The Holonic Enterprise as a Collaborative Information Ecosystem

Mihaela Ulieru
University of Calgary
Intelligent Systems Group
2500 University Drive N.W.
Calgary, AB, Canada T2N 1N4
Phone: +1-403-220-8616
ulieru@enel.ucalgary.ca
http://isg.enme.ucalgary.ca/

Scott S. Walker
(same)
Phone: +1-403-220-5798
swalker@enme.ucalgary.ca

Robert W. Brennan
(same)
Phone: +1-403-220-2991
brennan@enme.ucalgary.ca

ABSTRACT

In today's e-conomy the only chance for prosperity is to exploit optimally the emerging technologies based on which a new kind of infrastructure facilitates strategic partnerships among cyber-highway enabled participants. This paper merges the latest results obtained by the Holonic Manufacturing Systems (HMS) Consortium with the latest developed standards for platform interoperability released by the Foundation for Intelligent Physical Agents (FIPA) to propose a novel e-business model: the Holonic E-nterprise. Including the e-marketplace and e-factory as sub-models, this new paradigm links the three levels of a global collaborative organization (inter-enterprise, intra-enterprise and machine level) to build a web-centric ecosystem partnering in which the workflow is harmoniously managed. After clarifying the proposed concept we define a mapping between holons and agents, introducing the concept of mediator. We identify several patterns of holonic collaboration and throughout the paper identify their particularities at each level. The Holonic Enterprise extends both the HMS and FIPA models. On one side it extends the holonic manufacturing paradigm with one top level, the inter-enterprise one. On the other side it extends the multi-agent system (MAS) paradigm to the hardware (physical machine) level.

Keywords

Holonic Systems, Multi-Agent Systems, Global Collaborative Enterprises, e-Business, Supply Chain Management, Dynamic Scheduling, Reconfigurable Production, Distributed Intelligent Control.

1. INTRODUCTION

A holonic enterprise is a holarchy of collaborative enterprises, where the enterprise is regarded as a holon. (Here the term enterprise is used in a broad, generic manner: entity, system, 'thing', agent). The term holon was coined by Arthur Koestler [23] to denominate entities simultaneously exhibiting both autonomy and cooperation capabilities which demand balancing of the contradictory forces that define each of these properties on a behavioural level. One main characteristic of a holon is its multiple granularity manifested through replication into self-similar structures at multi-resolution levels. Such a heterarchical decomposition turns out into a nested hierarchy of fractal entities—named a holarchy. A holonic enterprise has three levels of granularity, as illustrated in Figure 1.

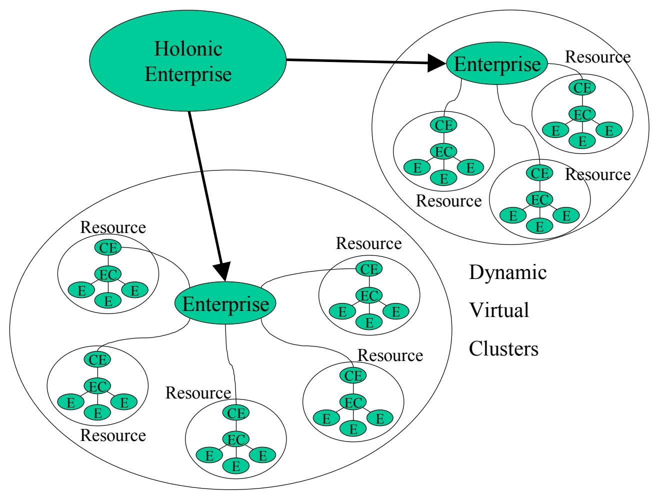


Figure 1 – Dynamic Virtual Clustering Pattern in the Holonic Enterprise

1.1 Global Inter-Enterprise Collaborative Level

At this level, several holon-enterprises cluster into a collaborative holarchy to produce products or services. The clustering criteria support maximal synergy and efficiency. Traditionally this level was regarded as a mostly static chain of customers and suppliers through which the workflow and information was moving from the end customer who required the product to the end supplier who delivered it. In the holonic enterprise the supply chain paradigm is replaced by the collaborative holarchy paradigm (Fig. 1). With each collaborative partner modeled as an agent that encapsulates those abstractions relevant to the particular cooperation, a dynamic virtual cluster (Fig. 1) emerges which can be configured on-line according to the collaborative goals (e.g. by finding the best partners for the collaboration). Such a dynamic collaborative holarchy can cope with unexpected disturbances (e.g. replace a collaborative partner who cannot deliver within the deadline) through on-line reconfiguration of the open system it represents. It provides on-line order distribution across the available partners as well as deployment mechanisms that ensure real-time order error reporting and on-demand order tracking.

1.2 Intra-Enterprise Level

Once each enterprise has undertaken responsibility for the assigned part of the work, it has to organize in turn its own internal resources to deliver on time according to the coordination requirements of the collaborative cluster. Planning and dynamic scheduling of resources at this level enable functional reconfiguration and flexibility via (re)selecting functional units, (re)assigning their locations, and (re)defining their interconnections (e.g., rerouting around a broken machine, changing the functions of a multi-functional machine). This is achieved through a replication of the dynamic virtual clustering mechanism, having now each resource within the enterprise cloned as an agent which abstracts those functional characteristics relevant to the specific task assigned by the collaborative holarchy to the partner. Reconfiguration of schedules to cope with new orders or unexpected disturbances (e.g. machine failure) is enabled through re-clustering of the agents representing the actual resources of the enterprise, as illustrated in Figure 2. The main criteria for resource (re)allocation when (re)configuring the schedules are related to cost minimization achieved via multi-criteria optimization.

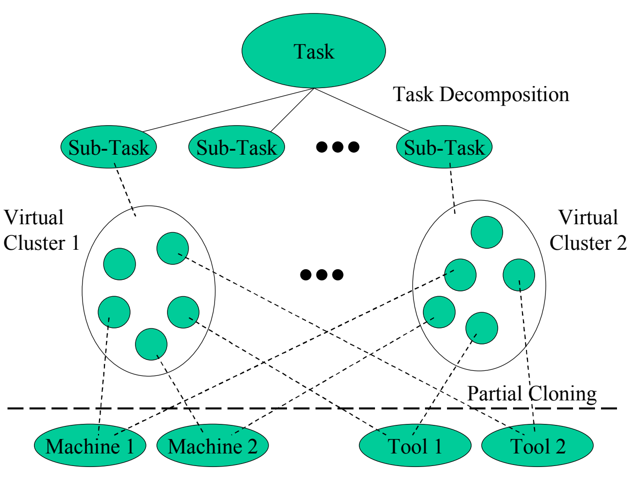


Figure 2 – Task Distribution Pattern at the Intra-Enterprise Level

1.3 Machine (Physical Agent) Level

This level is concerned with the distributed control of the physical machines that actually perform the work. To enable agile manufacturing through the deployment of self-reconfiguring, intelligent distributed automation elements (see Figure 3), each machine is cloned as an agent which abstracts those parameters needed for the configuration of the holonic control system managing the distributed production.

2. PERSPECTIVES ON HOLONS AND AGENTS

Increasingly, there exists in today's society a culture of holistic approaches to all aspects of life. For example, people are embracing:

- Alternative/holistic medicines – with an emphasis on maintaining optimum health proactively, and not just treating individual symptoms or conditions;
- Sustainable development – with the recognition that that our use of technology has an impact on our environment (i.e. the rest of the systems);
- Concurrent engineering – with the recognition that individual engineering endeavours do not occur independently "in a vacuum", but rather are intimately linked, including design, shop floor control, supply chain management, et cetera; and,
- Object-Oriented and Agent-Oriented Software Engineering.

This holistic approach is related to the concept of the "holon", envisioned in the 1960's by Koestler [23].

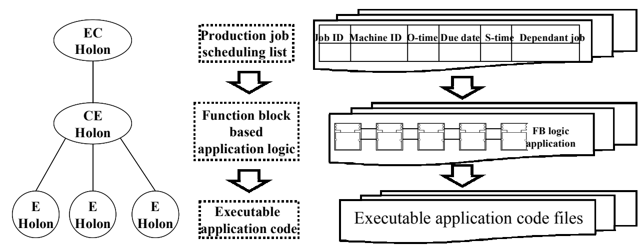


Figure 3 – Task Deployment Pattern at the Holonic Control Level

2.1 Holons As Bricks For Building Complex Systems

The word "holon" is an amalgamation of the Latin word "holos" meaning whole, and the suffix "on" which implies particle or part. So, holon = whole & part. The term comes from a recognition that any entity in a natural or organizational system is simultaneously a whole entity comprised of sub-entities, and a sub-entity portion of one or more super-entities. Also implicit is a recognition that systems (natural or organizational) cannot evolve from one level of complexity to significantly higher levels of complexity without the existence of stable intermediate forms (which are obviously also holons) to act as intermediate steps toward the goal state. This recognition is already implicit in some areas of engineering, such as Object-Oriented (O-O) Software development [30] in which objects are composed of objects, each of which can be considered a stable intermediate form or stage (i.e. holon).

The development of complex systems—such as global manufacturing consisting of: automated business-to-business communications, supply chain management and Just-In-Time delivery; web-centric/consumer-centric sales, delivery and service; shop floor production, planning and control—would benefit from the use of a holonic approach which enables workflow management through robust e-business models that channel the flow of information throughout the ecosystem partnering. Such a model has the following advantages:

- Eases the access to business and enables development of strategic partnerships with remote players in the global market;
- Increases the visibility of the enterprise;
- Simplifies the design and development process by enabling incremental solution development. (The entire solution need not appear all at once, but rather can be developed incrementally with stable intermediate forms/holons as sub-goals which are themselves useful and worthwhile entities, but which become significantly more so when integrated with other stable forms/holons to create more completely developed forms/holons.)
- Results in more robust (natural) systems, as these intermediate forms/holons are themselves stable and self-sufficient, for e.g. integratable stack-like solutions shown in Fig. 4 (http://www.hourgroup.com/e-energy/reg2001.html).
- Enable the paradigm shift in business and technology by transfer of natural models—such as evolutionary algorithms, emergent intelligence, parallel computation, et cetera.

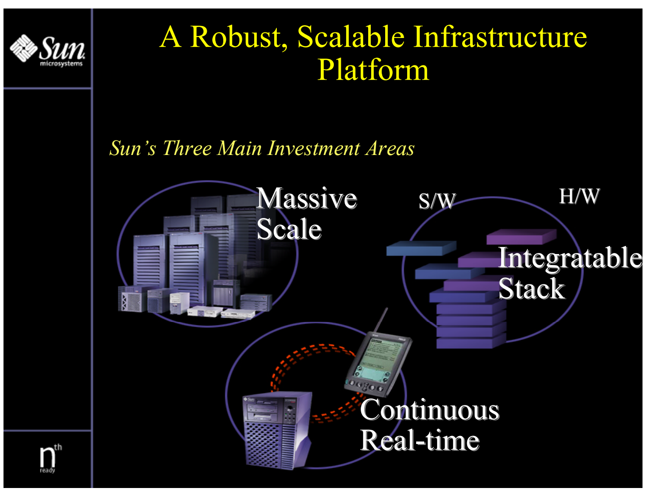


Figure 4 – Holons as Integratable Stacks

The fundamental qualities and characteristics of natural and organizational systems, identified by Koestler with the introduction of his term "holon" [23], seem to match well with the requirements at various levels of an enterprise as will be described in sections 3 through 6. These "holonic" traits were first recognized as being desirable in a manufacturing shop floor production planning, scheduling, and control system—i.e. an intra-enterprise application—which led to the concept of Holonic Manufacturing Systems and the HMS Consortium [17]. These initial ideas were soon being applied to both the inter-enterprise and physical machine levels of the manufacturing enterprise [7]. Most importantly, however, the holonic enterprise approach is applicable to any enterprise domain, as shown in this paper.

2.2 The Relationship Between Holons And Agents

A system decomposition and analysis based on holonic principles naturally suggests a distributed software implementation, with autonomously executing cooperative entities as building blocks. The Multi-Agent Systems (MAS) paradigm seems to be well suited to implementing a holonic abstraction of a problem which is fundamentally distributed in nature; Object Oriented Software Engineering [30] also seems ideally matched to the holonic paradigm, given its recursive structure.

As illustrated in Figure 5, the stable intermediate forms/holons of the system can be implemented at the lower levels by objects, at the medium level by agents, and at the higher levels by groups of agents, with these mapping decisions being application-specific. (Of course, these are only the software portions of the holons.) Also, holons should have an interface which is simple and cohesive, just like in O-O systems [11] or any effective organisational structure, and which is itself a holon.

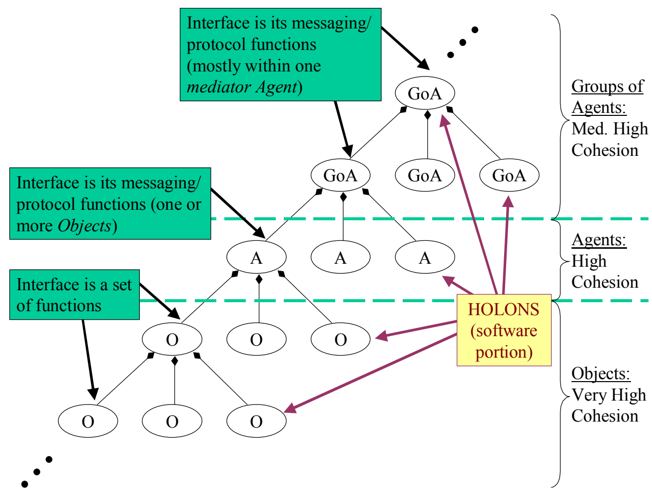


Figure 5 – Multi-granular Decomposition of Holons into Agents and Objects

2.3 Mapping A Holonic System Into A Multi-Agent System: The Mediator

At this point, we will consider the question of how to map a holonic system description into a MAS, and especially the form of the interface when a holon consists of a group of agents, as in the top of Figure 5. Since agents will be the software tool chosen to build this dynamic and interactive system—for reasons already mentioned—it is important to examine early on what specific requirements and limitations are introduced by this tool for this particular task.

How does one build agents and groups of agents which fulfill the holonic philosophy? (The third option, mapping holons into objects, is an almost trivial task that needs no discussion here.) The basic condition for holonic systems is that a holon is simultaneously a "whole" and a "part" of some other whole/holon. This means that holons may contain other lower level holons, and may themselves be contained in other higher level holons, resulting in a recursive architecture. The agents to be used to implement this holonic system will be considered independently executing processes on some machine/device. In this case, if a one-to-one mapping of holon to agent is performed, it is much more difficult to practically implement an agent (than it is to conceptualize a holon) which is itself a component of a higher level agent and which also contains lower level agents.

Here, the concept of a mediator agent comes into play. The mediator will fulfill two main functions. First, it acts as the interface between the agents in the holon and between the agents outside the holon (i.e. acts as a type of facilitator); conceptually, it can be thought of as the agent that represents the holon. Second, it may broker and/or supervise the interactions between the sub-holons of that holon; this also allows the system architect to implement (and later update) a variety of forms of interaction easily and effectively, thereby fulfilling the need for flexibility and reconfigurability. This mediator as described can actually be considered a static mediator, and will exist primarily at the boundary of a homogeneous holon (such as an ordering holon, for example), as illustrated in Figure 6.

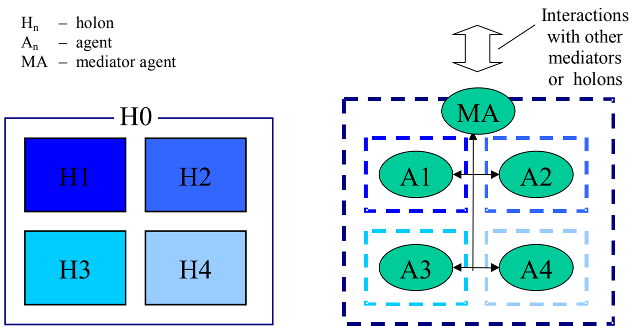


Figure 6 – Mapping Holonic Systems into MAS via Mediators

Another type of mediator is a "dynamic mediator". In the case of interactions between heterogeneous holons, such as an order holon and various resource scheduling holons, ease of system design may be supported by employing a dynamic mediator agent to broker and/or supervise the interactions within a group of holons/agents. These groups, or clusters, represent interactions to accomplish a specific task. These interactions can be called a "conversation", and the group of agents/holons involved can be called a "dynamic virtual cluster" (dynamic because they form and then dissolve as tasks are initiated and then completed, and virtual because they represent a logical or functional decomposition of the system interactions, and not a structural grouping of system components). This, too, is illustrated in Figure 6. Dynamic mediators, dynamic virtual clusters, and conversations are all central concepts of the approach presented in [47].

For the designer of a manufacturing system, the control perspective is often the most important. Implementing the control dimension of a holonic system is a challenge, mainly due to the multi-level view of this controller. Mediators have been used as interfaces to and coordinators of a holonic manufacturing system, as in [31]. For our purpose, the concept of the mediator agent can be very naturally extended to fulfill the role of meta-control, whereby the mediator holarchy is the holonic meta-controller (for those control activities in the "Group of Agents" levels or above of the structural/functional holarchy) [6].

2.4 Collaborative Holons As Information Ecosystems

Figure 7 presents the flow of information throughout the Holonic Enterprise regarded as a Collaborative Information Ecosystem. The MAS inhabits the environment of computers, controllers, and networks; as such, each agent is part of a holon. Its inputs are the machine's sensors, data from storage, and interactions or communications with other agents and with humans; its outputs are the physical control of machines, data to storage, and interactions or communications with other agents and with humans. There are benefits to having heterarchically and hierarchically distributed system components which possess high coherence, low coupling, and some autonomy (i.e. a holonically organized multi-agent system—or HMAS), as discussed by [23] for natural systems, which are used to model enterprise systems at all levels. Design choices and the designer's main technical concerns will vary depending upon the level of the holonic enterprise being considered: the inter-enterprise level, the intra-enterprise level, or the physical machine level.

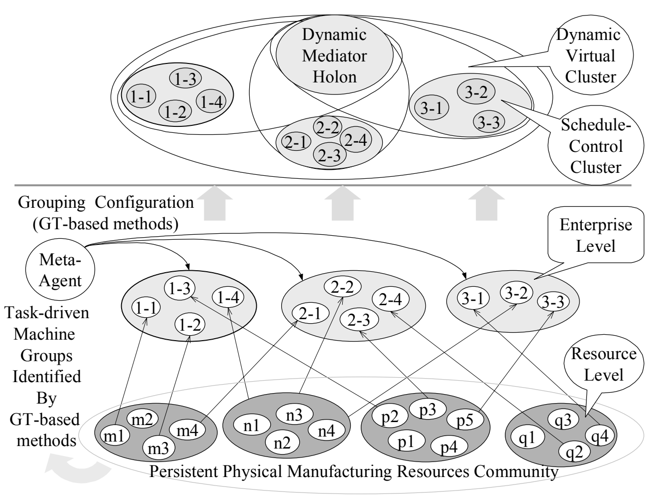


Figure 7 – Holonic Enterprise as a Collaborative Ecosystem

3. PATTERNS OF HOLONIC COLLABORATION

The common mechanisms that characterize the collaborative information ecosystem created by the three levels of a holonic enterprise follow the design patterns for adaptive multi-agent systems identified by [39] (Figure 7). The overall architecture of the Holonic Enterprise builds on the Metamorphic Architecture Pattern which replicates at all levels.

- **Metamorphic Architecture Pattern**. The overall architecture of the Holonic Enterprise builds on this pattern that replicates at all levels.

This pattern works by synergetic integration of two other patterns:

- **Dynamic Virtual Clustering** configured to minimize cost and enabling for flexible, reconfigurable structures. At all levels of the holonic enterprise, task propagation occurs by a process of virtual cluster (or holarchy) formation. This pattern is facilitated by the general layered architecture of the holonic enterprise. Each level described previously is divided into a number of autonomous layers that appear to interact through an API (application programming interface). Code is run asynchronously on these layers, providing functional separation between the layers.
- **Mediator Agent Pattern** supporting the decision-making process that creates and (re)-configures the collaborative cluster of enterprises.

To abstract those characteristics of the entities in each cluster that are relevant for the particular collaboration at each level we use the:

- **Partial Cloning Pattern**. This pattern defines which of the enterprise's characteristics (attributes and functionality) we need to abstract into agents at each level when modeling the holonic enterprise as a collaborative multi-agent system.

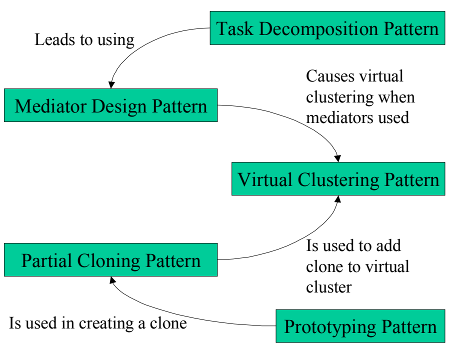


Figure 8 – Pattern Interaction within the Metamorphic Architecture

The workflow coordination throughout the collaborative ecosystem is managed by the mediator agent via the

- **Task Decomposition-Distribution Pattern** [31]. This pattern is enhanced with capability to distribute harmoniously among the participants the overall task assigned to the collaborative holon, at each level. The main mechanisms by which this pattern works are:
  - ***task distribution among the cluster's entities*** (outside-in view from the mediator "down" into each collaborative partner at that level); and,
  - ***task deployment within each entity*** (inside-out view – from the entity, regarded as a holon with distributed resources available to it for accomplishing the assigned task, to the mediator).

Both mechanisms call for appropriate negotiation strategies [20] to enable appropriate loading of each collaborative partner according to its available resources as well as deadline commitment and delay justification through appropriate argumentation and persuasion strategies. An excellent tool for inducing decentralization into the holonic collaboration is the institutionalized power [22] that transfers complete responsibility regarding the "how" of the task's accomplishment to the entity to which the task was assigned once this entity has accepted the delivery conditions.

Propagation of the task decomposition-distribution pattern throughout the granular levels of the holonic enterprise requires two kind of ontologies to enable 'inter-entity' communication, which define an

- **Ontology Pattern**. This consists of two kind of ontologies, namely for 'peer-to-peer' communication at each level (that is inter-agent communication among entities that form a cluster); and for 'inter-level' communication that enables deployment of tasks assigned at higher levels (by the mediator) on lower level clusters of resources.

Although they work at each level to manage the flow of information and materials within the holonic enterprise these patterns have specific particularities within each level of the collaborative holarchy. The purpose of our work is to identify these particularities and clearly define the policies and services supported by the patterns as well as the mechanisms that would enable their implementation within each level. The architectural requirements for implementing holonic enterprises as multi-agent systems are determined by the balancing the driving forces that characterize the particularities of the patterns at each level.

4. PARTICULARITIES AT THE INTER-ENTERPRISE LEVEL

4.1 Forces To Be Balanced

Here the driving forces are triggered by the *objectives* of any relationship-based enterprise [33], as in Figure 9:

- Cost minimization - achieved via: maximum synergy (obtained by clustering the 'best' partners). Efficiency is obtained by openness to continuously sense the market's pulse and rapid (re)configuration to respond quickly to changes, as well as by the ability to respond to errors in a timely fashion. This in turn triggers new objectives:
- On demand order tracking, on-line order error reporting, ability to quickly replace a collaborative partner if it does not fulfills its commitments.
- Competitiveness on the global market. The collaborative cluster can achieve competitiveness only through continuous optimization of the collaborative cluster with maximum synergy as criteria. If a partner does not perform according to expectations (e.g. does not honour commitments, does not deliver on time, does not bid strongly enough to compete with its outside competitors) it will be replaced with a more suitable partner. This decision and appropriate negotiation will be performed by the mediator.

The driving forces are:

- Need for optimal clustering (i.e. always group the best partners) – requires on-line reconfiguration of the collaborative cluster to respond to changes in market demands as well as to the needs for maintaining optimal configuration.
- Need to balance the autonomy of each individual partner with the cooperative demands of the collaborative cluster – through negotiation that can range from simple bidding (proposal and counter-proposal) to complex argumentation and persuasion strategies. An example of the latest: the cluster sets a deadline and requirements to coordinate among the partners while partners need to argue their position and integrate the deadline with their other priorities). The cluster sets the 'rules of the game' through component protocols [43]. Preferences can be captured via a utility function such that clustering best partners can be achieved via cost minimization (e.g. via fuzzy entropy minimization [41]).

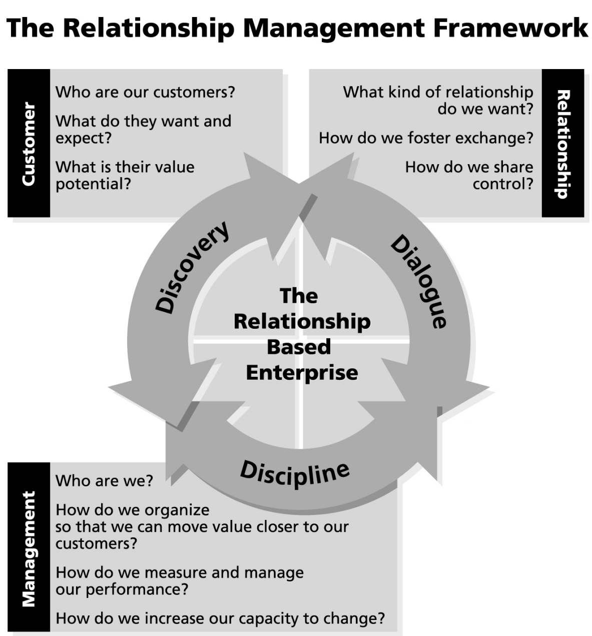


Figure 9 – Inter-Enterprise Level Relationships Management Framework
(from http://www.dmr.com/sitecontent/frameset/new_book.htm - with DMR permission)

4.2 Patterns Particularities and Required Services

The need to balance these forces leads to the following pattern particularities that in turn demand for specific services :

- **Metamorphic Architecture Pattern**. A main requirement to implement the pattern at this level is to ensure inter-enterprise/inter-node/inter-platform communication among the participants in the collaboration. FIPA [13] has already developed strong services that enable each enterprise to become a node in a collaborative network of "Agentcities" [1]. Each enterprise shall implement its software to run on FIPA compliant agent platforms, such that Agents on different platforms will be able to communicate with each other and access each other's services to create new value added services for the collaborative cluster. An extremely important issue related to inter-platform accessibility at this level is Security standards that would enable a fair balance of the autonomy and cooperative forces by enabling enough access to the collaborative cluster's entities to each-other services while keeping secrets safe.
- **Partial Cloning Pattern**. The main attributes that each enterprise has to abstract into agents at this level are: providing goods and services with which it can enter the collaborative demand-supply game; and, marketing strategies [21], that is those related attributes and functions that enable a company to penetrate into an existing cluster and also to be chosen when a new cluster is formed.
- **Mediator Agent Pattern**. The decision-making particulars in this pattern are strongly determined by the abstractions made in the partial cloning pattern as well as by the implementation mechanisms of the task decomposition-distribution pattern. The main driver of the "inside-out" enterprise-to-cluster negotiation is obtaining the trust of the mediator in charge with the coordination of the collaborative cluster. In implementing a flexible utility function for the "outside-in" cluster-to-enterprise decisions, factors to consider include how much the cluster needs the services provided by the particular enterprise under evaluation, and whether the cost of keeping this partner is worthwhile, or whether he is better replaced. An interesting method to decide between selecting or keeping a partner is suggested by [12]. They use fuzzy similarity to select the partner whose proposal is most similar to opponent's last offer and whose trust degree is higher.
- **Virtual Clustering Pattern**. To form and always keep a "best" cluster, the mediator needs Grouping Policies [9] such as obligation [29], constraining and authorization that also enforce the security requirements on each partner [26], and that enable nested management structures. Contractual frameworks that enable nested management structures [27] are essential clustering mechanisms dealing with autonomy in policy-restraining contexts and under security constraints. This is resonant with the concept of Cooperation Domain [42] introduced by Dr. James Christensen [16] in connection with the holonic control concepts developed by the Holonic Manufacturing Systems Consortium [17].
- **Task Distribution-Decomposition Pattern**. At this level, of critical importance are the compliance mechanisms (such as "reputation" and "regimentation") that can be enforced by the mediators upon the partners to coerce them in fulfilling their obligations when they assume responsibility for the assigned task [36]. Complex normative concepts enable interactive contractual design based on control mechanisms such as influence as a negotiation framework which configures the collaborative cluster.

4.3 Mechanisms Needed To Implement The Required Services

Several industry leaders run to win the race for providing *the* infrastructure for global inter-enterprise collaboration. Among them, Sun Microsystems [18] stands out through their palette of web-Centric services built on a robust, scalable infrastructure platform, which enables integration of the best components available to enhance the e-services as the e-market evolves (see Figure 10).

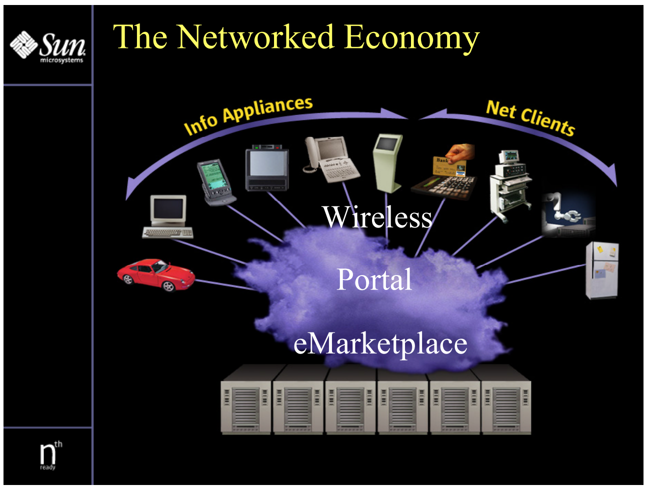


Figure 10 – Sun's Area of Influence Covers All Aspects of the Global Market

Sun's Portals are the gateway through which partners can join consortia of holonic enterprises and play competitively in the global marketplace (see Figure 11).

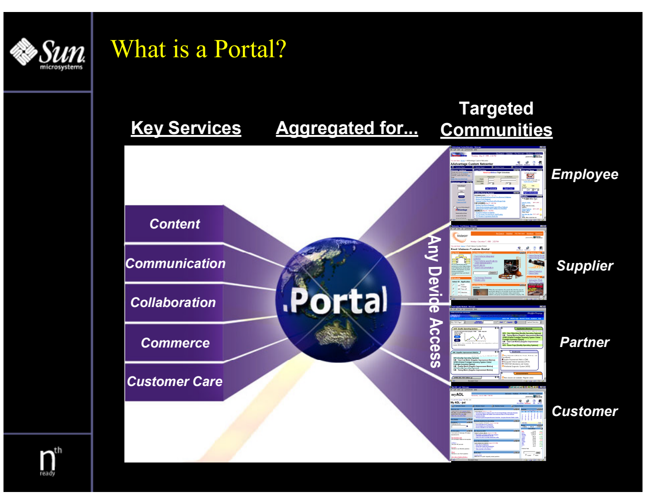


Figure 11 – Portals as Gateways to Access the Holonic Enterprise Model

On these basic services each enterprise can build with the FIPA standards to add the services and policies needed for holonic collaboration. Extended services at the inter-enterprise level include:

- Wireless access to information for e.g. on-line order tracking and error reporting on Manager's cell-phone screen, either on demand or as proactive notification by the system;
- On-line banking and financial services among the collaborative partners in the cluster; and,
- Coordination mechanisms such as order 'ready' reporting to synchronize with the work done by the other collaborative partners.

5. INTRA-ENTERPRISE LEVEL

The same patterns of holonic collaboration work to build the functionality inside each enterprise in the collaborative cluster. At this level the collaborative partners are the sections and departments within the enterprise among which the overall task for the enterprise has to be distributed and scheduled. A glimpse from the collaborative dance of the partnering enterprises is illustrated in Figure 12, from a manufacturing supply-chain perspective, that focuses the received orders down to the manufacturing resources which make the ordered products for delivery to the customer.

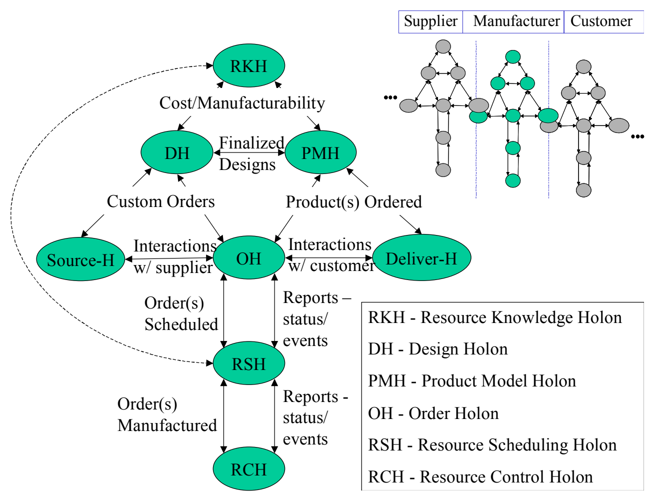


Figure 12 – Zooming Inside one Enterprise in the Collaborative Chain (for a Manufacturing Supply Chain Example)

Internal to the enterprise, as at the inter-enterprise level, all objectives arise from the same main force, namely the need to be competitive. As such, the primary objective at this level is to provide any necessary support for this required inter-enterprise competitiveness, specifically in terms of cost, time, quality, and service. The inter-relatedness of enterprise activities are increasingly being recognized, both inter- and intra-enterprise. Additionally, it is desirable to optimize the use of resources, include the ability of the system to improve over time (i.e.

learning). Of course, it is necessary to maintain both the robustness and the security of the system at all times.

5.1 Forces To Be balanced

Once the tasks have been deployed to each collaborative partner each enterprise has to organize its own resources to fulfill its commitments. The driving forces at this level are related to optimization of resources and rapid reconfiguration of schedules with adaptation of schedules to accommodate new orders in a timely manner.

The balancing forces at this level are therefore:

- Need to keep one's position within the collaborative cluster, by respecting deadlines. Here, planning and scheduling—with ability to reconfigure both plans and schedules—are required. Also applicable are enforce trust-mechanisms (as described in section 4), and the ability to accommodate new orders on-the-fly and prioritize the work flow via reconfigurable scheduling.
- Need to keep autonomy, and to stay competitive in the market. Security policies; Advertising; Bidding; Interfacing with the cooperative cluster (what do we abstract for each cluster to enforce cooperation and keep position privileged within the cluster while keeping open for new deals and competitive on the market such that I will be preferred when new clusters look for services that I provide?); etc. How does one define 'look-out' ontologies to interact with the cooperative clusters that are 'out there' such as to be able to make the best choices from the offers available, given the limited resources one has and the commitments already made to partners who one does not want to lose.

5.2 Required Services To Balance The Forces

The main services needed to balance the forces governing the intra-enterprise level are built on the same patterns of holonic collaboration described in Section 3. To stay competitive, both in the collaborative cluster and outside it, each enterprise shall be supported by the following services:

- online ordering and negotiation, custom ordering, order tracking, and error reports;
- online and flexible production (re)configuration, and fault recovery;
- online and dynamic scheduling of orders; and,
- concurrent engineering processes (product design, production, production design/planning, etc.).

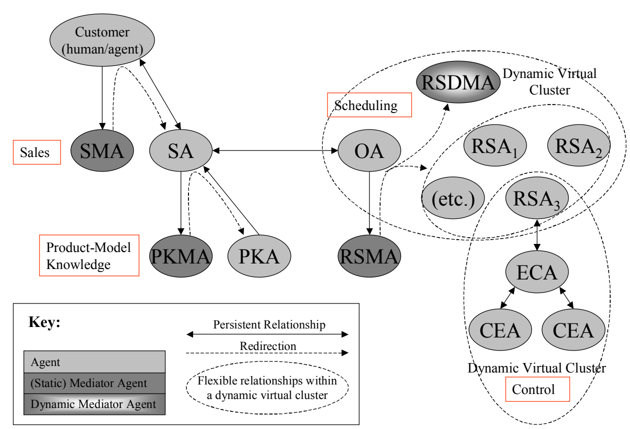
From an infrastructure perspective, the services needed to sustain the holarchy inside each enterprise are:

- registration of agents in the system
- resources for finding information (including other agents) required for agent activities; and,
- message routing, security management, error handling, and other such mundane tasks.

These capabilities can be implemented by the designer of the HMAS from scratch and/or by using an appropriate MAS platform tool. A downside to using a (current) MAS platform is the probable lack of a holonic organization to the agent platform system, but the ease of implementation with the MAS platform likely outweighs this downside if the system designer makes every attempt to conform to the holonic ideal.

The patterns of holonic collaboration that support these required services inside each enterprise are:

- **Mediator Agent Pattern**. This pattern supports at this level channelling all access to the system, to ensure security and robustness of the collaborative ecosystem inside the enterprise. As described in section 2, the holonic analysis process will consider the fractal nature of communications within a holarchy, and thus will ideally channel all access to a holon (i.e. communication arising external to a holon) through mediators and brokers (agents which act on behalf of a holon) as illustrated in Figure 13. These brokers and mediators are chosen in the initial holonic and multi-agent analysis and design processes. The mediators come in two flavours, static and dynamic. Static mediators represent and moderate over static holons, which exist throughout the life of the holonic system, while dynamic mediators represent and moderate over dynamic holons (i.e. dynamic virtual clusters [31]), which are groups of agents that are dynamically clustered and dissolved as needed to perform necessary tasks. An example of such a task is the dispatching, scheduling, and processing of an order on the resources of a manufacturing job shop, as shown in Figure 13.



SMA – Sales Mediator Agent
SA – Sales Agent
PKMA – Product Knowledge Mediator Agent
PKA – Product Knowledge Agent
RSA – Resource Scheduling Agent
OA – Order Agent
ECA – Execution Control Agent
CEA – Control Execution Agent
RSDMA – Resource Scheduling Dynamic Mediator Agent

Figure 13 – Dispatching, Scheduling, & Processing/Control by Agents within the Holonic Enterprise

- **Dynamic Virtual Clustering Pattern**. This pattern [31] accomplishes at this level the dynamic distribution of tasks by clustering Sections and Departments within the enterprise in collaborative clusters that can accomplish the subtasks, as shown in Figures 7 and 13. It also enables dynamic scheduling and reconfiguration of production for quick response to market demand as well as for fault recovery. Any sort of automated matching—between product orders and the resources on which they are to be processed—requires a common ground for comparison. One possibility, used frequently in manufacturing, is to use Group Technology to provide standardized descriptions for product features and resource capabilities [3], which can allow the determination of product families and resource work-cells/lines and illuminate overlapping resource capabilities for flexible routing of orders. The automated decomposition of products/orders into their constituent features and processes is then possible. This is useful for automated order routing [44], as well as for the concurrent design process [46].

- **Task Decomposition-Distribution Pattern**. This pattern enables on-line (re)scheduling of the production resources. For the manufacturing job shop example, the problem is not at all trivial. The flexibility in routing orders to resources, discussed above, combined with the stochastic order arrival (i.e. online orders) and the existence of redundant resources for order processing, make this scheduling problem quite intractable—i.e. NP-hard with no known constructive algorithms [15]. This pattern can be implemented by using a holonic approach employing evolutionary tuning of heuristic scheduling rules within a MAS automated scheduler, as presented in [44] and illustrated in Figure 14.

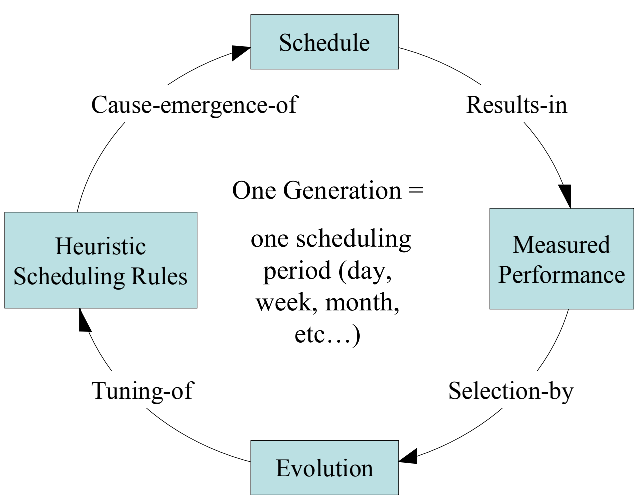


Figure 14 – Intelligent Scheduling Using Evolved Heuristic Rules

- **Partial Cloning Pattern**. At this level we need to abstract into agents those relevant characteristics of the enterprise (for ex., to support dynamic scheduling). This re-assignment of work at the logical level of the schedule leads to the reconfiguration of the work-flow across the entire organization, and ultimately to the deployment of the tasks (chains of machine operation) on the physical machines at the shop-floor control level. Clearly defined ontologies are needed to translate the reconfigured schedules into distributed control functions.

- **Ontologies Pattern**. At this level ontologies are needed to deploy the scheduled tasks down to the machine control level. For this messaging protocols have been developed and standardised by FIPA [13], and there are already many (interoperable) agent platform implementations to choose from to facilitate the implementation of the MAS, such as "FIPA-OS" [14]. The use of peer-to-peer ontologies, and the required protocols for messaging, will be supported/implemented by the MAS platform and shall not be discussed further here. However of high concern is the Schedule-to-machine ontology, which is being developed by the Product Design and Manufacturing Group within FIPA (under the lead of James Christensen.)

5.3 Mechanisms Needed To Implement The Services

The objectives and services, as outlined, point us toward the use of various mechanisms for implementing the intra-enterprise HMAS.

- Redundantly networked computers/workstations/ controllers, as the environment that the agents populate.

The hardware environment for the agents in a holonic system is limited by four main items: (1) computational speed, (2) flexibility in agent processing location (i.e. the computer), (3) network bandwidth, and (4) flexibility in routing of communications over the network. Obviously, each computer must be able to meet the computation demands placed upon it by its populating agents, though some load balancing between computers can occur. A systems crash due to hardware, operating system, or platform problems will remove agents from the system and possibly a network communication node. When considering system robustness in this case, bypassing the failure has an obvious appeal from both computational and networking redundancy/flexibility points-of-views.

- A multi-agent system platform upon which the agents can run (FIPA-compliant).

Emerging technologies generally benefit from standardisation, and holonic and multi-agent systems are no exception. The emerging holonic standards are generally being applied at the physical machine level of the holonic enterprise (see section 6), but the emerging agent standard [13] has found application at all levels of the holonic enterprise.

6. PHYSICAL MACHINE LEVEL (Holonic Control Level)

Current industrial control at the physical machine level is typically implemented using large, and often expensive, hardware platforms that support monolithic computer control applications. As a result, when the control system is installed, commissioning can take months to complete, and once the system is operational, changes are often complex and difficult. Both of these factors contribute significantly to the total cost of the industrial control project.

The holonic enterprise approach is intended to rectify these problems through the development of a distributed intelligent control solution that is inherently adaptable and dynamically reconfigurable. This system will take advantage of distributed artificial intelligence to achieve significantly shorter up-front commissioning times as well as significantly more responsiveness to change. These potential benefits, in combination with the current trend towards low-cost, distributed computing platforms, will result in a much more attractive, low-cost automation solution for future manufacturers than current centralized solutions.

Distributed intelligent control involves matching the control model more closely with the physical system. This is particularly relevant to industrial process control systems that are required to control widely distributed, heterogeneous (i.e., different hardware manufacturers) devices in an environment that is prone to disruptions. With this model, control is achieved by the emergent behaviour of many simple, autonomous and co-operative entities (i.e., agents) which "decide locally not only how to act (as subroutines do), and what actions to take (as objects do), but also when to initiate their own activity" [35].

6.1 The Main Issues And Challenges At This Level

Here the goal is to ensure continuity of production through reliability of the machines and rapid machine reconfiguration in case of production change or break-down. Although there has been a considerable amount of work on the application of Multi-Agent Systems (MAS) to soft or non-real-time systems (e.g., internet [2], supply chain management [38], manufacturing planning and scheduling [5, 8, 32, 37]), very little work has been done on applying these techniques to the lower, process control level. The main barriers at the real-time control level result from the difficulty of implementing MAS concepts in a stochastic environment where hard real-time constraints must be met to achieve safe system operation.

The primary distinction between non-real time and real time systems is that real time systems tightly link correctness with timeliness. In other words, deadlines must be met under hard real time (i.e., tasks must finish by a specified time) and soft real time (i.e., tasks must meet deadlines on average) constraints [10]. As well, real time systems are typically safety-critical systems [24] (i.e., the system should not incur too much risk to persons or equipment), and as a result, characteristics such as timeliness, responsiveness, predictability, correctness and robustness are of fundamental importance. Because of the more stringent requirements for latency, reliability and availability, it follows that the step from the non-real time or soft real time domain is a large one, requiring new models and methodologies for distributed control.

As already mentioned, the basic goal at this level is to ensure continuity of production. As a result, the holonic enterprise at the physical machine level must allow users to develop control applications and then arrange those applications for execution, allow these control applications to be reconfigured at runtime (i.e., without disrupting production), and ensure that appropriate process monitoring and fault recovery occurs. In [40] we propose a versatile diagnosis and prediction concept that can anticipate the faults while continuously improving its knowledge base and as well can keep track of the status of the other machines.

6.2 Forces To Be Balanced

The main forces from whose balance the holonic control patterns emerge are defined by the requirements on the control system:

- Need to enable the user to *develop* an application using basic and composite function blocks and application prototypes (templates) from a library.
- Need for the system to be capable of arranging for compiling of the code into low-level application code and *distributing* of this application code to appropriate resources for execution.
- Need to manage *timing and precedence relationships* while executing the distributed function blocks.
- Need for *monitoring and fault recovery*. The purpose of monitoring is to ensure that the control system performs as intended, or in other words, that no latent faults occur. When monitoring for faults, the control system should watch for failures (events occurring at specific times), and errors (inherent characteristics of the system). The types of responsibilities that our control system will have in this area are: diagnosis of program execution, monitoring for exceptions that are thrown by function block code during execution, and monitoring the system state for inconsistencies (e.g., deadline control).
- Need for *safety*. To achieve a safe system, typically two general concepts are used. First, safety channels (i.e., fault monitoring and recovery code) are separated from non-safety channels (i.e., control code). This decomposition technique is typically referred to as the "firewall concept" [24]. Second, redundancy is applied in the system in the form of homogeneous redundancy where clones or exact replicas of code are used (only protects against random failures), or in the form of diverse redundancy (see Figure 15) where different means are used to perform the same function (this protects against random and systematic failures).
- Need for *run-time reconfiguration*, if changes are required unexpectedly. This may involve simply replacing portions of the running application at the granularity level of an individual function block or, the removal of a function block and the addition of a different function block or group of function blocks (in this figure a PID function block is replaced by a FUZZY logic control function block).

6.3 Patterns of Holonic Control

We will illustrate the particularities of the Holonic Collaborative patterns at the control level on the architecture in Figure 16. The architecture that we propose enables the physical machine level to meet the basic application and fault monitoring and recovery requirements of the holonic enterprise. This is a multi-layer architecture consisting of four temporally decomposed layers: execution control (EC), control execution (EC), execution (E), and hardware (H/W). As we move down the layers shown in this figure, time scales become shorter and real-time constraints change from soft to hard real-time; as well, the degree of agency decreases (i.e., higher agents are more sophisticated but slower, while lower agents are fast and light-weight). The EC layer is concerned with "high-level" planning issues such as for reconfiguration control. The CE layer is concerned with arranging for the distribution of applications across multiple resources. The E layer is concerned with the execution of the application. The H/W layer is the physical platform, or the resource being controlled. The patterns are supported by this architecture as follows:

- **Virtual Clustering Pattern**. The Base Level (E and H/W) supports a virtual cluster of devices that are concerned with the joint execution of a task or a number of tasks; the holons within this virtual cluster (or holarchy) are composed of an information part (E agents shown in Figure 15) and a physical device (the H/W shown in Figure 15 such as a CNC or robot). An example of this pattern at the physical machine level is shown in Figure 16. In this case, it is a lot easier to split control into a number of autonomous layers that appear to interact with each other through an API where code is run asynchronously. When a task is sent down from the intra-enterprise level, everything at the physical machine level happens through task propagation (and concurrently) through cluster formation. As well, in the event of a need for fault recovery, we can track back to the point where further reconfiguration need not be considered.

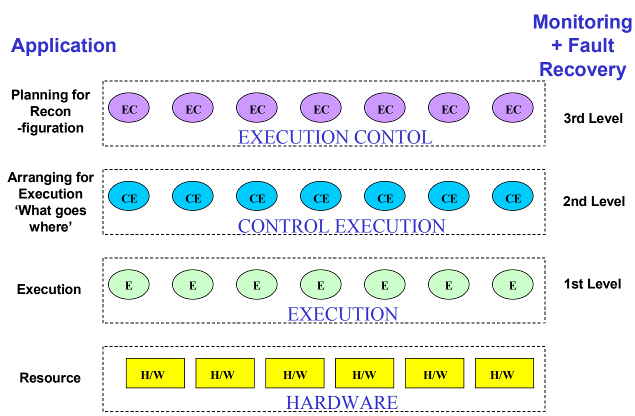


Figure 15 – Holonic Control Architecture

The sequences of actions that occur at this level are as follows. First, a task is sent down to the execution control layer from a higher layer (intra-enterprise level). Next, the first "cluster" of EC/CE agents is formed to handle the task execution. The second cluster shown in Figure 16 is formed next, and is responsible for the distribution of the control application (in this case a simple PID control application). Finally, each bit of distributed code is executed on a specific hardware platform as shown by cluster 3.

Extending this idea further, we can now think of the task as creating a "distributed holonic controller" that spans all of the agents and hardware at the EC, CE, E and H/W levels of the physical machine level that are involved in the execution of the task. This distributed holonic controller is a virtual cluster, or holarchy, formed by all of the holonic distributed control system agents involved in the task decomposition and can be thought of as a logical entity that "lives" on various resources. It is potentially a very dynamic entity since the distribution of the application across resources may change during its lifetime as a result of dynamic reconfiguration.

- **Task Decomposition-Distribution Pattern**. This pattern is implemented at the Control Execution level (CE) to distribute the execution control code to the appropriate resources.

- **Partial Cloning Pattern** enables the abstraction of relevant machine signals from the base level which will be used by the holonic controller at the CE level.

- **Mediator Pattern**. Each Execution Control and Control Execution agent plays the role of mediator when a change occurs in its area of action.

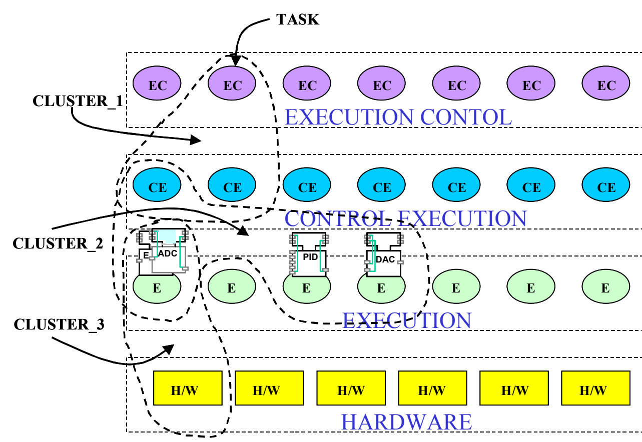


Figure 16 – Task Propagation via Layering and Clustering

6.4 Mechanisms For Holonic Control

Recently, there have been a number of advances in distributed intelligent control which provide the tools to move away from the traditional centralized, scan-based programmable logic controller architecture towards a new architecture for real time distributed intelligent control. In particular, there have been a number of advances recently in programming languages [25, 45], models for distributed control [19] and software methodologies [28, 34]. As well, there have been numerous advances in the development of intelligent field devices (e.g., sensors and actuators) that combine built-in processing capabilities with standard communication interfaces. These "Fieldbus" devices have been the focus of the IEC 1158 standard for a number of years, as well as the recent emergence of proprietary solutions such as Siemens' Profibus and Allen-Bradley's DeviceNet.

The International Electro-technical Commission (IEC) 61499 standard is one example of this new trend [19]. This standard addresses the need for modular software that can be used for distributed industrial process control. In particular, this standard builds on the function block portion of the IEC 61131-3 standard for PLC languages [25] and extends the function block (FB) language to more adequately meet the requirements of distributed control in a format that is independent of implementation. Figure

17 summarizes the basic control application requirements at this level using the IEC 61499 function block model [19].

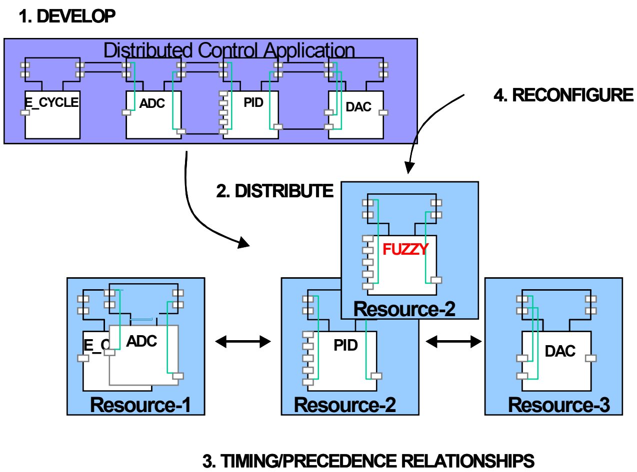


Figure 17 – Implementation of the Holonic Collaborative Patterns at the Control Level Using the Function Block Paradigm

Because of the close relationship between the proposed approach and existing object-oriented and agent-oriented approaches, the existing literature in this area is particularly relevant. One such approach, with roots in the object-oriented analysis and design methodology [4], is UML for Real Time [28]. This is an extension of the Unified Modeling Language (UML) [4] that is "specifically fine-tuned for the development of complex, event-driven, real time systems" [28], such as industrial process control systems. Although still in the early stages, methodologies such as UML for Real Time in combination with models such as IEC 61499 will provide a good starting point for the development of real time distributed intelligent control design methodologies. Through the use of these techniques, a distributed intelligent control architecture may be developed.

7. CONCLUSIONS

In response to the e-information technology revolution which has challenged all aspects of the global e-conomy, we introduced a global collaborative paradigm, the Holonic E-nterprise. Blending two recently developed e-industrial models, the holonic manufacturing systems (HMS) and multi-agent systems (MAS), it follows the 'relationship-based enterprise' paradigm shift at all levels of a cluster of collaborative organizations. Regarded as a new business model, the Holonic E-nterprise enables participants in the global market to enter strategic partnerships via the web-enabled cyber-playground by harmoniously managing the workflow throughout the nested holarchy of the collaborative information ecosystem. At the highest inter-enterprise collaborative level, the main shift is from the closed system philosophy of the traditional supply chain management to the open system philosophy governing a collaborative cluster of partners devoted to the same goal. Inside each enterprise the planning and scheduling level transfers the tasks onto the available resources—the lowest level—via dynamic, reconfigurable software technologies, in a manner which also supports monitoring and fault-recovery for order processing. At the physical machine level recent advances in distributed control system models, software and hardware are used to realize a distributed process automation system with intelligent control components.

There are already several tools that facilitate the implementation of our Holonic E-nterprise model. Those players that will enter the global e-conomy through a gateway to the Holonic E-nterprise will definitely race with a high competitive advantage. Our research at the physical machine level is intended to take advantage of recent advances in distributed control system models, software and hardware to realize a distributed process automation system with intelligent control components. The successful development of this new approach will result in an industrial control approach with significantly shorter up-front commissioning times (e.g., intelligent field devices that can be added and removed from the system seamlessly) as well as significantly more responsiveness to change (e.g., by utilizing the reactive properties of autonomous and cooperative control agents) than current PLC and SCADA systems.

8. ACKNOWLEDGEMENTS

This paper builds on the results obtained over the past 10 years by the members of the Intelligent Systems Group (ISG – http://isg.enme.ucalgary.ca) at the University of Calgary, under the leadership of Professor Douglas H. Norrie. Without his vision and continuous support, this work would not have been possible. Thank you, Doug, for being such an inspiring mentor to us all!

9. REFERENCES

[1] Agentcities website, http://www.agentcities.org/.

[2] Aparicio IV, M., Internet-scale network intelligence, *IEEE Internet Computing*, September/October, pp. 38-40, 1999.

[3] Bauer, A., Bowden, R., Browne, J., Duggan, J., and Lyons, G., Shop Floor Control Systems: From Design to Implementation, Chapman & Hall, 1994.

[4] Booch, G., *Object-Oriented Analysis and Design with Applications*, Second Edition, Addison-Wesley, 1994.

[5] Brennan, R., Performance comparison and analysis of reactive and planning-based control architectures for manufacturing, *Robotics and Computer Integrated Manufacturing 16(2-3):191-200*, 2000.

[6] Brennan, R. W., and Sorensen, C., Evaluating machine agent coupling of reactive and planning-based control architectures for manufacturing, *IEEE International Conference on Systems, Man and Cybernetics 2000* (SMC2000), Nashville, TN, Oct. 8-11, 2000.

[7] Christensen, James H., Holonic Manufacturing Systems: Initial Architecture and Standards Directions, in *Proceedings of the First European conference on Holonic Manufacturing systems*, European HMS Consortium, Hanover, Germany, 1994.

[8] Dilts, D., Boyd, N., and Whorms, H., The evolution of control architectures for automated manufacturing systems, *Journal of Manufacturing Systems 10(1):70-93*, 1991.

[9] Distributed Software Engineering (DSE) Group of the Department of Computer Science at Imperial College – Policies for Network and Distributed Systems Management, http://www-dse.doc.ic.ac.uk/Research/policies/.

[10] Douglass, B., *Doing hard time: Developing real-time systems with UML, objects, frameworks, and patterns*, Addison-Wesley, 1999.

[11] Eliens, A., *Principles of Object-Oriented Software Development*, Second Edition, Addison-Wesley, 2000.

[12] Faratin, P., Sierra, C. and Jennings, N. R., Using similarity criteria to make negotiation trade-offs, in *Proceedings of the 4th International Conference on Multi-Agent Systems*, Boston, pp. 119-126, 2000.

[13] Foundation for Intelligent Physical Agents (FIPA) website, http://www.fipa.org/.

[14] FIPA-OS website, an open-source FIPA-compliant Agent Platform, at http://fipa-os.sourceforge.net/ .

[15] French, Simon, *Sequencing & Scheduling: An Introduction to the Mathematics of the Job Shop*, Prentice Hall, 1983.

[16] HOLOBLOC website, http://www.holobloc.com/.

[17] Holonic Manufacturing Systems (HMS) Consortium website, http://hms.ifw.uni-hannover.de/.

[18] Hornberger, Reed, Adapting to Net Economy, presentation on the Energy E-Business Reality website, http://www.hourgroup.com/e-energy/reg2001.html, 2001.

[19] IEC TC65/WG6, Voting Draft: Function Blocks for Industrial Process-Measurement and Control Systems, Part 1 Architecture, International Electrotechnical Commission, 2000.

[20] Jennings, N. R., Faratin, P., Lomuscion, A. R., Parsons, S., Sierra, C., and Wooldridge, M., Automated Negotiation: Prospects, Methods and Challenges, *International Journal of Group decision and Negotiation 10(2):195-215*, 2001.

[21] Jennings, N. R., Faratin, P., Norman, T.J., O'Brien, P. and Odgers, B., Autonomous Agents for Business Process Management, *International Journal of Applied Artificial Intelligence 14(2):145-190*, 2000.

[22] Jones, A. J. I., and Sergot, M. J., A Formal Characterisation of Institutionalised Power, *Journal of the IGPL 4(3):429-445*, June, 1996.

[23] Koestler, A., *The Ghost In The Machine*, Arkana, 1967.

[24] Leveson, N., *Safeware*, Addison-Wesley, 1995.

[25] Lewis, R., *Programming Industrial Control Systems Using IEC 1131-3*, IEE, 1996.

[26] Lupu, E., Sloman, M., Dulay, N., Damianou, N., Ponder: Realising Enterprise Viewpoint Concepts, *4th International Enterprise Distributed Object Computing Conference* (EDOC2000), Makuhari, Japan, Sept. 25-28, pp. 66-75, 2000.

[27] Lupu, E., and Sloman, M., Conflicts in Policy-based Distributed Systems Management, *IEEE Transactions on Software Engineering 25(6):852-869*, November/December, 1999.

[28] Lyons, A., UML for real-time overview, Technical Report of ObjecTime Ltd, 1998.

[29] Marriott, D., Sloman, M., Implementation of a Management Agent for Interpreting Obligation Policy, *IEEE/IFIP Workshop on Distributed Systems Operations and Management* (DSOM '96), Laquila, Italy, October, 1996.

[30] Martin, J. and Odell, J. J. *Object-Oriented Methods*, Second Edition, Prentice Hall, 1998.

[31] Maturana, F. and Norrie, D. H., Multi-agent Mediator Architecture for Distributed Manufacturing, *Journal of Intelligent Manufacturing* 7, 257-270, 1996.

[32] McFarlane, D. and Bussmann, S., Developments in holonic production planning and control, *International Journal of Production Planning and Control 11(6):522-536*, 2000.

[33] McKenzie, R., *The Relationship-Based Enterprise*, McGraw-Hill, 2001.

[34] Odell, J., Parunak, H., and Bauer, B., Extending UML for agents, ERIM Center for Electronic Commerce, http://www.erim.org/~vparunak, 2000.

[35] Parunak, H., Autonomous agent architectures: a non-technical introduction, Industrial Technology Institute Report, 1993.

[36] Sergot, M. J., and Richards, F.C.M., On the Representation of Action and Agency in the Theory of Normative Positions, in *Proceedings of the Fifth International Workshop on Deontic Logic in Computer Science* (DEON'00), Toulouse, France, January, 2000.

[37] Shen, W., and Norrie, D. H., Agent-based systems for intelligent manufacturing: a state-of-the-art survey, *Knowledge and Information Systems 1*, pp. 129-156, 1999.

[38] Shen, W., Ulieru, M., Norrie, D. H., and Kremer, R., Implementing the internet enabled supply chain through a collaborative agent system, *Agents'99 Workshop on Agent Based Management of the Internet-Enabled Supply-Chain*, pp. 55-62, 1999.

[39] Shu, S., and Norrie, D. H., Patterns for Adaptive Multi-Agent Systems in Intelligent Manufacturing, in *Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems* (IMS'99), Leuven, Belgium, September 22-24, pp. 67-74, 1999.

[40] Ulieru, M., Web-Centric Diagnostic and Prediction System for Global manufacturing, *Proceedings of IFSA-NAFIPS 2001*, Vancouver, BC, July 24-29, 2001.

[41] Ulieru, M., and Norrie, D. H., Fault Recovery in Distributed Manufacturing Systems by Emergent Holonic Re-Configuration: A Fuzzy Multi-Agent Modeling Approach, *Information Science*, September 2000.

[42] Ulieru, M., Norrie, D. H., Kremer, R., and Shen, W., A Multi-Resolution Collaborative Architecture for web-Centric Global Manufacturing, Information Science, Vol. 127, August, 2000.

[43] Vulkan, N., and Jennings, N. R., Efficient mechanisms for the supply of services in multi-agent environments, *International Journal of Decision Support Systems 28(1-2):5-19*, 2001.

[44] Walker, Scott S., Brennan, Robert W., and Norrie, Douglas H., Demonstrating Emergent Intelligence: An Evolutionary Multi-Agent System for Job Shop Scheduling, in *Proceedings of the Genetic and Evolutionary Computation Conference* (GECCO), San Francisco, July, 2001. (in press)

[45] Wang, L., Brennan, R. W., Balasubramanian, S., and Norrie, D. H., Realizing holonic control with function blocks, *Integrated Computer-Aided Engineering 8(1)*, January, 2001.

[46] Xue, D., Modeling of Product Life-cycle Knowledge and Data for an Intelligent Concurrent Design Systems, Chapter 6 in *Knowledge Intensive Computer Aided Design*, Finger, S., Tomiyama, T., and Mantyla, M. (eds.), Kluwer Academic Publishers, pp. 117-142, 1999.

[47] Zhang, H., and Norrie, D. H., Holonic Control at the Production and Controller Levels, *Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems*, Leuven, Belgium, 215-224, 1999.