented models that form the architectures’s basic building blocks [138]. From these models, product, resource and order holons are created [138]. The product state models provide a template for centralized databases of product info [140]. The HoMuCS architecture is explained further detail in [137] .

3.1.1.6 Agent Architecture Taxonomy :

A vast number of multi-agent architectures exist in the literature and to review them would

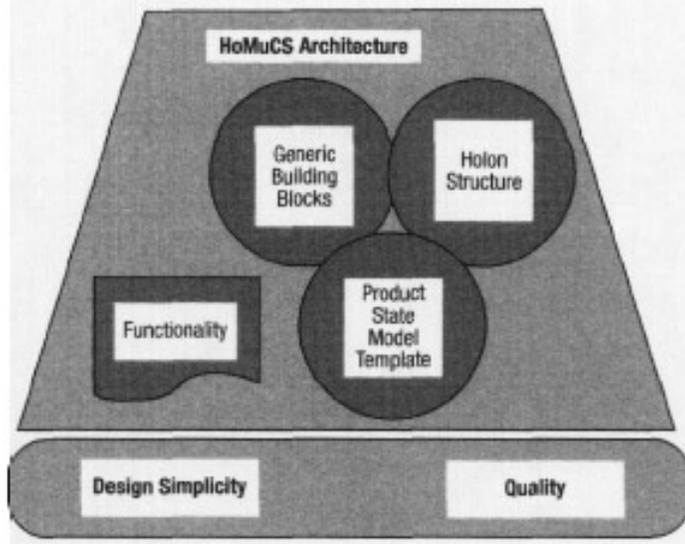


Figure 14: Outline of HoMuCS Architecture [138]

be outside of the scope of this report. The objectives of the multi-agent system do not necessarily align with those of HMS [152], but it does nevertheless provide many of the software developments necessary for the implementation of HMS [39]. Hence, a basic agent and multi-agent system taxonomy with examples is provided for the reader of holonic manufacturing systems. A much more thorough review of multi-agent systems can be found in [216] and [217]. Additionally, attempts to standardize these agent and system architectures have been made [87][181].

3.1.1.6.1 Reactive Agents :

Maes defines single agent architectures as:

Definition 3.1.1. Agent Architecture – A framework that specifies how the agent can be decomposed into the construction of a set of component modules and how these modules should be made to interact. The total set of modules and their interactions has to provide an answer to the question of how sensor data and the current internal state of the agent determines the actions and future internal state of the agent [151].

Three basic types of agent architectures conform to this definition: reactive, deliberative and hybrid[141]. Reactive agents have a pre-specified action for every possible sensory input (from the environment or other agents), and if there exists more than one sensed input simultaneously, the actions or behaviours are completed in a pre-specified order of importance

[43]. Brooks' Subsumption Architecture is one such example [40], and is graphically represented in Figure 15. Intelligence is an emergent property of multi-reactive agent systems,

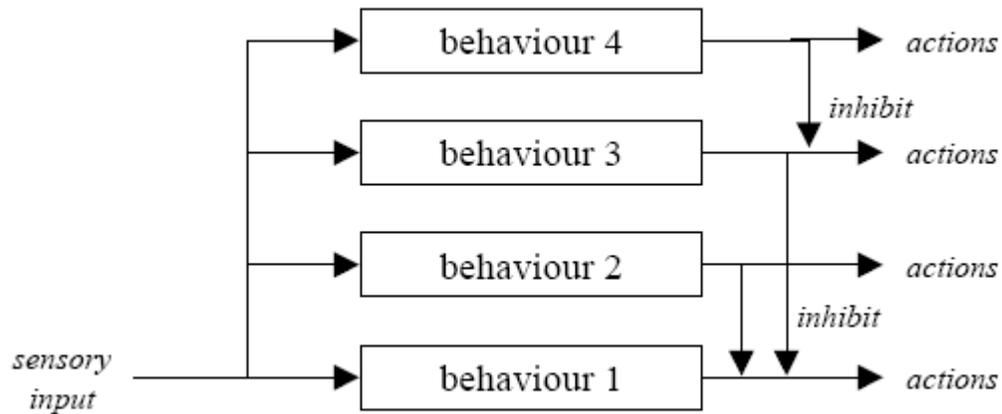


Figure 15: A Subsumption Agent: A Reactive Architecture [43][40]

but in general pro-active or goal directed behaviour is difficult to achieve [141].

3.1.1.6.2 Deliberative Agents :

Deliberative agents may be viewed as the anti-thesis to reactive agents. They explicitly represent goals (or desires), formulate beliefs based upon accumulated sensory input, and develop plans (or intentions) to achieve those goals [43]. Rao's Belief-Desire-Intention (BDI) architecture provides a good example of deliberative agents [197], and is diagrammatically shown in Figure 16. Deliberative agents can be both reactive and proactive but their implementation is impeded by the inability to decompose real-world events into an accurate symbolic representation [141]. Additionally, they may have difficulty responding in a timely fashion as they must both collect sensory input over time and compute through what may be a very complex logic [43].

3.1.1.6.3 Hybrid Agents :

Hybrid architectures incorporate elements from both reactive and deliberative agents usually into a layered architecture [141]. Lower levels reactively respond to sensed inputs while higher levels formulate plans and cooperate with other agents [43]. One example is Muller's InteRRaP architecture [177] and it is diagrammed in Figure 17. Hybrid agents are reactive, pro-active, autonomous and cooperate with other agents [141]. However, there remain difficulties in coordinating the inter-level interactions. Additionally, as a whole agent ar-

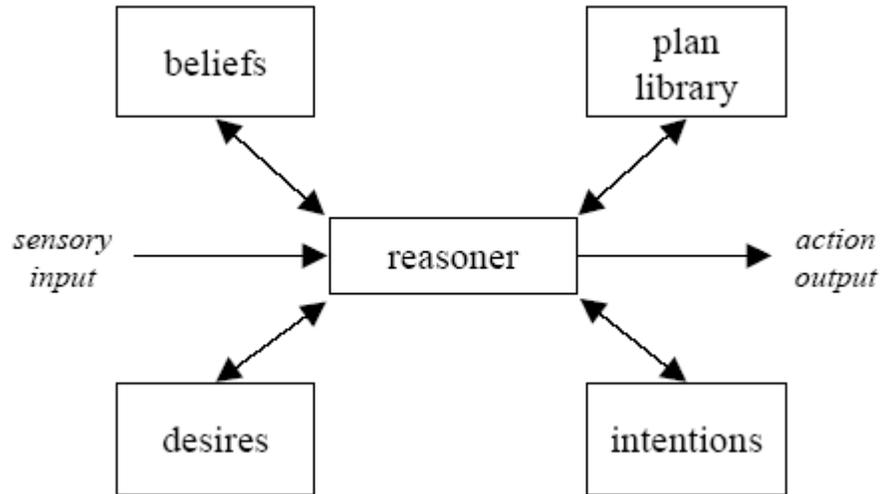


Figure 16: A BDI Agent: A Deliberative Architecture [43][197]

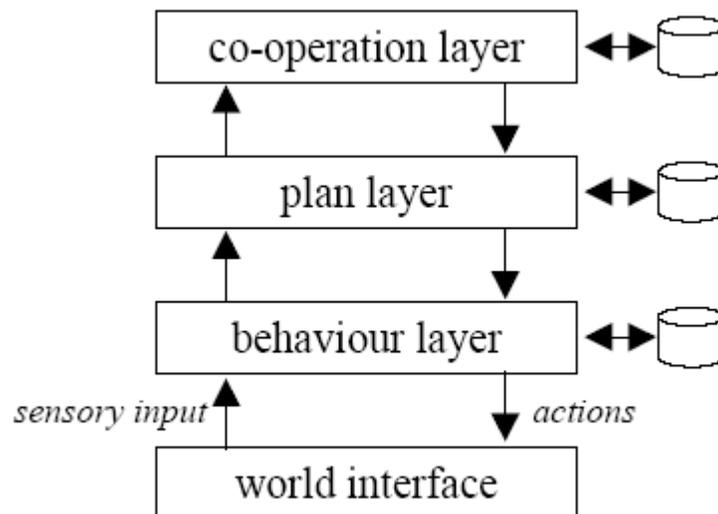


Figure 17: An InteRRaP Agent: A Hybrid Architecture [43][177]

chitectures lack a well formed design methodology [262]. Further examples of hybrid agent architectures may be found in [88].

3.1.1.7 Multi-Agent System Taxonomy :

There are many ways to group these agent architectures into a cohesive multi-agent system. These architectures may be divided into hierarchical, federation and autonomous agents

approaches of which latter two are of greater interest in holonic manufacturing systems development [216]

3.1.1.7.1 Federation Architectures :

Federation architectures are further subdivided into facilitator, broker and mediator architectures but all have the defining feature of a central agent that coordinates the actions of the others [141]; much like a the staff agent in PROSA. For this reason, Weiss states that they may be implemented in HMS [258]. In the facilitator variation, facilitator agents receive incoming messages from a predefined group of agents and then translate, process and route these messages to the appropriate agent [211]. A diagram of the Desktop Utility Agent System is provided as an example in Figure 18 [65]. Broker architectures, as seen in Figure 19,

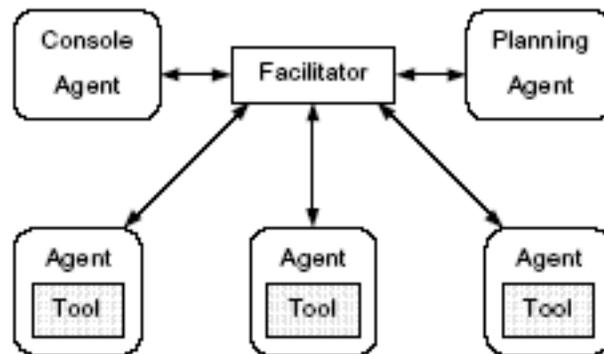


Figure 18: A Desktop Utility Agent System: A Facilitator Architecture [65]

are similar to facilitator agents but they also add monitoring and notification functionality and may be contacted by an local agent in the same system [211]. Mediators include all the

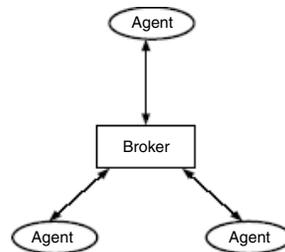


Figure 19: An Example Broker Architecture [211]

functionality of broker agents and extend their ability to coordinate the actions of more local

agents[141]. For example, they may use the recruiting mechanism as seen in Figure 20 to establish communication links between agents to later allow agents to communicate directly for the duration of their relationship [211]. Further examples federation architectures may be found in [21],[99], [166], [165], [164], [163], [162], [189], [219], [215]. The most prominent and perhaps most applicable of these is the metaMorph architecture developed at the University of Calgary [19], [17], [16], [18], [20], [36], [35], [168], [169], [214], [213], [210], [218].

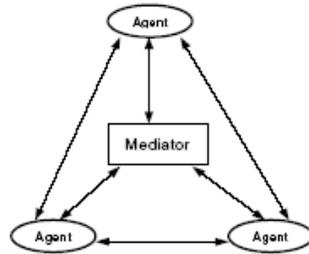


Figure 20: An Example Mediator Architecture [211]

3.1.1.7.2 Autonomous Agent Architectures :

Autonomous agent architectures are different in that they do not rely on any centrally located agents [216]. In this sense, they are similar to the lowest negotiation levels of HCBA or PROSA (without staff holons). Shen defines them by their functionality. They must: 1.) not be controlled by another software agent or human being 2.) be able to communicate directly with and have knowledge about the environment and other agents 3.) have their own sets of goals and motivations [216]. Figure 21 gives a conceptual diagram [211].

3.1.1.8 Miscellaneous Conceptions :

A number of other conceptual architectures have been proposed to the HMS research community, but in general, they have yet to be deeply investigated [130], [178].

3.1.2 Detailed Design and Simulated Implementations

Section 3.1.1 provided many conceptual HMS architectures whose primary purpose was to delineate the logical boundaries of HMS components. This section describes holonic-type designs and simulated implementations. Specifically, they are based upon the full specification of an industrial problem into an HMS simulation. As they were developed at the same time

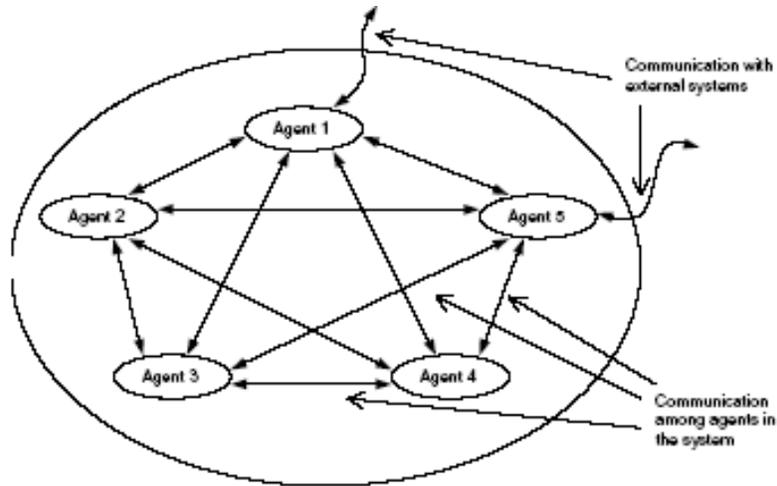


Figure 21: A Conceptualized Autonomous Agent Architecture [211]

as the conceptual descriptions, they often do not follow the same logical structures. Nevertheless, they do clarify some of the practical issues in HMS development and demonstrate some effective tools used in their resolution.

3.1.2.1 Rockwell/BHP Steel Rod Mill Water Cooling System :

The simulation of holonically controlled water cooled steel rod mill was one of the first tractable implementations of holonic principles. In a steel rod mill, hot steel coming out of the furnace is rolled, water cooled and then air cooled to a set temperature on a laying head as seen in Figure 22. The series of five water coolers is represented as separate negotiating holons [4]. The goal of the simulation was to demonstrate the system's performance, robustness, autonomy, and flexibility [4]. Each cooling control holon was given functionality as illustrated in Figure 23 [170]. More physically, each cooling holon controlled a water cooling box, and a valve that restricts water inflow [170]. The system was simulated in a number of conditions: normal, one failed holon, one added holon, and modified temperature sensor [4]. With respect to the demonstration goals, the simulation showed that high performance was achieved in normal conditions and that robustness was maintained in partial failure conditions [4]. Additionally, the system demonstrated a high level of agility to new hardware configurations [4].

3.1.2.2 VTT Automation Robot Cell :

In 1997, VTT Automation designed a simulated holon manufacturing robot cell based upon

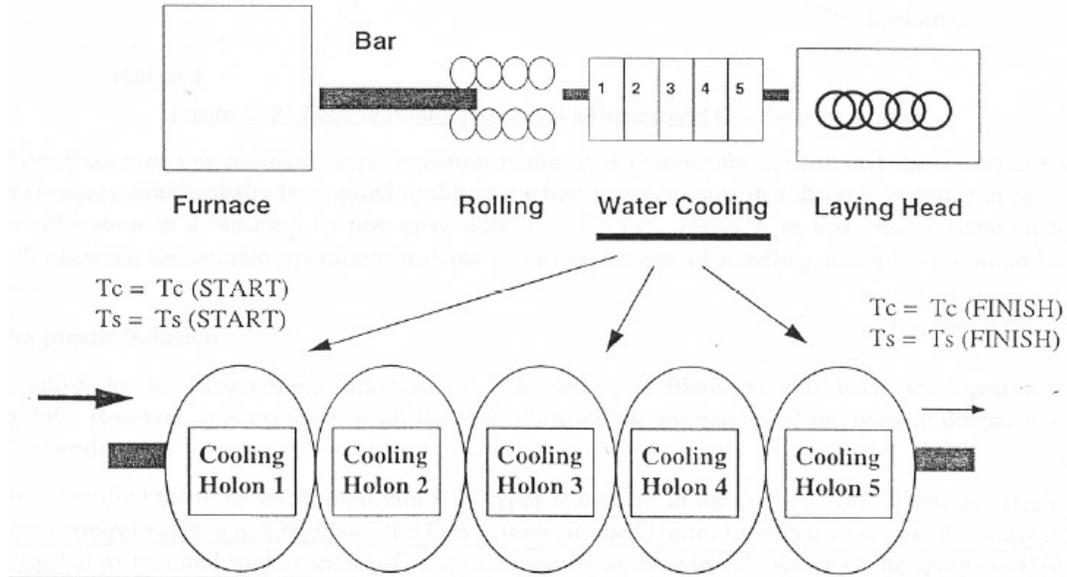


Figure 22: Water Cooling System Schematic [4]

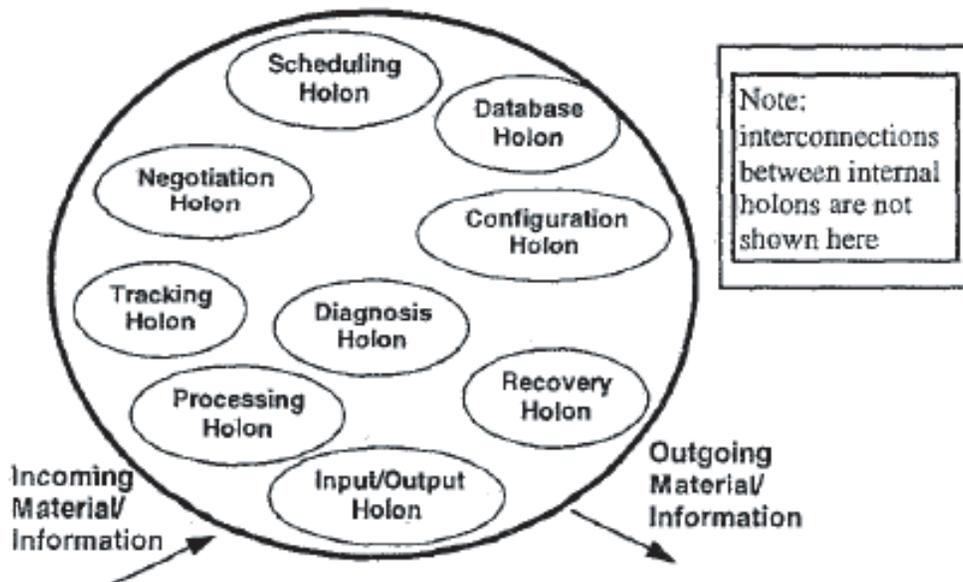


Figure 23: Holon Architecture [4] [170]

groundwork found in [115] and [113]. The distinguishing feature of this simulated architecture was the modularity and automation of their design implementation. The simulation was carried out within control and geometric environmental models; each residing on separate workstations connected by TCP/IP based LANs [114]. The geometric environmental model

residing on a Silicon Graphics Indigo Elan workstation uses the IGRIP 3D simulator software to create geometric models of the manufacturing devices [123]. It also uses the Graphical Simulation Language (GSL) code to manage the (virtual) machines' state changes which are passed to the control model via TCP/IP sockets [114]. The control model is a graphics and text based and was developed using the Structured Analysis for Real-Time Systems (RT/SW) approach [257]. The CASE-tool SA-PROSA ²⁵ software package was used to create design graphics of composed of data flow and state transition diagrams [149]. These diagrams are later compiled directly into VHDL ²⁶ code using the VELVET compiler[182]. The VHDL code is then run on a Sun Sparc II workstation using the VHDL2000 simulator [194][114]. Further detail about the software development can be found at [136].

From a holonic manufacturing system point of view, the VTT simulation sits on a firm architectural base. As the work is in simulation, they specify the software components of the holon, but do take care to define the intra-holon machine interface as well as the inter-holon negotiation interfaces. They propose an agent is composed of proposer, planner, requestor, executor, monitor, and controller modules [113] which is an extension of the PEM concept proposed earlier in 1992 [115] ²⁷. A diagram of the basic function of this agent is shown in Figure 24 below. The interested reader is referred to the original text for further detail of the intra-agent interactions [114]. The agent architecture above is applied to each of the devices in the manufacturing robot cell: two Aitec ARS10 robots (with grippers, arms, and vision components), one injection molding machine, one welding machine, one conveyor, and two Automatic Guided Vehicles (AGV) [114]. The cell is graphically represented in Figure 25 [114]. Interestingly, Heikkila notes that the agents can be implemented in a hierarchical, heterarchical and multilayered structure, and ultimately the heterarchical structure is implemented [114]. A rough sketch of this structure is show in Figure 26 [114]. This simulation demonstrated controlled and resilient behavior in the presence of faults, and adapted to unpredictable situations effectively [114]. Heikkila does note however that network traffic grew rapidly with the number of holons (perhaps due to the heterarchical structure) [114]. This was corrected by redirecting messages to only a subset of holons deemed relevant a priori [114]. This may suggest the need for the aggregation or hierarchy found in the PROSA or

²⁵This software is not related to the PROSA Holonic Manufacturing System Architecture.

²⁶VHDL is the Very high speed integrated circuit Hardware Description Language [114]. A thorough text can be found at [144].

²⁷The PEM Model is composed of Planning Execution and Monitoring and are widely considered to be three of an agent's primary tasks [115]. This is also consistent with the basic holon functionality in the HoMuCS architecture [138].

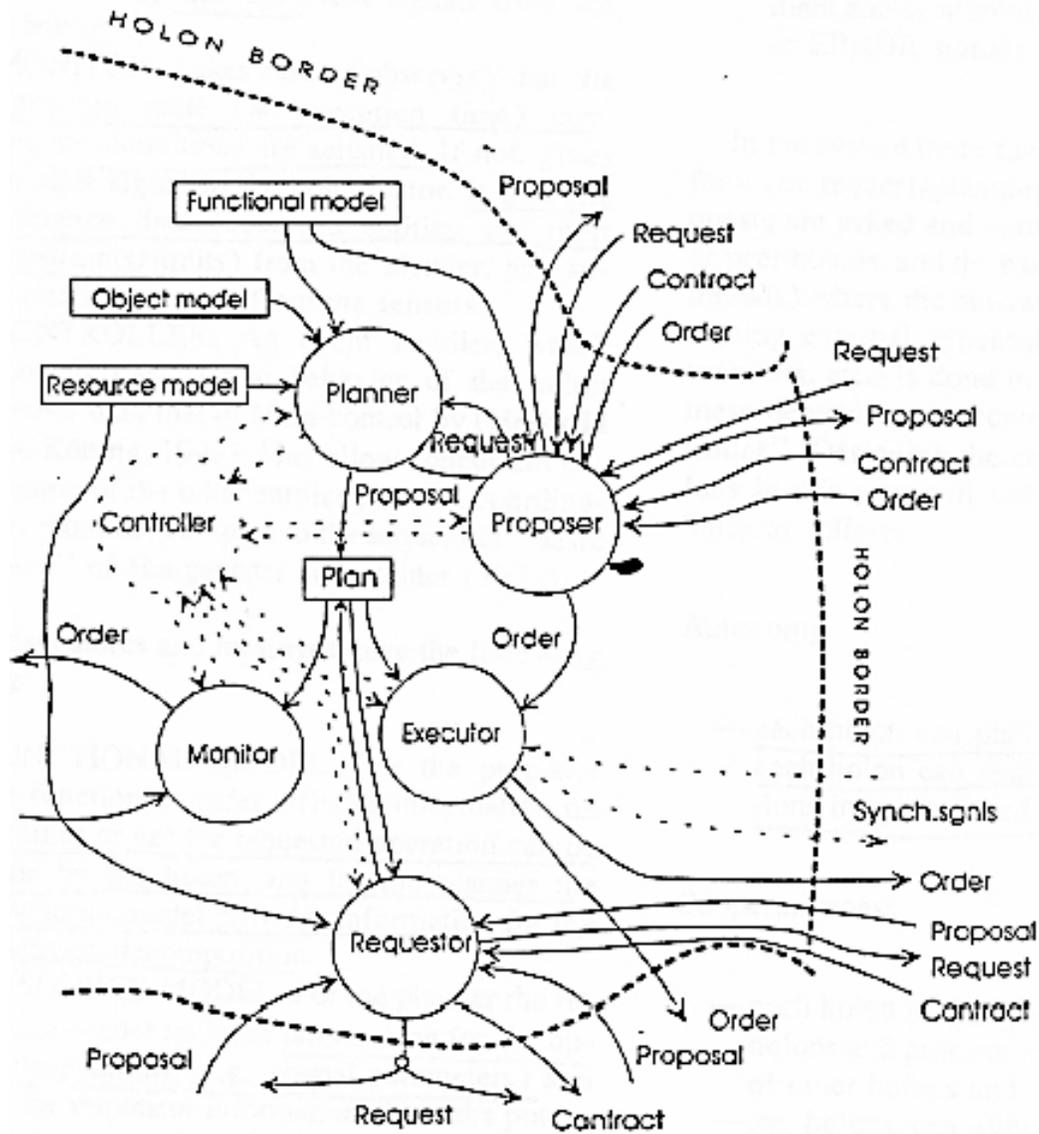


Figure 24: The VTT Automation Holon Structure [114]

HCBA respectively.

3.1.2.3 Holden's Engine Operations Engine Assembly Line The Australia based Holden's Engine Operations (HEO) company implemented a simulated partial holonic manufacturing system for an engine manufacturing line using Simjava [176] and an Allen-Bradley S11 5/03 PLC²⁸ [101]. Their simulation requires a new control paradigm called Part-Oriented-

²⁸HEO also decomposed the control problem in detail for a crankshaft machining state called the Huller-Hille machine [101]. However, the detailed design is neither simulated nor implemented.

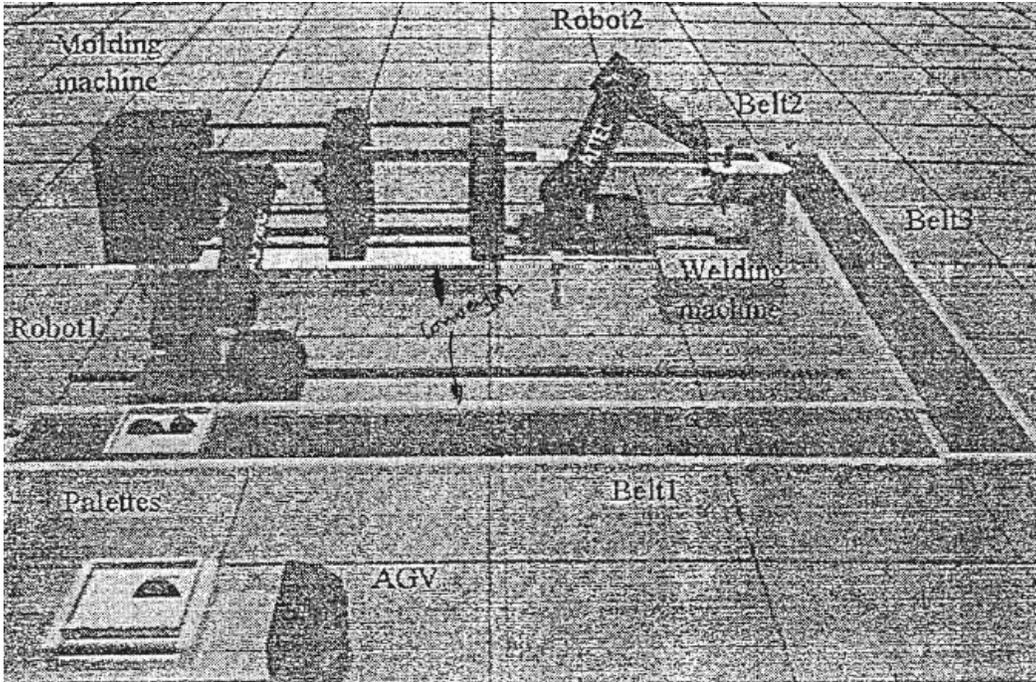


Figure 25: The VTT Simulated Manufacturing Robot Cell [114]

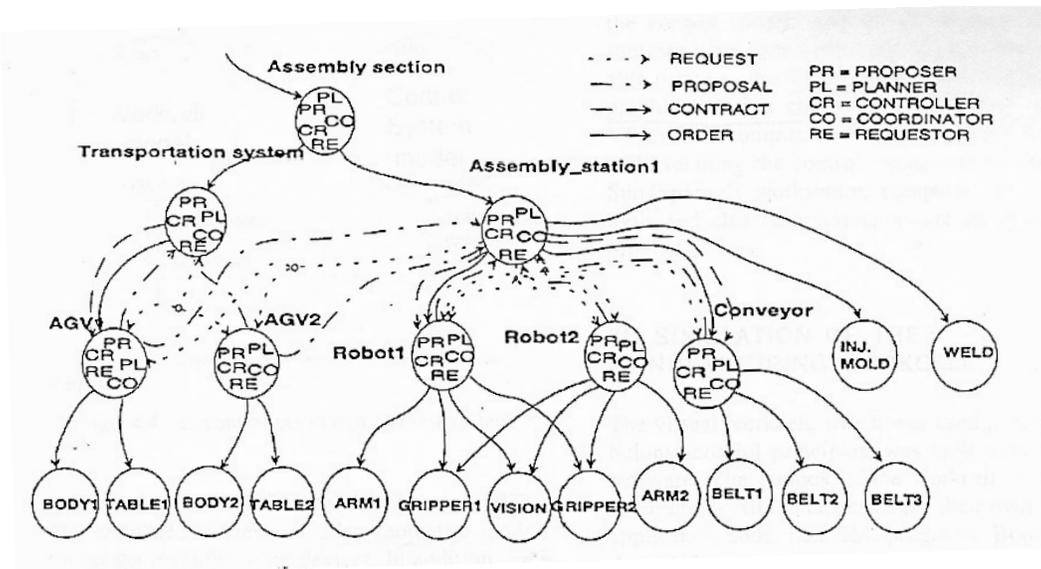


Figure 26: The Flat VTT Robot Cell Hierarchy [114]

Control [101]. In this paradigm, the manufacturing control centers around the evolution of a part from an initial to a final state through the use of services like translation and transfor-

mation. These services are then decomposed functionally into more basic services. In engine line, for example, the services are clamping, assembly, and transportation. These services are then associated statically or dynamically to their respective machine resources. Each instance of parts and resources is designated as a holon with message handling, self-diagnosis, and monitoring capabilities [101]. Additionally, part holons contain a specification of which series are required and how they are to be accomplished [101]. The resultant simulation demonstrated seamless PLC integration into Part-Oriented-Control, self-diagnosis capabilities and moderate messaging traffic [101]. As a result a (proprietary) prescriptive requirements specification was authored [101]. This detailed design is congruent with PROSA and HCBA and perhaps some of these prescriptive steps can be adapted to more sophisticated PROSA/HCBA designs.

3.1.2.4 Miscellaneous Designs A number of other virtual holonic manufacturing systems have been designed and implemented purely in software [179], [150], [75], [92], [93], [195], [226], [103], [161], [106],[107]. Many of these are mentioned later in Section 3.4 as their primary purpose was to illustrate innovative planning and scheduling algorithms.

3.1.3 Implementations

The architectures in the last section were only simulated. The following architectures have all been implemented on physical hardware. Many of them build off the conceptual description in Section 3.1.1 while others are entirely new like the simulated architectures in Section 3.1.2.

3.1.3.1 Daimler Chrysler Holomobiles :

In 2001, Daimler Chrysler fully implemented a physical holonic manufacturing system at its Mercedes Benz V6/V8 plant in Stuttgart, Germany [47]. The defining feature of this systems was the use of a number of holons to improve the performance of an existing conventionally controlled engine assembly line shown in Figure 27 [43]. Extensive analysis by simulation of the line showed that it was overly sensitive to machine and logical disturbances that caused deadlock and starvation [47]. In the event of elevated demand, it was not scalable to higher production level in the event of elevated demand [43]. Hence, the objective of the holonic system was to improve robustness and increase capacity in an incremental fashion [47]. As this was an industrial implementation, it was crucial that the implementation of the added holonic components minimize risk of investment and have a short time ramp-up to optimal performance [47]. Figure 28 shows the added holonic components. They were

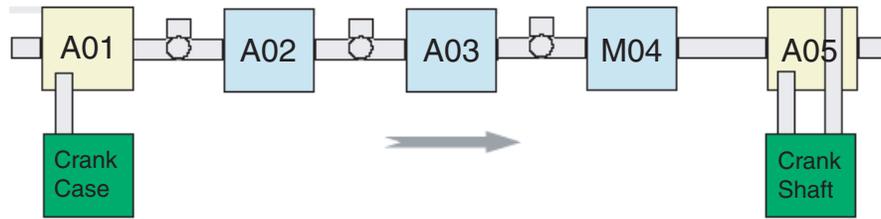


Figure 27: A Conventionally Controlled Mercedes Benz V6/V8 Engine Assembly Line [47]

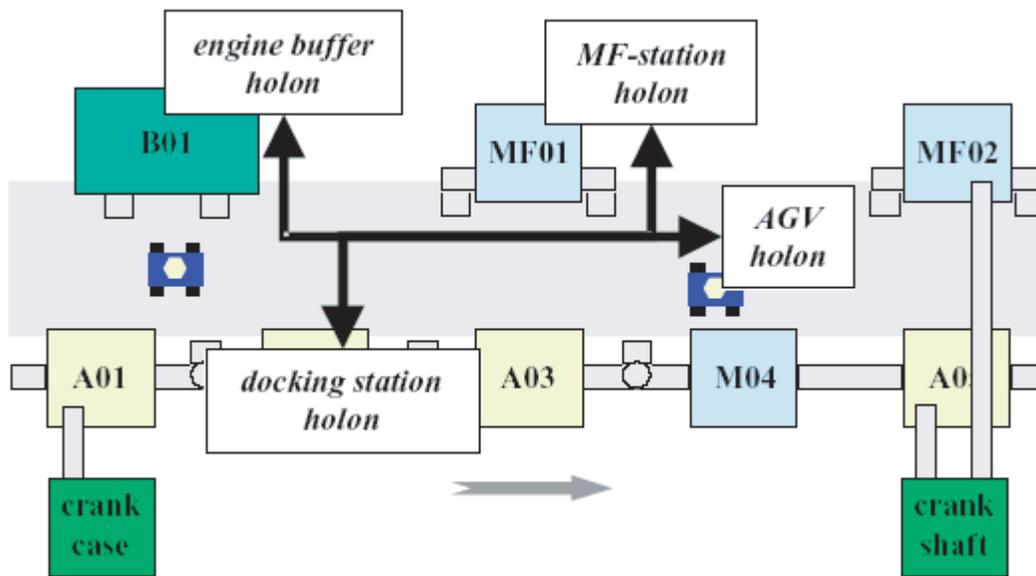


Figure 28: A Conventionally Controlled Line with Implemented Holons [47]

implemented in a three step low risk migration path: 1.) flexible buffers are introduced to alleviate the effect of disturbances 2.) multi-function stations are added to give specific stations more operational capacity. 3.) Multi-function stations are converted to dedicated assembly stations and connected with transportation elements [43]. Each of piece of newly implemented hardware had its own appropriate software agent. The holonic components could be shut of at any time to leave a fully functional conventionally controlled line [47]. Analysis of the new system performance was done in simulation. The addition of holonic components resulted in 37% improved robustness and 61% improved throughput [47].

This holonic system clearly demonstrated the advantages of not just flexible equipment but also flexible software or agents to manage their decisions [43]. Most interestingly, the adaptability of the holonic components is used to effectively implement a low risk migration

path [47]. This stated, this implementation more closely resembles a number of independent holons interacting with a decision-less conventional assembly line than a holarchy of holons negotiation amongst each other. These interactions may be flexible and temporary relationships but neither do they create a holarchy nor do they negotiate to find cooperative solutions ²⁹.

3.1.3.2 University of Cambridge Robot Assembly Cell :

The HCBA architecture was implemented at the University of Cambridge in a meter box robotic assembly cell [55]. The meter box comes in three color variations and is composed of three parts: A, B, and C, which from the box's main base, frame and transparent plate respectively [57]. The full product is shown in Figure 29 and its diagram is shown immediately below it in Figure 30 [259]. The meter box is assembled by screwing part B on to its base

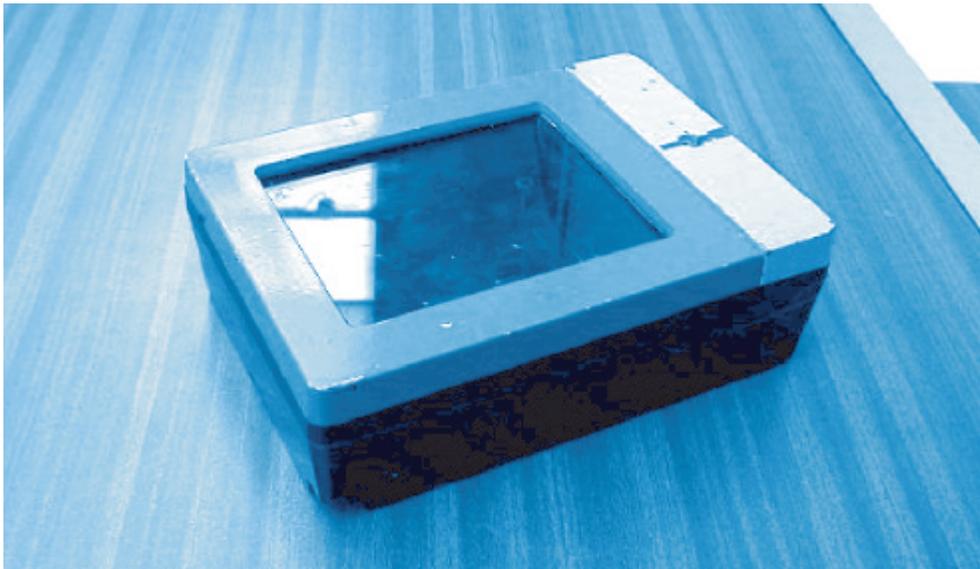


Figure 29: A Holonically Assembled Meter Box [55][259]

part A [55]. Subassembly AB is then flipped so that part C can be screwed to it [57]. This is accomplished with the robotic assembly cell shown in Figure 31 [57] ³⁰. The PUMA robot picks and places the parts, so that the table can rotate them, and the Hirata can screws them together [57]. The flipper robot flips subassembly AB prior to screwing in part C [55].

²⁹Daimler Chrysler is putting greater emphasis in developing manufacturing systems with inherent agility. Another implemented purely agent-based manufacturing system called the Production2000+ can be found at [45][46].

³⁰This same robot assembly cell was used develop another very similar multi-agent system [126].

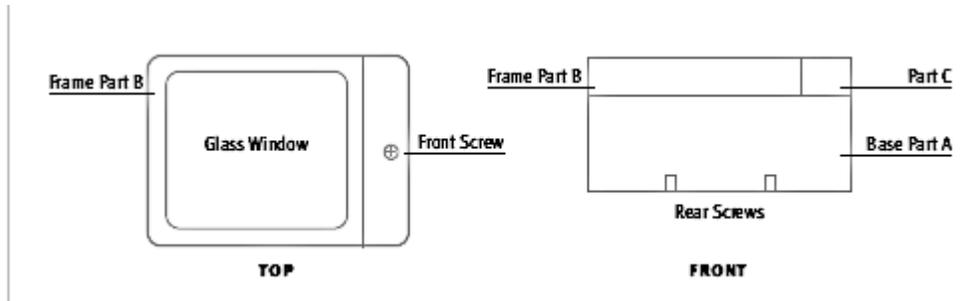


Figure 30: Robotic Assembly Cell Meter Box Diagram [55][259]

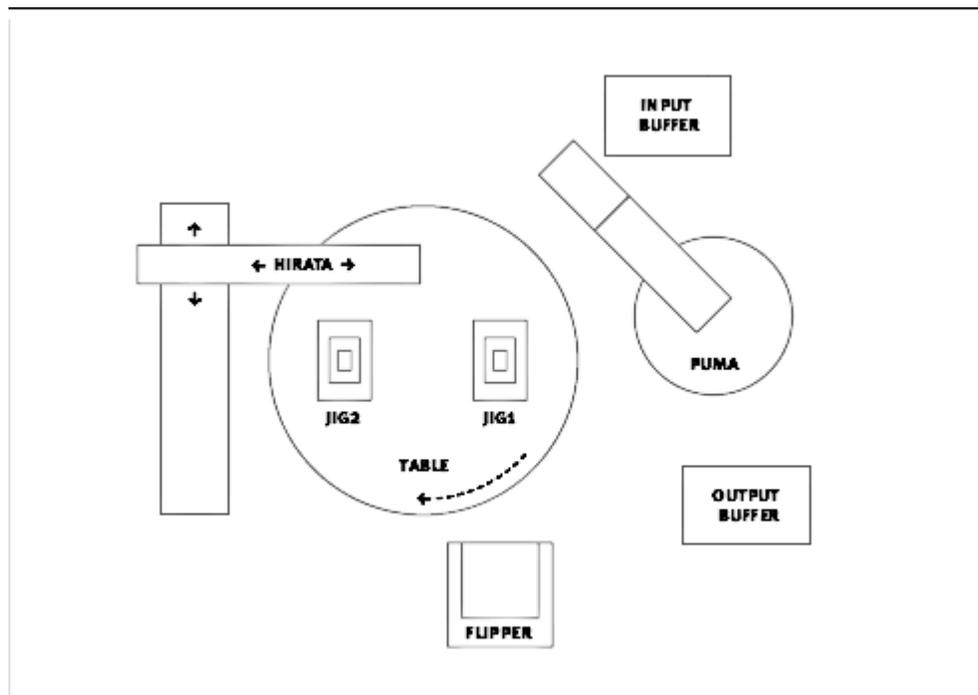


Figure 31: University of Cambridge Holonic Robot Assembly Cell [55][259]

The holonic integration of HCBA is completed in static and dynamic phases which are shown in Figure 32 [55]. In the first, a resource controller, connected to an Omron PLC C200HE, forms the lower level of the holons' software part [60]. It is responsible for the manufacturing execution as well as the house off the BBS [55]. In HCBA, resource holons may not negotiate directly with each other and hence the installation of the resource controller completes static integration [60]. Dynamic integration requires the installation of the production controller which is made up of the software component responsible for the negotiation and message brokering for both product and resource holons [55]. This results in the

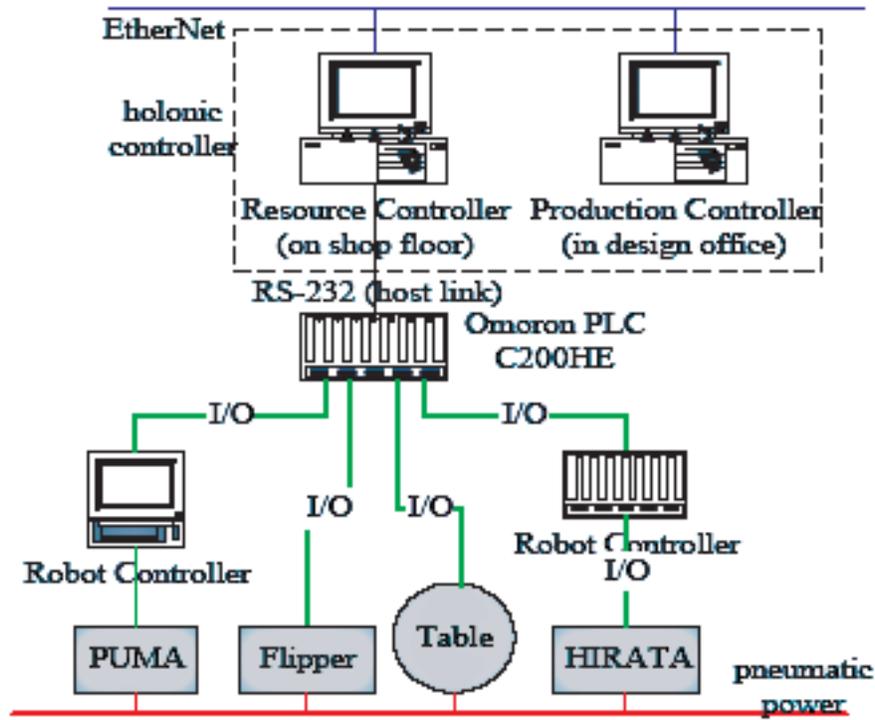


Figure 32: Static and Dynamic HCBA Integration [55]

fully implemented holarchy shown in Figure 33 and corresponds closely to the HCBA system architecture originally shown in Figure 13 on page 29 [60]. The product holon generates a

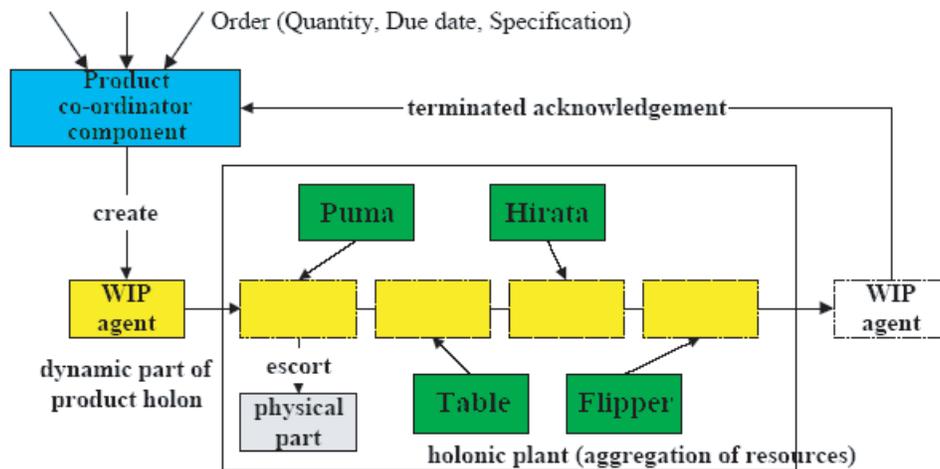


Figure 33: Univeristy of Cambridge Robot Assembly Cell HCBA Architecture [55]

WIP agent that escorts a physical part through the assembly cell. Further detail about this

implementation can be found in [53]. The primary advantages of this implementation is its separated systems integration sand its dynamic communication infrastructure which resulted in a functional holonic system [55].

3.1.3.3 KU Leuven In 1994, KU Leuven implemented one of the first HMS prototypes [240]. In the years that followed, the flexible assembly system was used to demonstrate and complete a voluminous body of research. Among the primary results, was the previously mentioned PROSA architecture [246]. This testbed is diagrammatically shown in Figure 34 [241]. It consists of one PUMA 500 Robot (from Unimation), two Scara 7545 robots (from

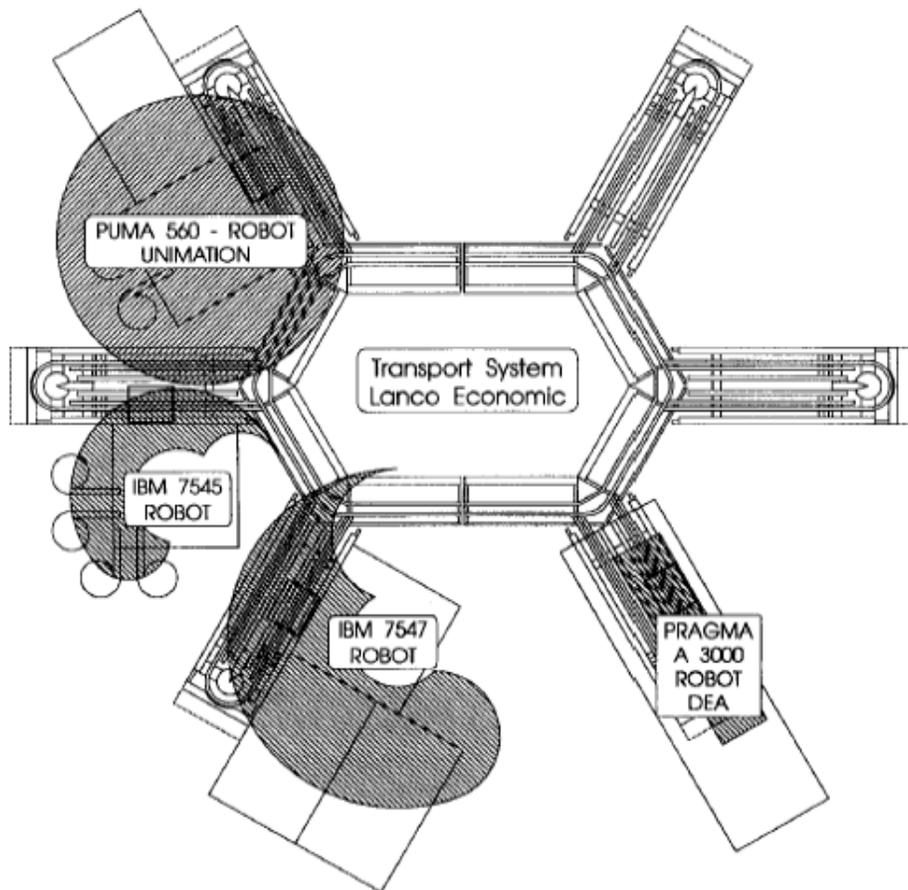


Figure 34: Holonic Flexible Assembly System at KU Leuven [241]

IBM), one Pragma A300 robot (from DEA) and one external transportation loops which together form the lowest level resource holons [241]. The system simultaneously assembles both electric switches and LEGO constructions using a PROSA type architecture [241]. A

centralized heuristics based reactive schedule is implemented with a staff holon and has been investigated throughway in [30], [31], [27], [26], [25]. The flexibly assembly system was also used to benchmark preliminarily holonic systems against hierarchical and heterarchical ones [242]. Many of their results, experimentally supported the arguments provided in Section 2. A number of fundamental research works on the implemented PROSA architecture can be found at [239], [24], [264].

3.1.3.4 Miscellaneous Implementations A number of quasi-holonic manufacturing systems that use embedded multi-agent systems have been reported. Some of these may be found at [6], [7], [51], [50], [98], [128], [147], [148].

3.2 Methodologies & Strategies

The methodologies section recognizes that a logical and step-wise framework is required for design, implementation and migration of holonic manufacturing systems. Many of the architectures presented in the previous section either were too conceptual to be very useful in a physical implementation or did not specify a detailed method by which their implementation could be repeated (perhaps with variations and/or in a different setting). Unfortunately, no fully comprehensive and applicable methodology has been written. This section, however captures some of the most useful tools for design and implementation of HMS. They are categorized as modelling tools, and design and migration methodologies and strategies.

3.2.1 Design & Modelling Tools

Holonic Manufacturing Systems research and development draws expertise from a wide variety of fields. Each of these brings its respective design and modelling tools to HMS development. This section highlights IDEF0 and Petri-Nets from industrial engineering as well as UML from object-oriented software engineering.

3.2.1.1 IDEF0 :

IDEF0 or Integrated Computer Aided Manufacturing Definition is probably the most widely used method for modelling manufacturing system functionality [78]. It is composed of basic building blocks that illustrate a manufacturing activity's inputs, controls, mechanisms, and outputs. A basic building block in addition to an illustrative example is shown in Figure 35 [108]. It is important to note that the links between IDEF0 blocks do not imply a flow of

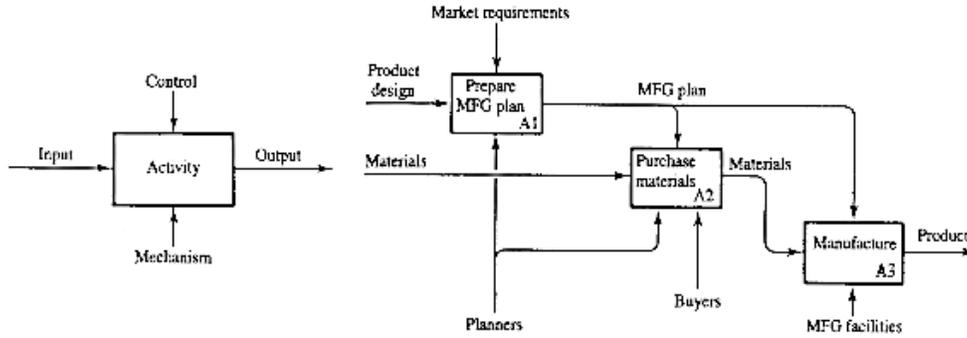


Figure 35: An Illustrative IDEF0 Example

information or material but rather imply a simple dependence [43]. Bussmann notes that IDEF0 networks have been used (with Petri-Nets) to create PLC programs [43]. IDEF0 modelling has been used in HMS development most notably in the HoMuCS architecture's functional models [265],[137]. Bussmann has also noted that it may be used as the input to the DACS multi-agent system design methodology [43].

3.2.1.2 Petri-Nets :

Petri-Nets is another widely used modelling technique. It is composed of places that represent conditions [199]. They are filled by tokens that denote the condition is true [199]. Finally, when the conditions become true they pass through a transition which leads to other places or states that may in and of themselves be conditions for further transitions [199]. An illustrative example shown in Figure 36 describe that a part, machine, robot need to be available for the robot to load the machine with part [199]. Petri-nets are particularly useful because they have a large body of theory that can be used to validate network properties [43]. Like IDEF0, Bussmann states that Petri-Nets can be used as the input to the DACS MAS design methodology [43]. Also, they have been applied at the KU Leuven and University of Cambridge laboratories [247], [59].

3.2.1.3 The Unified Modelling Language – UML :

Object-oriented programming has served an integral part in the development of holonic manufacturing systems partially because they share many of their principles in common [159]. For that reason, UML may be a useful HMS modelling language as it is an effective and simply way of capturing the salient features of an object-oriented software design [105]. While the language is fairly comprehensive of object-oriented language functionality, only

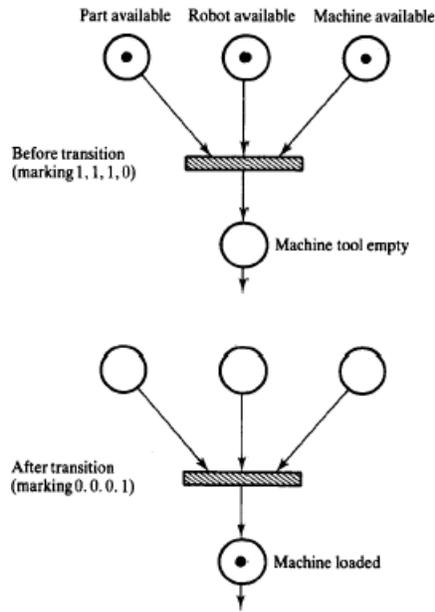


Figure 36: An Illustrative Petri-Nets Example [199]

an illustrative class diagram used in PROSA is shown in Figure 37 [246]. A number of tutorials are available on the internet [105]. Classes such as the holonic manufacturing

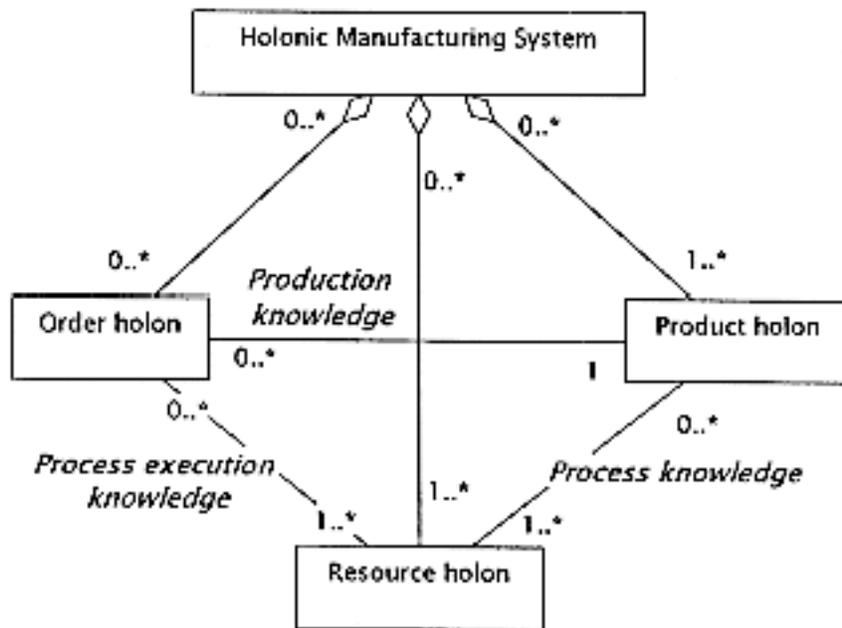


Figure 37: An Illustrative PROSA based UML example [246]

system, product, resource and order holons are represented by boxes connected by three types of lines. Lines with diamonds represents that one class is contained within the other. Solid lines as those between product, resource and order holons show that they are associated with each other. Finally, lines with arrows (not shown) represent that one class is a specialization of another. One can imagine, for example a CNC robot holon connected to the resource holon block via an arrowed-line. Finally, the diagram is filled with number that represent the number of classes in that relationship. The notation 1^* means "at least one" [246].

3.2.1.4 Agent Building Tools :

Agent functionality goes well beyond the original intention of object-oriented languages. The functionality for agent implementation still exists, however, the software engineering process is less than straightforward. A number of agent building environments have been developed to facilitate the process. These include the open source JADE [125], [22], JACK by Agent Oriented Software Group [5],[126], and Zeus by BT [102].

3.2.2 Design Methodologies & Strategies

The review of HMS design methodologies centers around the following: "There are major issues to be resolved including how to define holons for a given problem, what should be an appropriate holonic architecture, how to design effective cooperation mechanisms for good system performance, and implementation issues such as plug-and-play capability for fast deployment". [104] Numerous partial attempts at an HMS design methodology have been made. Even Axiomatic approaches have been investigated [180]. However, all fall short in providing a prescriptive and easily usable comprehensive design methodology. This section, reviews some of these partial methodologies and strategies including those that may come from multi-agents systems research.

3.2.2.1 PROSA :

The KU Leuven research group in addition to the PROSA architecture provided a loose four step guideline for designing a holonic system [248]. In the first step, holons are identified. In the second, the system is designed in detail and implemented. Special care is taken to reuse system components wherever possible and to proceed in a bottom up approach according to the guidelines found in [248]. The third step provides for the installation and configuration of the HMS and finally the HMS is operate in the last step [32]. While these steps might seem intuitive, only the first is describe within the details of the PROSA architecture [248].

The identification of holons, however, is completed in a logical step-wise fashion using UML as seen previously in Figure 37 [246].

3.2.2.2 HoMuCS Methodology :

The HoMuCS methodology uses the HoMuCS architecture to create physically implemented architectures as seen in Figure 38 [138]. The methodology begins with an analysis phase

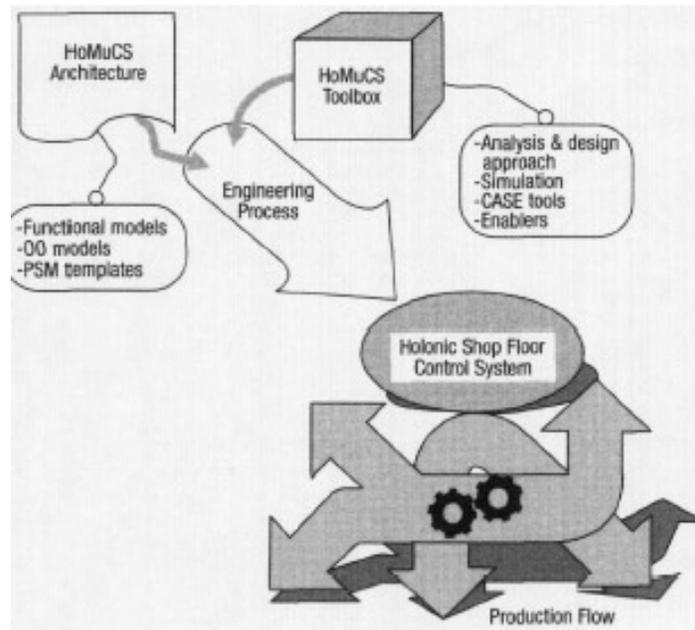


Figure 38: An Overview of the HoMuCS Methodology [138]

which relies on the functional models to guide the basic holon design [137]. Additionally, the designer investigates the details of the functionality as dictated by his hardware specific requirements [138]. This process is basically one of extending the detail of the already existing functional models. In its original version, this process is completed using IDEF0 [137]. The detailed functional models dictate the detailed design of the object-oriented models [138]. These models can be describe in UML as the HoMuCS methodology suggests [137]. Once again this is a process of instantiating and specializing the general holon classes provided in the HoMuCS architecture [138]. The product state models are then filled with product specific information [137]. Finally, these three models are then tied together in an integrated implementation. The HoMuCS toolbox facilitates this process with simulation and analysis tools [138]. This methodology is explained in full detail in [137] and is applied to a number of industrial case studies [206], [225].

3.2.2.3 DACS Multi-Agent System Design Methodology :

The recently published Designing Agent Based Control Systems (DACS) design methodology for production control takes a very different approach from the modelling tools and methodologies that came before it. Its primary advantages are its great degree of usability by an industrial control engineer, its reuse of existing protocols, and its focus on a prescriptive approach based upon decision oriented models [43]. That said, it is a purely a MAS methodology, and any HMS features from its application is purely the result of the transferability of MAS concepts to HMS³¹. The methodology follows three ordered and well-defined phases which begin after the specification of the plant's mechanical components, arrangement and behaviour. In the first phase, the system's control decisions are analyzed [43]. This is composed of identification of the systems' decisions and then determining the dependencies between them [43]. In the second phase, the agents are identified [43]. This follows an iterative process: clustering the decision tasks into agents, and improving the decision model, if necessary [43]. Finally, the agent interaction protocols are chosen [43]. This consists of classifying the dependencies between agents, matching them to protocols in an existing protocol library, and customizing them for their use if necessary [43]. This methodology is explained in full detail and evaluated in [86].

3.2.3 Migration Methodologies & Strategies

The issue of migration from a conventional to a holonic system is critical [101]. There are comparatively few factories (green-field sites) being built across the globe and hence if holonic manufacturing systems are to be widely adopted industrially, it must show a clear low-risk path to quickly upgrade a conventional system to a holonic one [48]. McFarlane describes a general migration framework in Figure 39 [172]. Figure 39a shows a conventional CIM hierarchy. Figure 39b depicts many of the HMS developments to date. They give distributed holonic solutions for a given level most likely without addressing other levels. The methods and strategies described in the following sections describe the later states in the framework. First, the planning, scheduling and execution layers and the control and physical plant layers are integrated separately into the software and physical parts of a holon. Eventually, the two parts are merged to form a full holonic solution [57].

³¹It is the opinion of the author that Holonic Manufacturing System represent a specialized class of Multi-Agent Systems. Hence, a methodology that is sufficiently general for a Multi-Agent Systems will not be sufficiently prescriptive to describe a Holonic Manufacturing System.

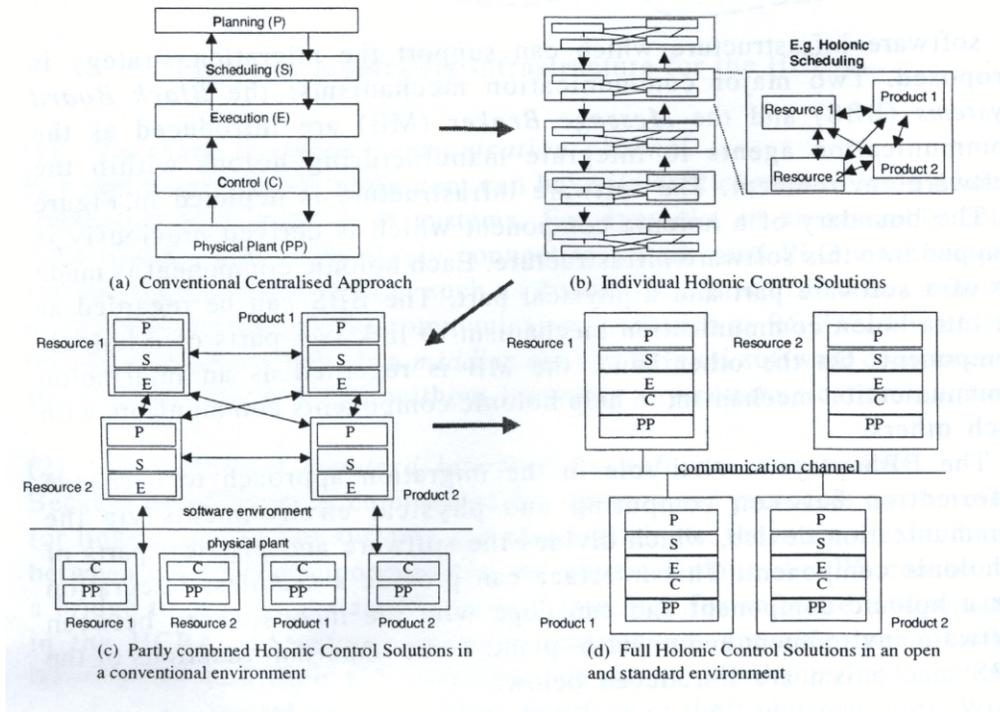


Figure 39: A Migration Framework for a Holonic Factory [57]

3.2.3.1 HCBA Migration Strategy :

The HCBA architectures was proposed with an inherent strategy to migrate from Figure 39b to the full holonic solution in 39 [54]. Its two-phase integration process is essentially a bottom-up migration strategy [57]. Holon formation starts at the machine level and crystallizes upwards to incorporate the standard CIM levels one by one [54]. At the incorporation of the planning and scheduling levels, the CIM realization of planning and scheduling are abandoned in favor of the product controller which provides the same functionality for each holon [55]. These steps are described in greater detail in [53].

3.2.3.2 Daimler-Chrysler Holomobiles Migration Strategy :

The Holomobiles demonstration mentioned previously in Section 3.1.3.1 used a three phase implementation method [47]. Its migration path is inherently different from HCBA however because it assumes that the plant capacity needs to be augmented [47]. The augmented capacity can be implemented using HMS "plug-n-play" nature without detrimentally affecting the underlying conventional assembly line [43]. This migration, however, is not prescriptive enough to describe how it may be repeated. The implementation is also in a sense a hybrid

system because it has not implemented inter-holon negotiation. Hence, the migration path is not complete because it does not describe the full construction.

3.2.3.3 Holden’s Engine Operations Migration Strategy :

Holden’s Engine Operations HMS simulation also provided a useful three-phase migration strategy [101]. In phase one, the services in the previously mentioned Part-Oriented-Control is decomposed [101]. Its feasibility is then determined in simulation by investigating 1.) the holon structure and boundaries 2.) the intra/inter holon messaging requirements, and 3.) the degree to which existing controllers can incorporate a Part-Oriented-Control/holonic framework [101]. In phase 2, the newly defined interfaces are implemented but no new functionality is added [101]. Finally, in phase 3, diagnostic, reconfiguration, and learning capabilities are inserted within the previously defined holon boundaries [101]. This seems like a feasible migration strategy that might be an important part of a more comprehensive methodology.

3.3 Protocols

Thus far, the developments section has given rather broad system wide views of holonic manufacturing systems. Section 3.1 identified the building blocks that form holonic manufacturing systems’ proposed architectures while Section 3.2 attempted to identify the methodologies by which to develop these architectures. Attention now takes on a finer scope and turns to the mechanisms that give the holons their desired behaviour. These are comprised of the protocols by which holons communicate amongst each other, the algorithms from which inter-holon behaviours emerge and the interactions by which holons drive their respective devices.

When speaking of protocols, one must differentiate between them and algorithms [114]. Protocols simply specify the communication method which maps state history to actions [203]. Strategies or algorithms are a particular method of using the protocol to achieve a particular result [203]. In this section, two particularly useful protocols are used.

3.3.1 Contract-Net Protocol

The Contract-Net Protocol, originally proposed in 1980 is probably the most versatile and best-proliferated protocol in holonic manufacturing systems and multi-agent systems. It is a simple negotiation protocol that involves a manager and contractor [224]. The manager has

a task that requires completion [224]. He announces it to a number of contractors which bid on the task within a specified time frame [224]. The manager evaluates the bids, chooses the best one and awards the task to the contractor with the best bid [224]. Bussmann provides a simple graphical representation of the protocol and it is shown in Figure 40 below [43]. There

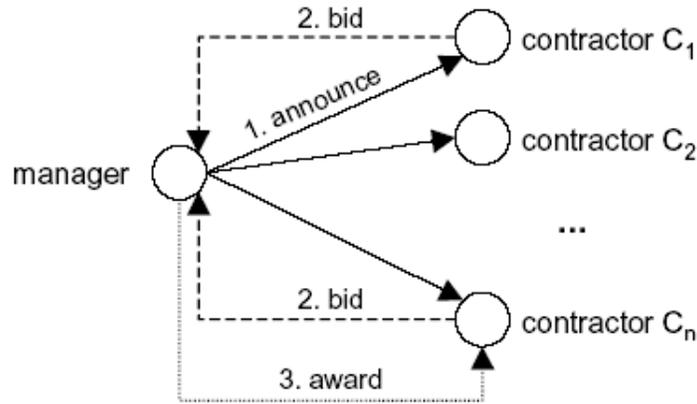


Figure 40: The Contract-Net Protocol [43][224]

also exist numerous versions and extensions of the protocol such as the Extended Contract Net Protocol [89], the Market-Driven Contract Net [10], and the Levelled Commitment Contract Protocol [204]. One may also find applications in many holonic manufacturing systems and multi-agent system algorithms [216] of which many are reviewed in Section 3.4.

3.3.1.1 The Tri-Base Acquaintance Model The Tri-Base Acquaintance (3bA) Model may be viewed as a protocol or method to implement autonomous agents or holons at a minimum of network communication traffic [191]. One method of communication, (exemplified by a vanilla Contract-Net Protocol), is a general broadcast: one holon sends a message to all the other holons in the system [191]. This may be viewed as a completely connected web of nodes and the number of connections grows quadratically. Another method, typified by federation multi-agent systems requires the use of a central facilitator/broker/mediator [191]. This may be viewed as a star-shaped network and the number of connections grows linearly with the number of holons. The final method, the basis for the 3bA model, uses a facilitator-less acquaintance model where each holon maintains a certain level of social knowledge about other holons so as to minimize the number of information requests [191]. This proposed acquaintance model continues to use the Contract-Net Protocol but adds a three based knowledge structure [191]. The cooperation base maintains rather static infor-

mation such as the addresses and predefined responsibilities of the other holons in the system [191]. The task base includes information about what type of tasks a particular holon can complete a task as well as plans on how to carry them out [191]. This information changes a bit more frequently. The state base is in constant flux and maintains the load and state of other holons [191]. The intuitive idea of only requesting information via the network when it is needed forms the basis of the 3bA model. Implementation on a multi-agent system showed significant network traffic reduction [191]. Further detail about the multi-agent system implementation can be found at [153], [155], [154], [156], [190].

3.4 Algorithms

Having mentioned two of the more common protocols, their application in various algorithms can be explored. The algorithms surveyed in this section represent the temporary and flexible relationships in a holarchy's inter-holon communication. From a CIM point of view, these algorithms fulfill the planning and scheduling functionality of CIM's top two layers. Multi-agent planning and scheduling solutions have been proposed in recent years. Some integrate these two activities, while others separate them. For clarity, the distinction between the two activities is defined below.

Definition 3.4.1. Planning – The process of selecting and sequencing activities such that they achieve one or more goals and satisfy a set of domain constraints [216].

Definition 3.4.2. Scheduling – The process of selecting among alternative plans and assigning resources and times to the set of activities in the plan [216].

These algorithms distinguish themselves from conventional scheduling and planning algorithms by the requirements they must fulfill to exist within a multi-agent solution. McFarlane argues that they must have 1.) local (near) real-time reasoning capability, 2.) cooperative mechanisms between modular units, 3.) embedded models for autonomy 4.) diagnosis and error recovery routines and 5.) online learning capabilities [171]. Additionally, Bakers requires 1.) a balanced distribution of computational load among the agents, 2.) a physical correspondence between agents and their physical entities 3.) scalable growth of computational load versus physical entities [11]. The remainder of this section reviews the algorithms that best achieve these objectives. Although, integrated planning-scheduling algorithms better fulfill the criteria above, separated scheduling and planning algorithms are first treated as they can often be integrated later.

3.4.1 Scheduling Algorithms

Scheduling is one of the firm's most important operations activities. It is also one of the most difficult and as a result lead times and works in progress levels are inflated, machines are under-utilized and production completion dates can not be controlled or predicted [150]. Industrially, MRP and MRPII packages are often used but as effectively planning software, they are ill equipped for detailed scheduling [172], [119]. Furthermore, they can not respond easily to on-the-shop-floor events and disturbances [119]. For detailed scheduling, heuristics such as critical ratio (CR) or shortest processing time (SPT) are often used instead [150]³². However, they often yield suboptimal results whose performance is rarely measurable [119]. The crux of the problem is two fold. First, many scheduling problems are excessively computationally intensive; NP-complete in many instances [11]. (This might explain the industrial use of heuristics which at least guarantees a result in feasible time scale [11].) Secondly, even if a good schedule were found in a reasonable time scale, it would become invalid almost immediately as new jobs, rush orders and disturbances emerged [150]. Clearly, the algorithmic requirements previously mentioned would alleviate some of these difficulties. Within the scope of scheduling, the focus turned from finding computationally infeasible optima to finding measurable, consistent and near-optimal schedules within a reasonable time scale. A history of intelligent scheduling may be found in [193], while an excellent review of multi-agent scheduling algorithms can be found in [11]. Some of the most promising of these are reviewed here.

3.4.1.1 Lagrangian Relaxation Scheduling :

Lagrangian relaxation is one such near-optimal advance scheduling algorithm [150]³³. It iteratively minimizes a job-tardiness objective function that can be easily decomposed into smaller problems; each corresponding to a shop floor resource or holon [119]. One author states that up to 90% of the algorithm is distributable[120]. It also gives a lower bound on the objective function which in turn can be used to measure the algorithm's performance [150]. Additionally, its complexity is linear in both the number of machines and the length of the time horizon [119]. Hence, it represents a fully scalable solution. The algorithm is also reconfigurable as previous solutions can be used to seed new iterations that incorporate

³²Many other heuristics such as Earliest Starting Time (EST) and Fewest Operations Eemaining (FOR) exist [117]. Heuristic scheudling methods have also been applied within a holonic context [117],[220],[232]

³³Here the advance scheduling means that a number of tasks must be aggregated and then ordered and positioned in time in advance [11]. New tasks may be added on later iterations of the algorithm [119].

new orders, events and disturbances [120]. The algorithm has been used in numerous real life scenarios, including job-shops, and has shown to provide results within 10% of optimal [150]. One may find numerous examples of its application in [118], [120], [119], [68], [251], [146], [145], [52], [85] as well as later in Section 3.4.2.

3.4.1.2 Temperature Equilibrium Scheduling :

Temperature equilibrium scheduling algorithms are an entirely different type of algorithm. It uses the concepts of relative heat, latent heat and temperature to equilibrate the work load between holons [75]. Each holon has a temperature and relative heat to the other holons [92]. "Hot" holons transfer some of their workload to "cooler" holons subject to a Contract-Net based protocol [75]. The transfer of tasks must also take into account the latent heat which may be defined as the price paid to transfer a task from one holon to another [75]. This is a near real-time (as opposed to an advance) scheduling algorithm [93]. It has been used in simulation to improve throughput, machine utilization and mean delay over more rule-based methods [92], [93].

3.4.1.3 Market-Driven Scheduling :

Market-driven algorithm are probably the most common form of dynamic multi-agent scheduling algorithm. Numerous examples in multi-agent systems and holonic manufacturing systems have been applied in simulation but they are all variations on a theme [3], [11], [10], [12], [14], [13], [76], [121], [212]. All assume that the product order has already been decomposed into component processing tasks (planning)[11]. The principle feature is then to use the Contract-Net Protocol to correlate machines and tasks as new order come online in time [195]³⁴. Saad investigated one possible variation where resource holons contract product holons instead of vice versa [201]. Ramos and Sousa have also investigated market driven algorithms where the product holon is the manager [226]. They use a combined "forward influencing"/"backward influencing" method so that during bid calculation, each resources can evaluate both its availability and the effect of deadlines [227], [221]. This is analogous to the forward/backward scheduling algorithms found in industrial ERP/MES systems [11]. Conflict avoidance is also carefully treated [228]. Many other instances of market-driven scheduling exist and some of those that integrated planning functionality are treated in Section 3.4.2.2.

³⁴The negotiation can also be done with strategies that lie anywhere from purely selfish or purse cooperative [114]. Examples of the can be found in [157] and [114] respectively.

3.4.2 Integrated Planning Scheduling Algorithms

In all cases, scheduling algorithms require as an input the parts' component raw material and/or subassemblies as well as the type of machine/resources required to complete the processes that advance its production. This specification is by definition completed during the planning phase and can be completed in varying degrees of automation. Early in the life of CIM systems, bills of material (BOM) were recognized to greatly facilitate planning activities [108]. Later, CAD/CAM integration made its contributions [199]. However, there still remains a large gap between CAD/CAM outputs and planning specifications that may be used within a near real-time multi-agent system environment. Some authors have reported multi-agent based decomposition techniques [9],[23] while more recently the use of well-specified "recipes" has been proposed to bridge the gap [173]. Additionally, many others have postulated reference architectures that envision this functionality [256]. The remainder of multi-agent planning developments have mostly reported as integrated planning-scheduling algorithms; the most prominent of which are reviewed below. It is useful to note that in most cases the planning functionality of one integrated algorithm can be implemented with different type of scheduling algorithm

3.4.2.1 Lagrangian Relaxation with Planning :

Lagrangian Relaxation scheduling algorithms can acquire planning³⁵ functionality with a distributed two-phase "constraint satisfaction" planning method based upon the contract-net protocol and an adaptive consistency algorithm developed in [70] [103]. In phase one, part-assembly-part triples are generated. These triplets are then combined by satisfying contact constraints using a contract-net protocol to form a "globally consistent combination" or a fully assembled product. Phase two takes this combination and parses it into an assembly sequence and identifies the machine types that can perform each assembly operation. The output of the planning can then be used normally in Lagrangian Relaxation [103]. This combined algorithm has been implemented on a holonic architecture originally described in [110] using a simulated testbed composed of three different assembly operations, one AGV, and products. It was shown to generate efficient plans and schedules [103]. A similar demonstration based upon a Toshiba facility has also been demonstrated [104].

³⁵In this demonstration, planning refers to matching appropriate parts and assembly plans to products [103].

3.4.2.2 Market Based Approaches :

Market based integrated planning/scheduling approaches differ from pure market driven schedules by the addition of some decomposition method which is specified to varying degrees of detail in the literature [131], [249]. In one simulated implementation, order are accepted based upon a management agent's internal model of the expected profit for a given order [157]. The same agent is able to directly translate the order into a sequence of tasks which are then negotiated upon by a machine resources [157]. In another simulated implementation, design agents negotiate with manufacturing agents to fulfill features' specifications and tolerances [161]. Upon determination of feasibility, the design agent provides a list of (machine-type specific) tasks to a "matchmaker" agent which assigns them to machine agents via a contract net protocol [161]. A cost model based upon machining, tool change, and setup times is used to determine the value of the bids [161]. Yet another simulated implementation is done almost identically but with the usage of a "STEP" which includes product details including geometry, dimensions and functionality. A "STEP" parser and interpreter are then used to create a task file to be used in the planning process [106],[107].

3.5 Interactions

The previous section described the algorithm that managed inter-holon communication. This section introduces some of the more promising interactions necessary for intra-holon communication.

3.5.1 IEC 61499 Function Blocks

Implementing real-time functionality in a holonic manufacturing system is difficult because it is rarely straightforward to reconfigure part flows and interactions while maintaining real time constraints [90]. The IEC 61499 function block standard provides one proposed solution [234]. It builds upon the IEC 61331-3 standard for PLC and provides an encapsulated object with execution and algorithmic capabilities ³⁶ Its basic structure is divided into algorithm execution and execution control halves which are shown in Figure 41 [39]. The former functions on a data flow of sensed events similar to other function block models [39]. The latter uses an event flow to trigger the algorithm execution [39]. Brennan captures the functionality in the following sequence and its associated diagram in Figure 42 [39].

³⁶The standard PLC scan cycle creates significant inflexibility in executing scheduled events in a real-time fashion [39].

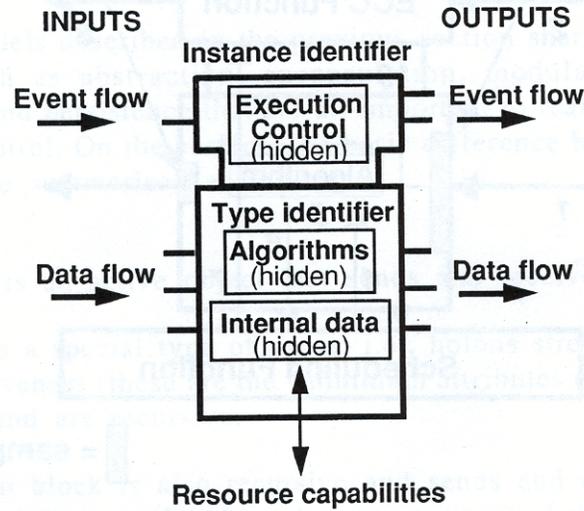


Figure 41: The IEC 61499 Function Block Model [39]

1. Relevant input variables values are made available.
2. The event at the event input occurs.
3. The execution control function notifies the resource scheduling function to schedule an algorithm for execution.
4. Algorithm execution begins.
5. The algorithm completes the establishment of values for the output variables.
6. The resource scheduling function is notified that algorithm execution has ended.
7. The scheduling function invokes the execution control function.
8. The execution control function signal an event output [39].

The advantages of IEC 61499 function blocks are many. Besides integral real-time functionality, their modularity lends themselves to distributed control systems [96]. They may also be aggregated in numerous flexible combinations [96]. Their encapsulation also allows for modular development (hence lower costs) of all holonic systems activities (i.e. sensing, actuation, control, communication) [96]. System integration costs will also be reduced as compatible software will be integrated; potentially with open-standard methodologies [96]. From a migration viewpoint, they provide an open-standard modelling framework which

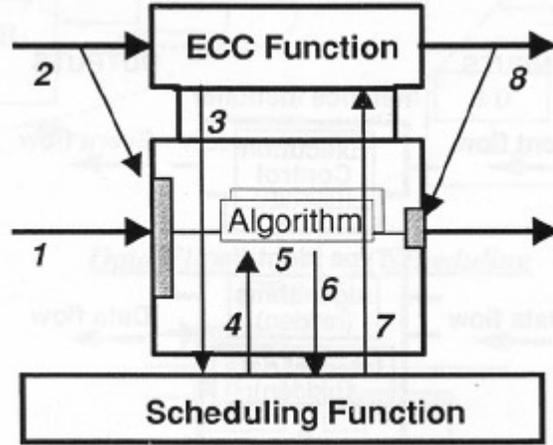


Figure 42: IEC 61499 Function Block Data/Event Synchronization [39]

can be used, for example, to reimplement academic prototypes to proprietary systems [96]. Hence, IEC 61499 function blocks can be viewed as a good way to implement agents, holons, and their respective systems [96]. As the IEC 61499 has developed, it has received a great deal of attention within the MAS and HMS community. There have been many investigations into the use of IEC 61499 Function blocks [37], [253], [254], [252], [255], [272], [271], [275]. In particular, improvements in fault tolerance [94], and reconfigurability have been investigated [273], [274].

3.5.2 Function Block Operating System

The use and implementation of IEC 61499 function blocks is facilitated with the creation of a function block operating system (FBOS) [96]. Fletcher identifies two paths for achieving holonic reconfigurability [96]. In the first, the function blocks are coupled tightly to an object-oriented language [96]. This uses conventional computing models and operating systems to give real-time functionality [96]. For example, one could simulate the function block in JAVA, create a UML model with real-time and agent extensions prior to the eventual function block implementation [96]. The other option is the use of a FBOS to schedule and guarantee the timing of function block execution [96]. A FBOS would also share the manufacturing resources among conceivably numerous function blocks [96]. It can also encapsulate the function blocks to create a virtual machine that simulates the entire holonic manufacturing system [96]. One possible vision of a FBOS based HMS architecture is found in Figure 43. Finally, the FBOS can include a Real Time Guarantee System (RTGS) that directs interfaces

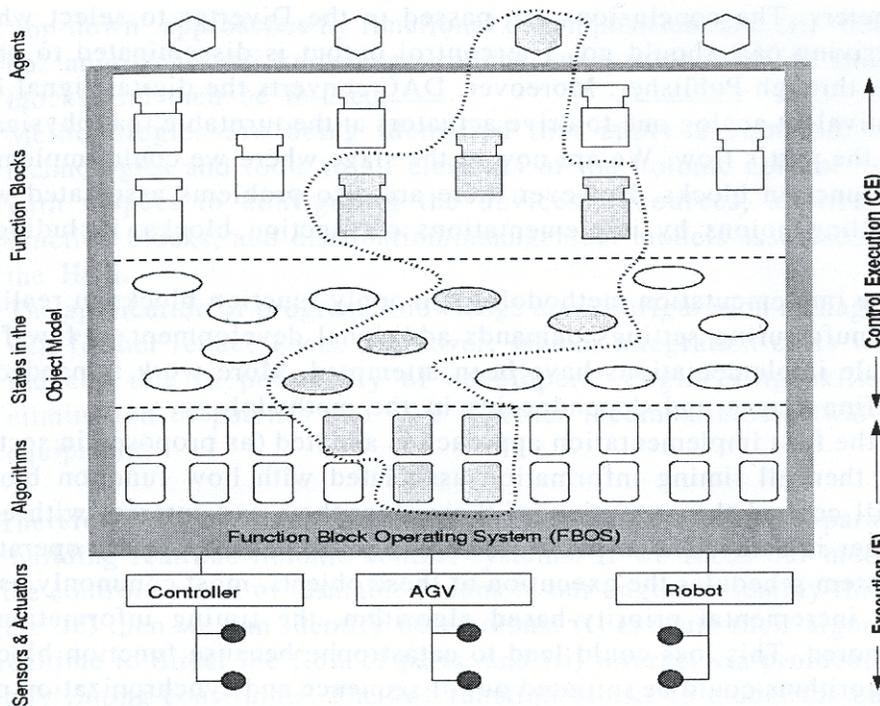


Figure 43: A FBOS based Holonic Manufacturing System

with manufacturing resources in order to enforce execution time periods and deadlines [96].

4 Open Issues & Industrial Adoption

After having reviewed the architectures, methodologies, protocols, algorithms and protocols in the Holonic Manufacturing System field, one can see a number of research gaps remain that prevent full industrial adoption. They can be divided grossly into design and implementation gaps.

4.1 HMS System Design

4.1.1 Availability of Proven Design Methodologies

As seen in the methodologies section, the holonic manufacturing systems field lacks a truly prescriptive, easily implementable design methodology. The DACS methodology as a multi-agent system methodology provides a good reference point but can not be taken in its present

form to design holonic manufacturing systems. Additionally, the DACS methodology, HCBA architecture, and the Daimler-Chrysler Holomobiles demonstration together provide the beginnings of a migration design methodology for an existing CIM architecture. Finally, the Holonic Manufacturing System Consortium will undoubtedly have to investigate in detail the existing CIM design methodologies as a litmus test for its own methodologies. Any enterprise when given the option of a developing manufacturing system will only choose an HMS design methodology if it has been favorably evaluated against a proven CIM methodology.

4.1.2 Analysis of Performance of Holonic Manufacturing Systems

This leads to the second unresolved design point. The architectures above, theoretical or implemented need to be systematically compared and evaluated against the current best alternative. Without such evaluation rigor, industrial partners will perceive the investment risk as too high for full implementation. Specifically, the HMS community has often stated that a holonic system is robust and adaptable to disturbances and failures. These particular advantages should be explicitly demonstrated.

4.2 HMS System Implementation

4.2.1 Appropriate environments for implementing holonic control

As a result of the evaluation and analysis done in the previous section, there should be a set of upon which an HMS system will most successfully function. It is unlikely that the agility of holonic manufacturing systems is required for all industrial sectors. Hence, the evaluation studies should give guidelines as to when best implement holonic systems. This may include an identification of the most suitable industrial sectors, product types, and market behaviors.

4.2.2 Establishing suitable standards for holonic control systems

Finally, there must be a greater degree of standardization of holonic systems. As reviewed, there exists a very large number of holonic architectures among which lies a great deal of disagreement. Greater agreement on holonic architectures, perhaps aided by a solid evaluation process, must be achieved. Ultimately, there may be many different HMS architectures as there are many CIM architectures. However, the designer should have access to a library of standardized HMS components. Certainly, the FIPA and IEC 61499 function blocks standards provide a promising point of development in this regard.

References

- [1] E. Adam, R. Mandiau, and C. Kolski. Homascow: A holonic multi-agent system for cooperative work. In *Workshop W04 on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS2000), in 11th International Workshop on Database and Expert System Applications (DEXA2000)*, pages 247–253, Greenwich, UK, 2000.
- [2] E. Adam, R. Mandiau, and C. Kolski. Application of a holonic multi-agent system for cooperative work to administrative processes. *Journal of Applied Systems Science: Special Issue*, 2(1):175–196, 2001.
- [3] H. Adelsberger and W. Conen. Economic coordination mechanisms for holonic multi-agent systems. In *Workshop W04 on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS2000), in 11th International Workshop on Database and Expert System Applications (DEXA2000)*, volume IEEE, pages 236–241, Greenwich, UK, 2000.
- [4] J. Agre, G. Elsley, D. McFarlane, J. Cheng, and B. Gunn. Holonic control of a water cooling system for a steel rod mill. In *Rensselaer’s Fourth International Conference on Computer Integrated Manufacturing and Automation Technology*, New York, 1994.
- [5] AOS. What is jack? Technical report, The Agent Oriented Software Group, <http://www.agent-software.com/shared/products/index.html>, 2004.
- [6] T. Arai, Y. Aiyama, Y. Maeda, M. Sugi, and J. Ota. Agile assembly system by ‘plug and produce’. *CIRP Annals - Manufacturing Technology*, 49(1):1–4, 2000.
- [7] T. Arai, Y. Aiyama, M. Sugi, and J. Ota. Holonic assembly system with plug and produce. In *Proceedings of the 2nd international workshop on intelligent manufacturing systems*, pages 119–127, Leuven, Belgium, 2001.
- [8] T. Arai, J. Ota, and Y. Aiyama. Assembly by non-grasping manipulation. *CIRP*, 46(1):15–17, 1997.
- [9] C. Arbib and F. Rossi. Optimal resource assignment through negotiation in a multi-agent manufacturing system. *IIE Transactions (Institute of Industrial Engineers)*, 32(10):963–974, 2000.

- [10] A. Baker. *Manufacturing Control with a Market-Driven Contract Net*. PhD thesis, Rensselaer Polytechnic Institute, NY, 1991.
- [11] A. Baker. A survey of factory control algorithms which can be implemented in a multi-agent heterarchy. *Journal of Manufacturing Systems*, 17:297–320, 1998.
- [12] A.D. Baker. Case study results with the market-driven contract net manufacturing computer-control architecture. In *AUTOFACT*, pages 31.17–31.35, Detroit, MI, 1992.
- [13] A.D. Baker. Metaphor or reality: A case study where agents bid with actual costs to schedule a factory. In S.H. Clearwater, editor, *Market-Based Control: A Paradigm for Distributed Resource Allocation*. Addison-Wesley, Reading, MA, 1996.
- [14] A.D. Baker and M.E. Merchant. Automatic factories. *Potentials, IEEE*, 12(4):15–20, 1993.
- [15] S. Balasubramanian and D.H. Norrie. A multi-agent intelligent design system integrating manufacturing and shop-floor control.pdf. In *Proceedings of the first International Conference on Multi-Agent Systems - ICMAS'95*, pages 1–9, San Francisco, 1995, 1995.
- [16] S. Balasubramanian, R. Brennan, and D. Norrie. Requirements for holonic manufacturing systems control. In *Workshop W04 on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS2000)*, in *11th International Workshop on Database and Expert System Applications (DEXA2000)*, pages 214–218, Greenwich, UK, 2000.
- [17] S. Balasubramanian, R. Brennan, and D.H. Norrie. Requirements for real time holonic control. *Journal of Applied Systems Science: Special Issue*, 2(1):44–60, 2001.
- [18] S. Balasubramanian, F. Maturana, and D. Vasko. An autonomous cooperative system for high speed sorting systems. In *Workshop W04 on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS2000)*, in *11th International Workshop on Database and Expert System Applications (DEXA2000)*, pages 254–259, Greenwich, UK, 2000.
- [19] S. Balasubramanian and D. Norrie. Intelligent manufacturing system control. In *Canadian Conference on Electrical and Computer Engineering 2*, pages 570–573, 1996.

- [20] S. Balasubramanian, X. Zhang, and D.H. Norrie. Intelligent control for holonic manufacturing systems. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 214(B10):953–961, 2000.
- [21] J. Barata and L.M. Camarinha-Matos. Implementing a contract-based multi-agent approach for shop floor agility. In *Proceedings. 13th International Workshop on Database and Expert Systems Applications. Holomas 2002*, pages 544–550, 2002.
- [22] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa. Jade a white paper.pdf. Technical report, Telecom Italia Laboratory, Torino, Italy, 2003.
- [23] G. Biswas, B. Sugato, and A. Saad. Holonic planning and scheduling for assembly tasks. Technical report, Centre for Intelligent Systems, Vanderbilt University, Nashville, TN, 1995.
- [24] L. Bongaerts. *Integration of Scheduling and Control in Holonic Manufacturing Systems*. PhD thesis, Department Werktuigkunde Afdeling Productietechnieken. Katholieke Universiteit Leuven, 1998.
- [25] L. Bongaerts, V.H. Brussel, P. Valckenaers, and P. Peeters. Reactive scheduling in holonic manufacturing systems: Architecture, dynamic model and cooperation strategy. *Advanced Summer Institute of the Network of Excellence on Intelligent Control and Integrated Manufacturing Systems*, 1997.
- [26] L. Bongaerts, Van Brussel H., and V. P. Schedule execution using perturbation analysis. In *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 1998.
- [27] L. Bongaerts, Y. Indrayardi, Brussel Van H., and Valckenaers P. Predictability of hierarchical, heterarchical and holonic control. In *Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems*, pages 167–176, Leuven, Belgium, 2001.
- [28] L. Bongaerts, Mon, L. Monostori, D. McFarlane, and B. Kadar. Hierarchy in distributed shop-floor control. In *Proceedings of IMS '98-ESPRIT Workshop in Intelligent Manufacturing Systems*, pages 97–114, Lausanne Switzerland, 1998.
- [29] L. Bongaerts, L. Monostori, D. McFarlane, and K. Botond. Hierarchy in distributed shop floor control. *Computers in Industry*, 43(2):123–137, 2000.

- [30] L. Bongaerts, Valckenaers P., Brussel Van H., and Peeters P. Schedule execution in holonic manufacturing systems. In *29th CIRP International Seminar Manufacturing Systems*, Osaka, 1997.
- [31] L. Bongaerts, P. Valckenaers, H. Van Brussel, and J. Wyns. Schedule execution for a holonic shop floor control system. In *Proceedings of the Advanced Summer Institute, NOE for ICIMS, Esprit*, pages 115–124, Lisboa, 1995.
- [32] L. Bongaerts, J. Wyns, J. Detand, H. Van Brussel, and P. Valckenaers. Identification of manufacturing holons. In *European Workshop on Agent Oriented Systems in Manufacturing*, pages 57–73, Berlin, 1996.
- [33] G. Booch. *Object-Oriented Analysis and Design with Applications. 2nd Edition*. Addison-Wesley, 1994.
- [34] G.M. Booth. *The Distributed System Environment*. McGraw-Hill, New York, 1981.
- [35] R. Brennan, S. Balasubramanian, and Norrie. A dynamic control architecture for metamorphic control of advanced manufacturing systems. In *Proceedings of the International Symposium on Intelligent Systems and Advanced Manufacturing*, pages 213–223, 1997.
- [36] R. Brennan, S. Balasubramanian, and D. Norrie. Dynamic control architecture for advanced manufacturing systems. In *Proceedings of International Conference on Intelligent Systems for Advanced Manufacturing*, Pittsburgh, PA, 1997.
- [37] R. Brennan, M. Fletcher, and D.H. Norrie. Reconfiguring real-time holonic manufacturing systems. In *Second International Workshop on Industrial Applications of Holonic and Multi-Agent Systems (Holomas 2001)*, Prague, 2001.
- [38] R. Brennan and D. Norrie. Holonic concepts: the 'janus effect' in manufacturing. In *8th Holonic Manufacturing Systems International Consortium Meeting*, Vancouver, BC, Canada, 1998.
- [39] R. Brennan and D.H. Norrie. Agents, holons and function blocks: Distributed intelligent control in manufacturing. *Journal of Applied Systems Science: Special Issue*, 2(1):1–19, 2001.

- [40] R. Brooks. A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of [legacy, pre - 1988]*, 2(1):14–23, 1986.
- [41] J. Brusey, M. Fletcher, M. Harrison, A. Thorne, S. Hodges, and D.C. McFarlane. Auto-id based control demonstration: Phase 2: Pick and place packing with holonic control. Technical report, Auto -ID White Paper, 2003.
- [42] S. Bussmann. An agent-oriented architecture for holonic manufacturing control. In *First Open Workshop, Proceedings of IMS '98-ESPRIT Workshop in Intelligent Manufacturing Systems*, Lausanne, CH, 1998.
- [43] S. Bussmann. *An Agent-Oriented Design Methodology for Production Control*. PhD thesis, University of Southampton, 2003.
- [44] S. Bussmann and D. McFarlane. Rationales for holonic manufacturing control. In *Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems*, pages 177–184, Leuven, Belgium, 1999.
- [45] S. Bussmann and K. Schild. Self-organizing manufacturing control: An industrial application of agent technology. In *Proceedings of the Fourth International Conference on Multi-Agent Systems*, pages 87–94, Boston, MA, 2000.
- [46] S. Bussmann and K. Schild. An agent-based approach to the control of flexible production systems. In *IEEE Symposium on Emerging Technologies and Factory Automation, ETFA*, volume 2, pages 481–488, Antibes-Juan les pins, 2001.
- [47] S. Bussmann and J. Sieverding. Holonic control of an engine assembly plant an industrial evaluation. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 5, pages 3151–3156, Tucson, AZ, 2001.
- [48] S. Bussmann and J. Sieverding. Holonic control of and engine assembly plant and industrial evaluation. In *IEEE Systems, Man, and Cybernetics Conference*, volume 1, pages 169–174, Berlin, Germany, 2001.
- [49] C.G. Cassandras. Autonomous material handling in computer integrated manufacturing. In A. Kusiak, editor, *Modelling and Design of Flexible Manufacturing Systems*. Elsevier Science Publishers, Amsterdam, 1986.

- [50] S. Cavalieri, L. Bongaerts, M. Macchi, and J. Wyns. A benchmark framework for manufacturing control. In *Proceedings of the Second International Workshop on Intelligent Manufacturing Systems*, pages 225–236, Leuven, Belgium, 1999.
- [51] S. Cavalieri, M. Taisch, and M. Garetti. An experimental benchmarking of two multi-agent systems for production scheduling and control. Technical report, Lausanne, 1998.
- [52] H. Chen and P. Luh. Scheduling and coordination in manufacturing enterprise automation. *Proceedings - IEEE International Conference on Robotics and Automation*, 1:389–394, 2000.
- [53] J.-L. Chirn. *Developing a Reconfigurable Manufacturing Control System – A Holonic Component Based Approach*. PhD thesis, University of Cambridge, 2001.
- [54] J.-L. Chirn and D. McFarlane. A migration of holonic approach to production control systems. In *1st IFAC Workshop on MAS'99*, Vienna, Austria, 1999.
- [55] J.-L. Chirn and D. McFarlane. A component-based approach to the holonic control of a robot assembly cell. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 4, pages 3701–3706, 2000.
- [56] J.-L. Chirn and D. McFarlane. A holonic component-based approach to reconfigurable manufacturing control architecture. In *Workshop W04 on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS2000), in 11th International Workshop on Database and Expert System Applications (DEXA2000)*, London, UK, 2000.
- [57] J.-L. Chirn and D. McFarlane. Building holonic systems in today's factories: A migration strategy. *Journal of Applied Systems Science: Special Issue*, 2(1):82–105, 2001.
- [58] J.-L. Chirn and D.C. McFarlane. A holonic component-based architecture for manufacturing control systems. In *MAS '99*, pages 219–223, Vienna, Austria, 1999.
- [59] J.-L. Chirn and D.C. McFarlane. Petri-nets based design of ladder logic diagram. Technical report, University of Cambridge Institute for Manufacturing, Cambridge UK, 1999.
- [60] N.N. Chokshi and D.C. McFarlane. Rationales for holonic manufacturing systems in chemical process industries. In *Second International Workshop on Industrial Applications of Holonic and Multi-Agent Systems (Holomas 2001)*, Prague, 2001.

- [61] J. Christensen. Holonic manufacturing systems - initial architecture and standards directions. In *First European Conference on Holonic Manufacturing Systems*, Hannover, Germany, 1994.
- [62] J. Christensen. Basic principles of hms architecture. In *HMS Symposium 2000, Japan*, 2000.
- [63] D. Cockburn and N.R. Jennings. Archon: A dai system for industrial applications. In G. O' Hare and N.R. Jennings, editors, *Foundations of Distributed artificial Intelligence*, pages 319–344. John Wiley and Sons, 1996.
- [64] HMS Consortium. Holonic manufacturing systems overview. Technical report, <http://hms.ifw.uni-hannover.de/public/Intro/overview.htm>, 2000.
- [65] S. Cranefield and M. Purvis. An agent-based architecture for software tool coordination. In *Intelligent Agent Systems: Theoretical and Practical Issues. Lecture Notes in Artificial Intelligence*, pages 44–58. Springer Verlag, 1997.
- [66] M. Cutkosky, R. Engelmor, R. Fikes, T. Gruber, M. Genesereth, W. Mark, J. Tenenbaum, and J. Weber. Pact: An experiment in integrating concurrent engineering systems. *IEEE Computer*, 26(1):28–37, 1993.
- [67] M.R. Cutosky, P.S. Fussell, and R.Jr. Milligan. The design of a flexible machining cell for small batch production. *Journal of Manufacturing Systems*, 3(1):39–58, 1984.
- [68] C.S. Czerwinski and P.B. Luh. Scheduling products with bills of materials using an improved lagrangian relaxation technique. *IEEE Transactions on Robotics and Automation*, 10(2):99–111, 1994.
- [69] T.P. Darr and W.P. Birmingham. Attribute-space representation and algorithm for concurrent engineering. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 10(1):21–36, 1996.
- [70] R. Dechter and J. Pearl. Network-based heuristics for constraint-satisfaction problems*1. *Artificial Intelligence*, 34(1):1–38, 1987.
- [71] S.M. Deen. Co-operation issues in holonic manufacturing systems. In H.G.J. Yoshikawa, editor, *Information Infrastructure Systems for Manufacturing*. Elsevier Science B V, 1993.

- [72] S.M. Deen. Cooperation issues in holonic manufacturing systems. In *IFIP Transactions B: Computer Applications in Technology*, volume B-14, pages 401–412, 1993.
- [73] S.M. Deen. A co-operation for holonic interactions in manufacturing. In *Proceedings of the Second International Working conference on Cooperating Knowledge Based Systems*, pages 103–124, 1994.
- [74] S.M. Deen. Cooperation framework for holonic interactions in manufacturing. In *Proceedings of the Second International Working Conference on CKBS*, pages 103–124, 1994.
- [75] S.M. Deen and M. Fletcher. Temperature equilibrium in multi-agent manufacturing systems. In *10th International Conference on Database and Expert Systems Applications (DEXA '00)*, pages 259–264, London, UK, 2000.
- [76] P. Dewan and S. Joshi. Distributed scheduling of job shop using auctions. In *Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems*, Leuven, Belgium, 1999.
- [77] D.M. Dilts, N.P. Boyd, and H.H. Whorms. Evolution of control architectures for automated manufacturing systems,. *Journal of Manufacturing Systems*, 10(1):79–93, 1991.
- [78] G. Doumeingts, B. Vallespir, and D. Chen. Methodologies for designing cim systems: A survey. *Computers in Industry*, 25:263–280, 1995.
- [79] N. Duffie. An approach to the design of distributed machinery control systems. *IEEE Transactions on Industry Applications*, 1A-18(4):435–442, 1982.
- [80] N. Duffie, R. Chitturi, and J. Mou. Fault tolerant heterarchical control of heterogeneous manufacturing entities. *Journal of Manufacturing Systems*, 7(4):315–328, 1988.
- [81] N. Duffie and R. Piper. Non hierarchical control of a flexible manufacturing cell. *Robotics and Computer Integrated Manufacturing*, 3(2):175–179, 1987.
- [82] N. Duffie and V. Prabdu. Real time distributed scheduling of heterarchical manufacturing systems. *Journal of Manufacturing Systems*, 13(2):94–107, 1994.
- [83] E.H. Durfee, V.R. Lesser, and D.D. Corkill. Trends in cooperative distributed problem solving. *IEEE Transactions on Knowledge and Data Engineering*, 1(1):63–83, 1989.

- [84] R.E. Fairley. *Software Engineering Concepts*. McGraw-Hill Company, New York, 1985.
- [85] L. Fang, P.B. Luh, and H. Chen. Improving the lagrangian relaxation approach for large job-shop scheduling. In *Proceedings of the IEEE Conference on Decision and Control*, volume 1, pages 552–557, Sydney, NSW, 2000.
- [86] A. Farid. An evaluation of the dacs methodology. Technical report, University of Cambridge Institute for Manufacturing, Cambridge UK, 2004.
- [87] FIPA. Fip agent management specification. Technical report, <http://www.fipa.org>, 2003.
- [88] K. Fischer. Agent based design of holonic manufacturing systems. *Robotics and autonomous Systems*, 27(1-2):3–13, 1999.
- [89] K. Fischer, J.P. Muller, M. Pischel, and D. Schier. A model for cooperative transportation scheduling. In *Proceedings of First International Conference on Multi-Agent Systems (ICMAS)*, pages 109–116, San Francisco, 1995.
- [90] M. Fletcher, R. Brennan, and D. Norrie. Modelling and reconfiguring intelligent holonic and manufacturing systems with internet-based mobile agents. *Journal of Intelligent Manufacturing*, 14(1):7–23, 2003.
- [91] M. Fletcher, R. Brennan, D. Norrie, and M. Fleetwood. Reconfiguring processes in a holonic sawmill. In *Proceedings of the 2001 IEEE Systems, Man and Cybernetics Conference*, 2001.
- [92] M. Fletcher and M. Deen. Task rescheduling in multi agent manufacturing. In *9th International Conference on Database and Expert Systems Applications (DEXA '99)*, Florence, Italy, 1999.
- [93] M. Fletcher, M. Deen, and P. Granby. An evaluation of rescheduling techniques and architectures in holonic manufacturing systems. *Journal of Intelligent Information Systems*, 2000.
- [94] M. Fletcher and S.M. Deen. Fault-tolerant holonic manufacturing systems. *Concurrency and Computation: Practice and Experience*, 13:43–70, 2001.
- [95] M. Fletcher, E. Garcia-Herreros, J. Christensen, M. Deen, and R. Mittmann. An open architecture for holonic cooperation and autonomy. In *Workshop W04 on Industrial*

- Applications of Holonic and Multi-Agent Systems (HoloMAS2000)*, in *11th International Workshop on Database and Expert System Applications (DEXA2000)*, volume TR/DAKE-004, pages 224–231, London, UK, 2000.
- [96] M. Fletcher, D.H. Norrie, and J. Christensen. A foundation for realtime holonic control systems. *Journal of Applied Systems Science: Special Issue*, 2(1):20–43, 2001.
- [97] M. Fox, J. Chionglo, and M. Barbuceanu. The integrated supply chain management system. Technical report, Department of Industrial Engineering. University of Toronto, 1993.
- [98] N. Fujita. Japanese holonic testbed project. In *HMS Symposium 2000, Japan*, 2000.
- [99] B.R. Gaines, D. Norrie, and A.Z. Lapsley. Mediator: an intelligent information system supporting the virtual manufacturing enterprise. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 964–969, 1995.
- [100] E. Garcia-Herreros, J. Christensen, J.M. Prado, and S. Tamura. Ims - holonic manufacturing systems: System components of autonomous modules and their distributed control. Technical report, HMS Consortium, 1994.
- [101] N. Gayed, D. Jarvis, and J. Jarvis. A strategy for the migration of existing manufacturing systems to holonic systems. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 319–324, San Diego, 1998.
- [102] K. Glanzer, A. Hammerle, and R. Geurts. The appliance of zeus agents in manufacturing systems. In *Second International Workshop on Industrial Applications of Holonic and Multi-Agent Systems (Holomas 2001)*, Prague, 2001.
- [103] L. Gou, T. Hasegawa, P. Luh, S. Tamura, and J. Oblak. Holonic planning and scheduling for a robotic assembly testbed. In *Proceedings of the 4th Rensselaer International Conference on Computer Integrated Manufacturing and Automation Technology.*, Rensselaer, NY, 1994.
- [104] L. Gou, P. Luh, and Y. Kyoya. Holonic manufacturing scheduling: architecture, cooperation, mechanism, and implementation. *Computers in Industry*, 37(3):213–231, 1998.

- [105] I. Graham and A. Wills. Uml tutorial version 1.4. Technical report, Trireme International Ltd., <http://uml.tutorials.trireme.com/>, 2004.
- [106] P. Gu and S. Balasubramanian. Bidding-based autonomous process planning and scheduling. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 2620, pages 184–194, 1995.
- [107] P. Gu, S. Balasubramanian, and D. Norrie. Bidding-based process planning and scheduling in a multi-agent system. *Computer and Industrial Engineering*, 32(2):477–496, 1997.
- [108] R. Hannam. *Computer Integrated Manufacturing – from concepts to realisation*. Addison-Wesley, Harlow, England, 1997.
- [109] J. Harrington. *Computer Integrated Manufacturing*. R.E. Kreiger, Huntington, NY, 1973.
- [110] T. Hasegawa, L. Gou, S. Tamura, P. Luh, and J. Oblak. Holonic planning and scheduling architecture for manufacturing. In *Proceedings of the International Working conference on Cooperative Knowledge Based systems*, pages 141–157, Keele, UK., 1994.
- [111] J. Hatvany. Intelligence and cooperation in heterarchic manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, 2(2):101–104, 1985.
- [112] H. Hayashi. The ims international collaborative program. In *Proceedings of the 24th International Symposium on Industrial Robots*, 1993.
- [113] T. Heikkila, M. Jarviluoma, and J. Hasemann. Holonic control of a manufacturing robot cell. Technical report, VTT Automation, 1995.
- [114] T. Heikkila, M. Jarviluoma, and T. Juntunen. Holonic control for manufacturing systems: Functional design of a manufacturing robot cell. *Integrated Computer Aided Engineering*, 4:202–218, 1997.
- [115] T. Heikkila and J. Roning. Pem-modelling: A framework for designing intelligent robot control. *Journal of Robotics and Mechatronics*, 4(5):437–444, 1992.
- [116] B. Henoch. A strategic model for reliability and availability in automatic manufacturing. *International Journal of Advanced Manufacturing Technology*, 3(5):99–121, 1988.

- [117] R. Hino and T. Moriwaki. Decentralized scheduling in holonic manufacturing system. In *Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems*, pages 41–47, Leuven, Belgium, 2001.
- [118] D. Hoiomt, P. Luh, and K. Pattipati. Job shop scheduling. In *Firt International Conference on Automation Technology*, pages 565–574, Taipei, Taiwan, 1990.
- [119] D. Hoiomt, P. Luh, and K. Pattipati. A practical approach to job shop scheduling problems. *IEEE Transactions on Robotics and Automation*, 9(1), 1993.
- [120] D.J. Hoiomt, J.B. Perkins, and P.B. Luh. Distributed scheduling of job shops. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2, pages 1067–1072, Sacramento, CA, USA, 1991.
- [121] M. Horvath, A. Markus, and J. Vancza. Cooperation via conflicts in manufacturing systems. In *Cooperative Knowledge Processing for Design of Engineering Systems*, 1995.
- [122] M. Huhns and M. Singh. *Readings in Agents*. Morgan Kaufmann, 1998.
- [123] Deneb Robotics Inc. *IGRIP User Manual and Tutorials*. Deneb Robotics Inc, Auburn Hills, MA, 1992.
- [124] K. Iwata, M. Onosato, and M. Koike. Random manufacturing system: a new concept of manufacturing systems for production to order. *CIRP Annals*, 43(1):379–383, 1994.
- [125] JADE. Jade: Java agent developement framework. Technical report, Telecom Italia Laboratory, <http://sharon.csel.it/projects/jade/>, 2004.
- [126] D. Jarvis, J. Jarvis, D. McFarlane, A. Lucas, and R. Ronnquist. Implementing a multi-agent systems approach to collaborative autonomous manufacturing operations. In *IEEE Aerospace Conference Proceedings*, volume 6, pages 62803–62811, Big Sky, MT, 2001.
- [127] A.T. Jones and C.R. McLean. A proposed hierarchical control model for automated manufacturing systems. *Journal of Manufacturing Systems*, 5(1):15–26, 1986.
- [128] M. Kawage. Holonic control of an assembly robot. In *Proceedings of the Annual Conference of Robotics Society of Japan*, Japan, 1994.

- [129] F. Kimura. Product and process modelling as a kernel for virtual manufacturing environment. *CIRP Annals*, 42(1):147–150, 1993.
- [130] D. Kirsch, D. Wieczorek, and S. Albayrak. Open agent architecture for the realization of holonic manufacturing systems. In *IMS 1998*, pages 407–415, 1998.
- [131] T. Kis, J. Vancza, and A. Markus. Controlling distributed manufacturing systems by a market mechanism. In *ECAI '96: 12th European Conference on Artificial Intelligence*, 1996.
- [132] L. Kleinrock. Distributed systems. *Communications of the ACM*, 28(11):1200–1213, 1985.
- [133] D. Koenig. *Computer Integrated Manufacturing – Theory and Practice*. Hemisphere, 1990.
- [134] A. Koestler. *The Ghost in the Machine*. Hutchinson and Co, London, 1967.
- [135] D. Kotak, S. Bardi, W.A. Gruver, and K. Zohrevand. Comparison of hierarchical and holonic shop floor control using a virtual manufacturing environment. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 1667–1672, Nashville, TN, USA, 2000.
- [136] Rannanjarvi L and H. T. Software development for holonic manufacturing systems. *Computers in Industry*, 37(3):233–253, 1998.
- [137] G. Langer. *HoMuCS - A methodology and architecture for holonic multi-cell control systems*. PhD thesis, Manufacturing Engineering: Technical University of Denmark. Lyngby, Denmark., 1999.
- [138] G. Langer and L. Alting. An architecture for agile shop floor control systems. *Journal of Manufacturing Systems*, 19(4):267–281, 2000.
- [139] G. Langer and A. Bilberg. Architectural considerations for holonic shop floor control. In *Proceedings of the International symposium on Manufacturing Systems (ISMS) at the world Manufacturing Congress*, Auckland, New Zealand, 1997.
- [140] G. Langer, C. Sorensen, and M. Larsen. Product state models in holonic shop floor control systems. In *Proceedings from the 2nd international workshop on intelligent manufacturing systems*, pages 23–33, Leuven, Belgium, 1999.

- [141] A.G. Laws, A. Taleb-Bendiab, and S.J. Wade. Towards a viable reference architecture for multi-agent supported holonic manufacturing systems. *Journal of Applied Systems Science: Special Issue*, 2(1):61–81, 2001.
- [142] H. Li and L.X. Li. Integrating systems concepts into manufacturing information systems. *Systems Research and Behavioural Science*, 17(2):135–147, 2000.
- [143] J.T. Lin and M.P. Tsai. Designing a holon-based manufacturing control system for automated storage and retrieval systems. *Journal of Applied Systems Science: Special Issue*, 2(1):127–151, 2001.
- [144] R. Lipsett, C. Schaeffer, and C. Ussery. Vhdl: Hardware description and design. Technical report, Kluwer Academic Publishers, Boston, MA, 1989.
- [145] F. Liu and P. Luh. Scheduling and coordination of distributed design projects. In *Proceedings of the 14th IFAC World Congress*, Beijing, China, 1999.
- [146] F. Liu, P. Luh, and B. Moser. Scheduling and coordination of distributed design project. *Annals of CIRP*, 47(1):111–114, 1998.
- [147] S. Liu. *Architecture and Coordination of a Holonic Automated Guided Vehicle System*. PhD thesis, Simon Fraser University, 1999.
- [148] S. Liu, W.A. Gruver, and D.B. Kotak. Holonic coordination and control of an automated guided vehicle system. *Integrated Computer-Aided Engineering*, 9(3):235–250, 2002.
- [149] Insoft Ltd. *PROSA Structured Analysis and Design: User's Manual*. Insoft Ltd, Oulu, Finland, 3.30 edition, 1991.
- [150] P. Luh and D. Hoiomt. Scheduling of manufacturing systems using the lagrangian relaxation technique. *IEEE Transactions Auto Control*, 38(7):1066–1079, 1993.
- [151] P. Maes. The agent network architecture (ana). *SIGART Bulletin*, 2(4):135–147, 1991.
- [152] V. Marik and M. Pechoucek. Holomas '00: Industrial applications of holonic and multi-agent systems. In *Workshop on Industrial Applications of Holonic and Multi-Agent Systems. 11th International conference on Database and Expert Systems Applications*, London, UK, 2000.

- [153] V. Marik, M. Pechoucek, J. Lazansky, and C. Roche. Pvs '98 agents:structures, models, and production planning application. *International Journal of Robotics and Autonomous Systems: Special issue on Multi-Agent Systems Applications*, 27(1-2):29–43, 1999.
- [154] V. Marik, M. Pechoucek, and O. Stepankova. Acquaintance model in re-planing and re-configuration. In *Advances in Networked Enterprises*, pages 175–186, Boston, 2000.
- [155] V. Marik, M. Pechoucek, and O. Stepankova. Proplant: Multi-agent system for production planning. *International Journal of Applied Artificial Intelligence*, 14(7):727–762, 2000.
- [156] V. Marik, M. Pechoucek, and O. Stepankova. Social knowledge in multi-agent systems. In *Lecture Notes in Artificial Intelligence (LNAI)*, pages 211–245, Berlin, 2001.
- [157] A. Markus, T.K. Vancza, and L. Monostori. Market approach to holonic manufacturing. *CIRP Annals - Manufacturing Technology*, 45(1):433–436, 1996.
- [158] J. Mathews. Organizational foundations of intelligent manufacturing systems - the holonic viewpoint. *Computer Integrated Manufacturing Systems*, 8(4):237–243, 1995.
- [159] J. Mathews. Organizational foundations of object-oriented programming. *Journal of Systems Software*, 34(3):247–253, 1996.
- [160] J. Matson and D. McFarlane. Tools for assessing the responsiveness of existing production operations. In *IEE Colloquium (Digest)*, number 213, pages 8–1, London, UK, 1998.
- [161] F. Maturana, S. Balasubramanian, and D. Norrie. A multi-agent approach to integrated planning and scheduling for concurrent engineering. In *Proceedings of the International Conference on Concurrent Engineering: Research and Applications*, pages 272–279, Toronto, Ontario, 1996.
- [162] F. Maturana, S. Balasubramanian, and D.H. Norrie. Learning coordination patterns from emergent behavior in a multi-agent based manufacturing system. In *Proceedings of ISAS 97, Intelligent Systems and Semiotics '97: A Learning Perspective*, Gaithersburg, Maryland, 1997.

- [163] F. Maturana, S. Balasubramanian, and D. Norrie. Intelligent multi-agent coordination system for advanced manufacturing. In *Conference on Intelligent Systems for Advanced Manufacturing*, pages 202–212, Pittsburgh, PA, 1997.
- [164] F. Maturana and D. Norrie. A generic mediator for multi-agent coordination in a distributed manufacturing system. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 952–957, 1995.
- [165] F. Maturana and D. Norrie. Distributed decision making using the contract net within a mediator architecture. *Decision Support Systems Journal, Special Issue*, 20(1):53–64, 1996.
- [166] F. Maturana and D. Norrie. Multi-agent co-ordination using dynamic virtual clustering in a distributed manufacturing system. In *Industrial Engineering Research - Conference Proceedings, IIE*, pages 477–496, 1996.
- [167] F. Maturana and D. Norrie. Multi-agent mediator architecture for distributed manufacturing. *Journal of Intelligent Manufacturing*, 7(4):257–270, 1996.
- [168] F. Maturana, W. Shen, M. Hong, and D.H. Norrie. Multi-agent architectures for concurrent design and manufacturing. In *IASTED International Conference on Artificial Intelligence and Soft Computing*, pages 355–359, Banff, 1997.
- [169] F. Maturana, W. Shen, and D.H. Norrie. Metamorph: an adaptive agent-based architecture for intelligent manufacturing. *International Journal of Production Research*, 37(10):2159–2173, 1999.
- [170] D. McFarlane. Holonic manufacturing systems in continuous processing: Concepts and control requirements. In *Proceedings of ASI 95*, pages 282–273, Portugal, 1995.
- [171] D. McFarlane. Modular distributed manufacturing systems and the implications for integrated control. In *IEE Colloquium (Digest)*, number 280, pages 5–1, London, UK, 1998.
- [172] D. McFarlane and S. Bussmann. Developments in holonic production planning and control. *Production Planning and Control*, 11(6):522–536, 2000.
- [173] D. McFarlane, J. Carr, M. Harrison, and A. McDonald. Auto-id’s three r’s: Rules and recipes for product requirements. Technical report, Auto-ID Centre, Cambridge, UK, 2002.

- [174] D. McFarlane, S. Sarma, J.L. Chirn, K. Ashton, and C.Y. Wong. Auto id systems and intelligent manufacturing control. *Engineering Applications of Artificial Intelligence*, 16(4):365–376, 2003.
- [175] D.C. McFarlane and S. Bussmann. Holonic manufacturing control: Rationales, developments and open issues. In S.M. Deen, editor, *Agent-Based Manufacturing*, chapter 13, pages 303–326. Springer-Verlag, Berlin, 2003.
- [176] R. McNab and F.W. Howell. Using java for discrete event simulation. In *Proceedings of the 12th UK Computer and Telecommunications Performance Engineering Workshop*, pages 219–228, University of Edinburgh, 1996.
- [177] J.P. Muller. *The Design of Intelligent Agents: A Layered Approach*. Springer-Verlag New York, Inc., 1997.
- [178] T. Neligwa and S.M. Deen. A study of the hms operational architecture. In *IEEE Proceedings 13th International Workshop on Database and Expert Systems Applications 2002.*, pages 516–520, 2002.
- [179] A. Ng, R. Yeung, and E. Cheung. Hscs - the design of a holonic shopfloor control system. In *Proceedings of IEEE Conference on Emerging Technologies and Factory Automation*, pages 179–185, 1996.
- [180] G. Nucci Franco and A. Batocchio. Towards an axiomatic framework to support the design of an holonic system. In *Second International Workshop on Industrial Applications of Holonic and Multi-Agent Systems (Holomas 2001)*, Prague, 2001.
- [181] P.D. O’Brien and R.C. Nico. Fipa - towards a standard for software agents. *BT Technology Journal*, 16(3):51–59, 1998.
- [182] Technical Research Centre of Finland Electronics Laboratory. Velvet user manual: The modelling guidelines and the usage of velvet. Technical report, Technical Research Centre of Finland – Electronics Laboratory, Oulu, Finland, 1993.
- [183] N. Okino. A prototyping of bionic manufacturing systems. In *Proceedings of ICOOMS*, pages 297–302, 1992.
- [184] N. Okino. Bionic manufacturing systems. In *Proceedings of the CIRP Seminars on Manufacturing Systems: Flexible Manufacturing Systems: Past, Present, Future*, volume 23, pages 73–95, Ljubljana, Slovenia, 1993.

- [185] H. Park, J. Tenenbaum, and R. Dove. Agile infrastructure for manufacturing systems: A vision for transforming the us manufacturing base. In *Defense Manufacturing Conference*, 1993.
- [186] H. Parunak. Autonomous agent architectures: A non-technical introduction. Technical report, Industril Technology Institute, 1993.
- [187] H. Parunak. Rappid project index page. Technical report, <http://www.iti.org/cec/rappid/>, 1997.
- [188] H. Parunak. Industrial and practical applications of dai. In G. Weiss, editor, *Multi-Agent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 377–424. MIT Press, Cambridge, MA, 1999.
- [189] M. Pechoucek, F. Macurek, P. Tichy, O. Stepankova, and V. Marik. Meta-agent a workflow mediator in multi-agent systems. In *Intelligent Workflow and Process Management The New Frontier for AI in Business*, volume 1, pages 110–116, San Francisco, 1999.
- [190] M. Pechoucek, V. Marik, and O. Stepankova. Coalition formation in manufacturing multi-agent systems. In *Workshop on Industrial Applications of Holonic and Multi-Agent Systems, 11th International Conference on Database and Expert Systems Applications*, pages 241–246, London, UK, 2000.
- [191] M. Pechoucek, V. Marik, and O. Stepankova. Towards reducing communication traffic in multi-agent systems. *Journal of Applied Systems Science: Special Issue*, 2(1):152–174, 2001.
- [192] Y. Peng, T. Finin, Y. Labrou, B. Chu, J. Long, W. Tolone, and A. Boughannam. A multi-agent system for enterprise integration. In *Proceedings of PAAM '98*, pages 533–548, 1998.
- [193] P. Prosser and I. Buchanan. Intelligent scheduling: past, present and future. *IEEE Intelligent Systems Engineering*, 3(2):67–78, 1994.
- [194] RACAL-REDAC and IKOS Systems Inc. *VHDL2000 User's Manual*. RACAL-REDAC and IKOS Systems Inc., Westford, MA, 1992.

- [195] C. Ramos. A holonic approach for task scheduling in manufacturing systems. In *Proceedings of IEEE Conference on Robotics and Automation*, volume 3, pages 2511–2516, Minneapolis, 1996.
- [196] S.P. Rana and S.K. Taneja. A distributed architecture for automated manufacturing systems. *International Journal of Advanced Manufacturing Technology*, 3(5):81–98, 1988.
- [197] A.S. Rao and M.P. Georgeff. An abstract architecture for rational agents. In C. Rich, W. Swartout, and B. Nebel, editors, *Proceedings of Knowledge Representation and Reasoning (KRgR-92)*, pages 439–442, 1992.
- [198] J. Rehg and H. Kraebber. *Computer Integrated Manufacturing*. Prentice Hall, first edition, 1994.
- [199] U. Rembold, B.O. Nnaji, and A. Storr. *Computer Integrated Manufacturing and Engineering*. Addison-Wesley, Wokingham, England, 1993.
- [200] G. Rzevski. Framework for designing intelligent manufacturing systems. *Computers in Industry*, 34(2):211–219, 1997.
- [201] A. Saad, K. Kawamura, and G. Biswas. Performance evaluation of contract net based hierarchical scheduling for flexible manufacturing systems. Technical report, Centre for Intelligent Systems, Vanderbilt University, Nashville, TN, 1997.
- [202] J. Saggi. Factory agile - european agile enterprise. cest eureka project. Technical report, <http://www.sarc.sk/akcie/eureka/data/1548.html>, 1996.
- [203] T.W. Sandholm and V.R. Lesser. Equilibrium analysis of the possibilities of unenforced exchange in multiagent systems. In C.S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 694–701, Montreal, Canada, 1995.
- [204] T.W. Sandholm and V.R. Lesser. Advantages of a leveled commitment contracting protocol. In *Proceedings of the National Conference on Artificial Intelligence*, volume 1, pages 126–133, Portland, OR, USA, 1996.
- [205] C. Schaeffer and J. Sieverding. Holonic production and material flow in industry. In *HMS Symposium 2000, Japan*, 2000.

- [206] J. Schnell, S. Andersen, G. Langer, and C. Sorensen. Development of a robot holon using an open modular controller. *IEEE Conference on Control Applications - Proceedings*, 2:1642–1647, 1999.
- [207] D. Seidel, M. Hoepf, J.M. Prado, E. Garcia-Herreros, T. Strasser, and J. Oblak. Ims – holonic manufacturing systems: System components of autonomous modules and their distributed control. volume HMS – Strategies, Vol. 0, 1994.
- [208] A. Shaw, D. McFarlane, Y. Chang, and P. Noury. Measuring response capabilities in the order fulfillment process. In *Proceedings 9th International Conference European Operations Management Association (EurOMA)*, Copenhagen, Denmark, 2002.
- [209] M.J. Shaw. Dynamic scheduling in cellular manufacturing systems: A framework for networked decision making. *Journal of Manufacturing Systems*, 7(2):83–94, 1988.
- [210] W. Shen, F. Maturana, and D. Norrie. Metamorph ii: an agent-based architecture for distributed intelligent design and manufacturing. *Journal of Intelligent Manufacturing*, 11(3):237–251, 2000.
- [211] W. Shen and D. Norrie. Facilitator, mediator or autonomous agents. In *In Proceedings of the Second International Workshop on CSCW in Design*, pages 119–124, Bangkok, Thailand, 1997.
- [212] W. Shen and D. Norrie. An agent-based approach for dynamic manufacturing scheduling. In *In Workshop Notes of the Agent-Based Manufacturing Workshop at Autonomous Agents 1998*, pages 117–128, Minneapolis, MN, 1998.
- [213] W. Shen and D. Norrie. An agent-based manufacturing enterprise infrastructure for distributed integrated intelligent manufacturing systems. In *Globalization of Manufacturing in the Digital Communications Era of the 21st Century: Innovation, Agility, and the Virtual Enterprise*, pages 579–590, London, UK, 1998.
- [214] W. Shen and D. Norrie. A hybrid agent-oriented infrastructure for modeling manufacturing enterprises. In *In Proceedings of KAW '98*, Banff, Canada, 1998.
- [215] W. Shen and D. Norrie. A mediator-centric architecture for holonic manufacturing systems. In *Proceedings of the Sixth IASTED International Conference on Robotics and Manufacturing*, pages 73–76, Banff, Canada, 1998.

- [216] W. Shen and D. Norrie. Agent-based systems for intelligent manufacturing: A state-of-the-art survey. *Knowledge and Information Systems: An International Journal*, 1(2):129–156, 1999.
- [217] W. Shen, D. Norrie, and J.P. Barthes. *MultiAgent Systems for Concurrent Intelligent Design and Manufacturing*. Taylor and Francis, 2000.
- [218] W. Shen and D.H. Norrie. Combining mediation and bidding mechanisms for agent-based manufacturing scheduling. In *Proceedings of the Interantional Conference on Autonomous Agents*, pages 469–470, Minneapolis, MN, USA, 1998.
- [219] W. Shen, D. Xue, and D. Norrie. An agent-based manufacturing enterprise infrastructure for distributed integrated intelligent manufacturing systems. 1997.
- [220] J. Shin and H. Cho. Planning and sequencing heuristics for feature-based control of holonic machining equipment. *The International Journal of Flexible Manufacturing Systems*, 13(1):49–70, 2001.
- [221] N. Silva and C. Ramos. Infrastructures and scheduling method for holonic manufacturing systems. *Proceedings of the IEEE International Symposium on Assembly and Task Planning*, pages 442–447, 1999.
- [222] H.A. Simon. *The Science of the Artificial*. MIT Press, Cambridge, MA, second edition, 1990.
- [223] J.A. Simpson, R.J. Hocken, and J.S. Albus. The automated manufacturing research facility of the national bureau of standards. *Journal of Manufacturing Systems*, 7(4):315–327, 1982.
- [224] R.G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, 1980.
- [225] C. Sorensen, G. Langer, and L. Alting. Developing a system architecture for holonic shop floor control. In *Proceedings of IFAC IMS'98*, Brazil, 1998.
- [226] P. Sousa and C. Ramos. A dynamic scheduling holon for manufacturing orders. *Journal of Intelligent Manufacturing*, 9(2):107–112, 1998.

- [227] P. Sousa and C. Ramos. Distributed architecture and negotiation protocol for scheduling in manufacturing systems. *Computers in Industry*, 38(2):103–113, 1999.
- [228] P. Sousa and C. Ramos. Distributed task scheduling. *Proceedings of the IEEE International Symposium on Assembly and Task Planning*, pages 436–441, 1999.
- [229] J.A. Stankovic. Misconceptions about real-time computing: a serious problem for next-generation systems. *Computer*, 21(10):10–19, 1988.
- [230] Suda. Future factory system formulated in japan. *Japanese Journal of Advanced Automation Technology*, 2(10):15–25, 1989.
- [231] Suda. Future factory system formulated in japan (2). *Japanese Journal of Advanced Automation Technology*, 23(3):51–61, 1990.
- [232] J. Sun, D. Xue, and D.H. Norrie. An intelligent production scheduling mechanism considering design and manufacturing constraints. In *The Third International Conference on Industrial Automation*, Montreal, 1999.
- [233] C. Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley, ACM Press New York, 1997.
- [234] IEC TC65/WG6. Voting draft – publicly available specification – function blocks for industrial process-measurement and control systems, part 1 - architecture. Technical report, International Electrotechnical Commission, 2000.
- [235] A. Tharumarajah, A. Wells, and L. Nemes. A comparison of the bionic, fractal and holonic manufacturing concepts. *International Journal of Computer Integrated Manufacturing*, 9(3):217–226, 1996.
- [236] A. Tharumarajah, A.J. Wells, and L. Nemes. Comparison of emerging manufacturing concepts. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 325–331, San Diego, CA, USA, 1998.
- [237] K. Ueda. A genetic approach toward future manufacturing systems. In *Proceedings of ICOOMS*, pages 303–308, 1992.
- [238] K. Ueda. A genetic approach toward future manufacturing systems. In *Proceedings of the CIRP Seminars on Manufacturing Systems: Flexible Manufacturing Systems: Past, Present, Future*, pages 221–228, Ljubljana, Slovenia, 1993.

- [239] P. Valckenaers. *Flexibility for Integrated Production Automation*. PhD thesis, KU Leuven, 1993.
- [240] P. Valckenaers, F. Bonneville, H. Van Brussel, and J. Wyns. Results of the holonic control system benchmark at kuleuven. In *4th Rensselaer International Conference on Computer Integrated Manufacturing and Automation Technology*, pages 128–133, Rensselaer, NY, 1994.
- [241] P. Valckenaers, H.V. Brussel, L. Bongaerts, and F. Bonneville. Programming, scheduling, and control of flexible assembly systems. *Computers in Industry*, 26(3):209–218, 1995.
- [242] P. Valckenaers, H. Van Brussel, F. Bonneville, L. Bongaerts, and J. Wyns. Ims test case 5. holonic manufacturing systems. page 31, Vienna, Aust, 1994.
- [243] P. Valckenaers, H. Van Brussel, J. Wyns, L. Bongaerts, and P. Peeters. Designing holonic manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, 14(5-6):455–464, 1998.
- [244] H. Van Brussel. Holonic manufacturing systems, the vision matching problem. In *Proceedings of the 1st European Conference on Holonic Manufacturing Systems*, Hannover, Germany, 1994.
- [245] H. Van Brussel, L. Bongaerts, J. Wyns, P. Valckenaers, and T. VanGinderachter. A conceptual framework for holonic manufacturing: Identification of manufacturing holons. *Journal of Manufacturing Systems*, 18(1):35–52, 1999.
- [246] H. Van Brussel, Wyns J., Valckenaers P, .Bongaerts L., and Peeters P. Reference architecture for holonic manufacturing systems. *Computers in Industry*, 37(3):255–274, 1998.
- [247] H. Van Brussel, Y. Peng, and P. Valckenaers. Modelling flexible manufacturing systems based on petri nets. *CIRP Annals*, 42(1):479–484, 1993.
- [248] H. Van Brussel, P. Valckenaers, J. Wyns, L. Bongaerts, and J. Detand. Holonic manufacturing systems and iim. In *Proceedings of Integration in Manufacturing Conference*, Galway, 1996.

- [249] J. Vancza and A. Markus. Holonic manufacturing with economic rationality. In *Proceedings of IMS-Europe, 1st International Workshop on Intelligent Manufacturing Systems*, pages 383–394, Lausanne, 1998.
- [250] M. Waldrop. *Complexity: The Emergin Science at the Edge of Order and Chaos*. Simon and Schuster, New York, 1992.
- [251] J. Wang and P. Luh. Scheduling job shops with batch machines using the lagrangian relaxation technique. *European Journal of Control*, 3(4):268–279, 1997.
- [252] L. Wang. Integrated design-to-control approach for holonic manufacturing systems. *Robotics and Computer Integrated Manufacturing*, 17(1-2):159–167, 2001.
- [253] L. Wang, S. Balasubmramanian, and D. Norrie. Agent-based intelligent control system design for real-time distributed manufacturing environments. In *Agent-based Manufacturing Workshop - Autonomous Agents' 98*, pages 152–159, Minneapolis, 1998.
- [254] L. Wang, S. Balasubramanian, D.H. Norrie, and R.W. Brennan. Agent-based control system for next generation manufacturing. In *IEEE International Symposium on Intelligent Control - Proceedings*, pages 78–83, Gaithersburg, MD, USA, 1998.
- [255] L. Wang, R. Brennan, S. Balasubmramanian, and D.H. Norrie. Realizing holonic control with function blocks. *Integrated Computer-Aided Engineering*, 8(1):81–93, 2001.
- [256] L. Wang and D.H. Norrie. Process planning and control in holonic manufacturing environment. *Journal of Applied Systems Science: Special Issue*, 2(1):106–126, 2001.
- [257] P.T. Ward and S.J. Mellor. *Structured Development for Real-Time Systems*, volume I-III. Yourdon Press, New York, 1986.
- [258] G. Weiss. *Multi-Agent Systems*. MIT Press, Cambridge, MA, 1999.
- [259] C.Y. Wong. Integration of auto-id tagging systems with holonic manufacturing. Technical report, Auto-ID Centre Cambridge, Cambridge, UK, 2001.
- [260] M. Wooldridge. Intelligent agents. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 27–77. MIT Press, Cambridge, 1999.
- [261] M. Wooldridge. *Reasoning about Rational Agents*. MIT Press, Cambridge, MA, 2000.

- [262] M. Wooldridge and N.R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.
- [263] M. Wooldridge, N.R. Jennings, and D. Kinny. Methodology for agent-oriented analysis and design. *Proceedings of the Interantional Conference on Autonomous Agents*, pages 69–76, 1999.
- [264] J. Wyns. *Reference Architecture For Holonic Manufacturing Systems - the key to support to evolution and reconfiguration*. PhD thesis, Department Werktuigkunde Afdeling Productietechnieken. , Katholieke Universiteit Leuven, 1999.
- [265] J. Wyns and G. Langer. Holonic manufacturing systems described in plain text, ideo, and object-oriented methods. In *Proceedings of the First International Workshop on Intelligent Manufacturing Systems*, pages 13–28, Lausanne, Switzerland, 1998.
- [266] J. Wyns, H. Van Brussel, and P. Valckenaers. Design pattern for deadlock handling in holonic manufacturing systems. In *Proceedings of Rensselaer’s international conference on agile, intelligent, and computer-integrated manufacturing (CAICIM98)*, Troy, New York, 1998.
- [267] J. Wyns, H. Van Brussel, P. Valckenaers, and L. Bongaerts. Workstation architecture in holonic manufacturing systems. In *Proceedings of the 28th International CIRP seminar on Manufacturing Systems*, Johannesburg, 1996.
- [268] W.H.R. Yeung and P.R. Moore. Object-oriented modelling and control of flexible conveyor systems for automated assembly. *Mechatronics*, 6(7):799–815, 1996.
- [269] C. Yinong and H. Zhongshi. Dependability modelling of homogeneous and heterogeneous distributed systems. pages 176–183, 2001.
- [270] A.A. Zaharudin, C.Y. Wong, D. McFarlane, and V. Agarwal. The intelligent product driven supply chain. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 393–398, Yasmine Hammamet, Tunisia, 2002.
- [271] X. Zhang, S. Balasubramanian, R. Brennan, and D. Norrie. Design and implementation of a real-time holonic control system for manufacturing. *Information Sciences*, 127:23–44, 2000.

- [272] X. Zhang, R. Brennan, and D. Norrie. Towards a function block-based holonic controller. In *2nd CIRP International Seminar on Intelligent Computation in Manufacturing Engineering (ICME 2000)*, pages 43–48, Capri, Italy, 2000.
- [273] X. Zhang, R. Brennan, D. Norrie, and X. Yuefei. Runtime adaptability of a concurrent function block model for a real-time holonic controller. In *Proceedings of the 2001 IEEE Systems, Man and Cybernetics Conference*, 2001.
- [274] X. Zhang, D.H. Norrie, R.W. Brennan, and Y. Xu. Multi-level reconfiguration control for holonic plc. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 1762–1767, Nashville, TN, USA, 2000.
- [275] B. Zhou, L. Wang, and D. Norrie. Design of distributed real-time control agents for intelligent manufacturing systems. In *Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems*, pages 237–244, Leuven, Belgium, 1999.