Presented at
First European Conference on Holonic Manufacturing Systems, Hannover, Germany, 1 December 1994.

HOLONIC MANUFACTURING SYSTEMS:
INITIAL ARCHITECTURE AND STANDARDS DIRECTIONS

James H. Christensen, Ph.D.
Senior Principal Engineer
Rockwell Automation/Allen-Bradley
Highland Heights, Ohio 44143

ABSTRACT

In February 1992, the governments of the USA, Japan, the European Community (EC), the European Free Trade Association (EFTA), Australia and Canada initiated a two-year feasibility study on an international program for Intelligent Manufacturing Systems (IMS) research. Six international consortia won approval to conduct one-year Test Cases as part of this Feasibility Study, concluding in March 1994. One of the consortia, with 31 international industry, academic and research institute partners, performed an extensive preliminary study in Holonic Manufacturing Systems (HMS) composed of *holons* - intelligent, autonomous, cooperative agents. This paper presents an initial architecture, systems engineering methodology, and standardization directions for twenty-first century manufacturing systems, based on the results of the HMS Test Case.

Introduction

Teams of industry experts, scientists, and engineers from the world's leading industrial nations have been working together over the past two years to build and test a framework for international collaboration in Intelligent Manufacturing Systems (IMS). The experiences of teams, coming together from Australia, Canada, Europe, Japan and the USA to work for one year on collaborative "test case" projects, formed part of a two year feasibility study that began in February 1992. This feasibility study proved that this kind of international collaboration could achieve significant results in a relatively short time.

Holonic Manufacturing Systems (HMS) was one of the six test cases. The HMS Consortium consisted of 31 Partners from all regions in the IMS program, comprising 15 industrial partners, 10 universities and 6 research institutes. Consortium coordinating partners were BHP Co. Ltd. in Australia, Queen's University in Canada, Softing GmbH in Europe, Hitachi Ltd. in Japan, and Rockwell Automation/Allen-Bradley in the USA.

Arthur Koestler developed the basic concepts of holonic systems in his groundbreaking book *The Ghost in the Machine*[1]. Koestler postulated a set of underlying principles to explain the self-organizing tendencies of social and biological systems. He proposed the term *holon* to describe the building blocks of these systems. This term is a combination of the Greek word *holos*, meaning "whole", with the suffix *-on* meaning "part", , as in prot*on* or neur*on*. This term reflects the tendencies of holons to act as autonomous individuals, yet cooperating to form apparently self-organizing hierarchies of subsystems, such as the cell/tissue/organ/system hierarchy in biology. Koestler used the term *holarchy* to describe these holonic hierarchies.

HMS Concepts

Prior to the organization of the HMS Consortium, preliminary work had been done in Japan on the adaptation of Koestler's concepts to manufacturing systems.[2],[3] The HMS Consortium built on this work to define the following holonic concepts as applied to manufacturing systems:

> **holon**: An autonomous and cooperative building block of a manufacturing system for transforming, transporting, storing and/or validating information and physical objects. The holon consists of an information processing part and often a physical processing part. A holon can form part of another holon.
>
> **autonomy**: The capability of an entity to create and control the execution of its own plans and/or strategies.
>
> **cooperation**: A process whereby a set of entities develop mutually acceptable plans and execute them.
>
> **holarchy**: A system of holons which can cooperate to achieve a goal or objective. The holarchy defines the basic rules for cooperation of the holons and thereby limits their autonomy.

> **holonic manufacturing system (HMS)**: A holarchy which integrates the entire range of manufacturing activities from order booking through design, production and marketing to realize the agile manufacturing enterprise.

Figure 1 illustrates the differences between holonic systems and other architectures such as master-slave and client-server systems.

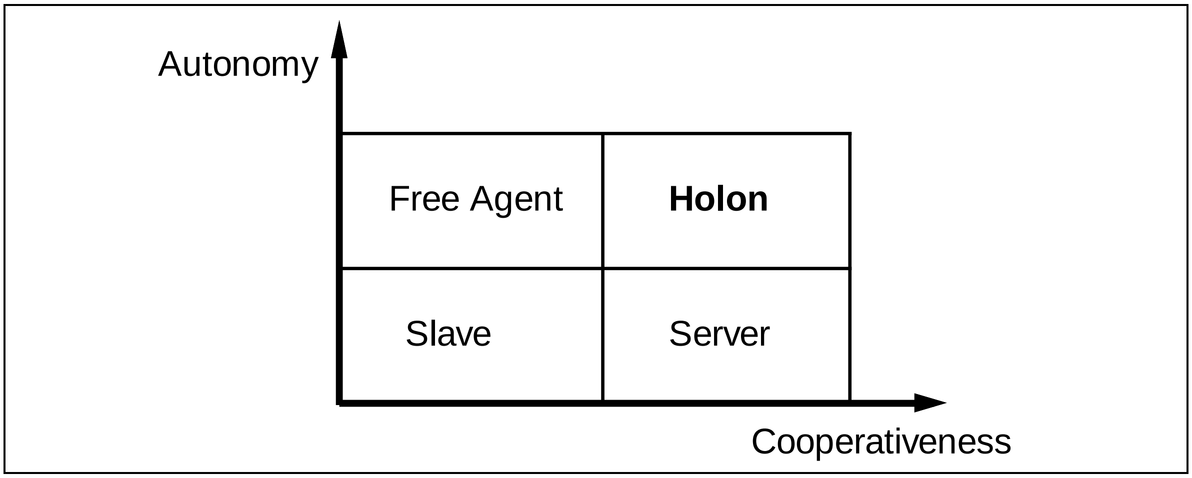

Figure 1 - Holons vs. traditional system elements[4]

Agility: Critical factors and architectural requirements

The principal characteristics of the successful "agile" manufacturing enterprise of the 21st century and its associated manufacturing systems have been identified as:

> The **agile manufacturing enterprise** is able to bring out totally new products quickly. It assimilates field experience and technological innovations easily, continually modifying its product offerings to incorporate them. Its products are designed to evolve. As the needs of users change, as improvements are introduced, users can readily reconfigure or upgrade what they have bought instead of replacing it. Reprogrammable, reconfigurable, continuously changeable production systems, integrated into a new, information intensive, manufacturing system, make the lot size of an order irrelevant. The cost of production is the same for 10,000 units of one model, as for one unit each of 10,000 different configurations of all the models of a single product. Agile manufacturing thus produces to order.... Similarly, quality in agile manufacturing advances from being measured in defects per part when sold, to customer gratification over the full life of the product.[5]

The HMS consortium has identified a number of critical factors in existing manufacturing systems which limit the achievement of this vision[6]. A selection of these critical factors of most relevance to system architecture is listed in Table 1.

TABLE 1
Critical factors for system architecture

- **Disturbance handling**: Provide better and faster recognition of and response to machine malfunctions, rush orders, unpredictable process yields, human errors, etc.
- **Human integration**: Support better and more extensive use of human intelligence.
- **Availability**: Provide higher reliability and maintainability despite system size and complexity.
- **Flexibility**: Support continuously changing product designs, product mixes, and small lot sizes.
- **Robustness**: Maintain system operability in the face of large and small malfunctions.

Table 2 presents a list of key architectural requirements which can be derived from the need to overcome these critical factors.

TABLE 2
Key architectural requirements

1. **Disturbance handling, availability, robustness**

   Provide intelligent system elements for self- and cooperative planning, scheduling, fault recognition, diagnosis, and repair.

2. **Human integration**
   - Provide more intuitive, flexible, responsive, user-customizable human interfaces.
   - Provide “intelligent assistants” to augment human intelligence and prevent human error.

3. **Flexibility**

- Provide greater human control over system configuration and functionality.
- Provide self-reconfiguration (“metamorphic”) capabilities.
- Support continuous/incremental changes in roles and relationships of system elements (“fluidity”).

Figure 2 illustrates the sort of architectural evolution that may have to be supported in accordance with the last requirement of Table 2. In this figure, circles represent executing elements, squares represent controlling elements, and lines represent communication among elements. In the final stage of evolution, each element (holon) is both self-controlling and self-executing (autonomous), while cooperating with other elements via communication and negotiation. In addition, each holon may itself be a *holarchy* composed of other holons. At this final stage, the architecture is composed of *self-similar* elements; that is, it becomes a *fractal* architecture[7].

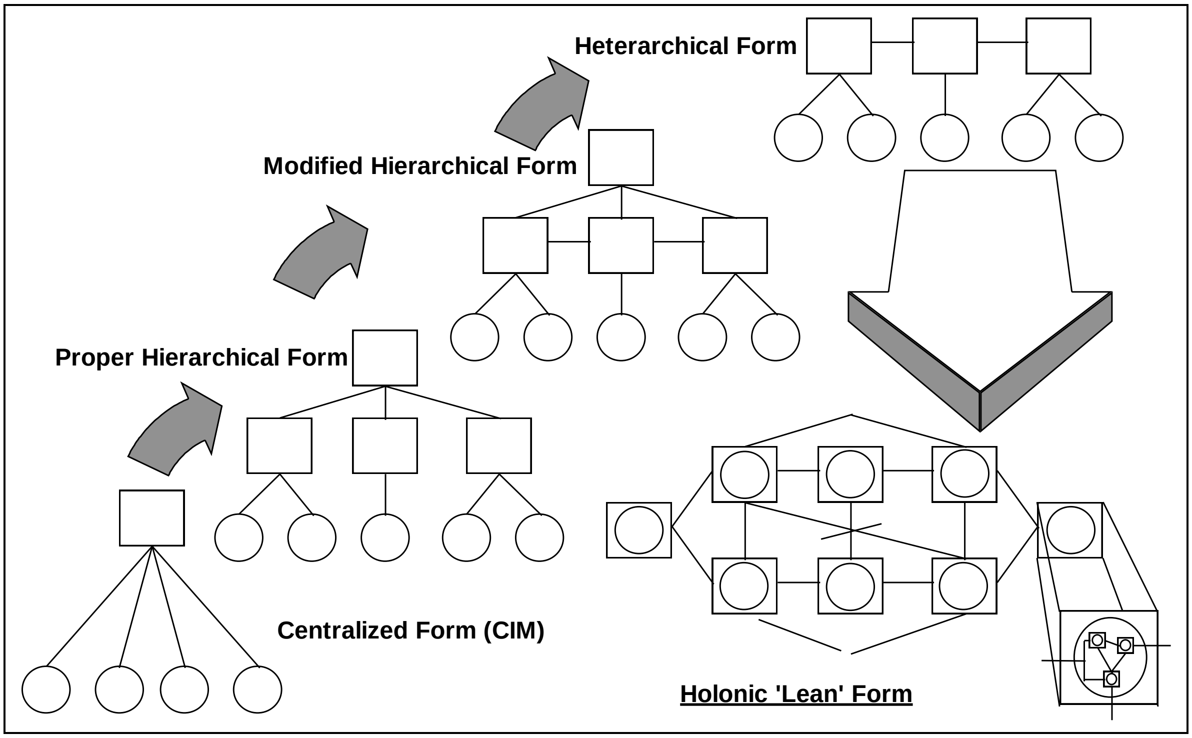

Figure 2 - Architectural evolution[8]

This type of architectural evolution is best implemented by a continuous improvement *(kaizen)* process as illustrated in Figure 3. With respect to system implementation, this process has been described as follows:

> System development is a change management process. The normal case is a change from 'something' to 'something else'. The first development cycle is a special case: a change from 'nothing' to 'something'[9].

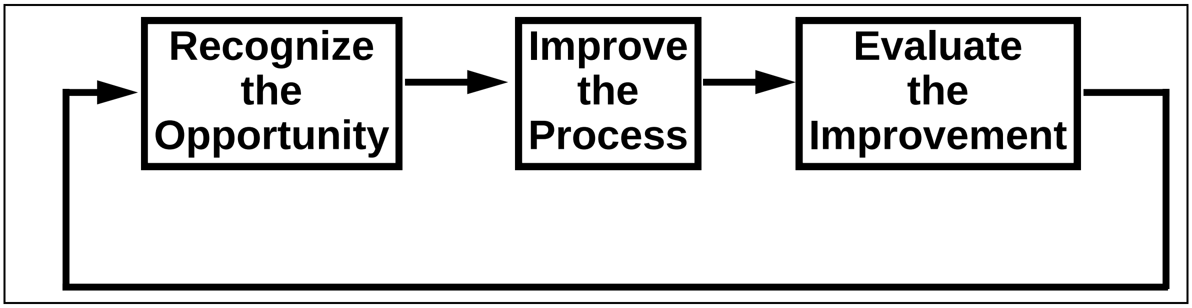

Figure 3 - Continuous Improvement *(kaizen)*

Architectural models

Figure 4 illustrates a possible architecture for a holon which can meet the requirements expressed above . This figure combines a conceptual drawing from prior Japanese work[3] with the Generic Activity Model ("GAM") developed by ISO TC184/SC5.[10]

Figure 4 shows that a holon may have *interfaces* through which it exchanges *information*, *material*, or *resources* (which themselves may be holons) with other holons or the environment. In this conception, the human can also be regarded as a *resource* which may enter or exit the holon in the same manner as other resources, such as tools, cutting fluid, etc.

Note that information may flow through any of the internal components of the holon, especially if the "processing system" is an *information* processing system. For instance, if the information processing system is a data base management system, the "intelligent control system" could be an expert system front end interacting with the human.

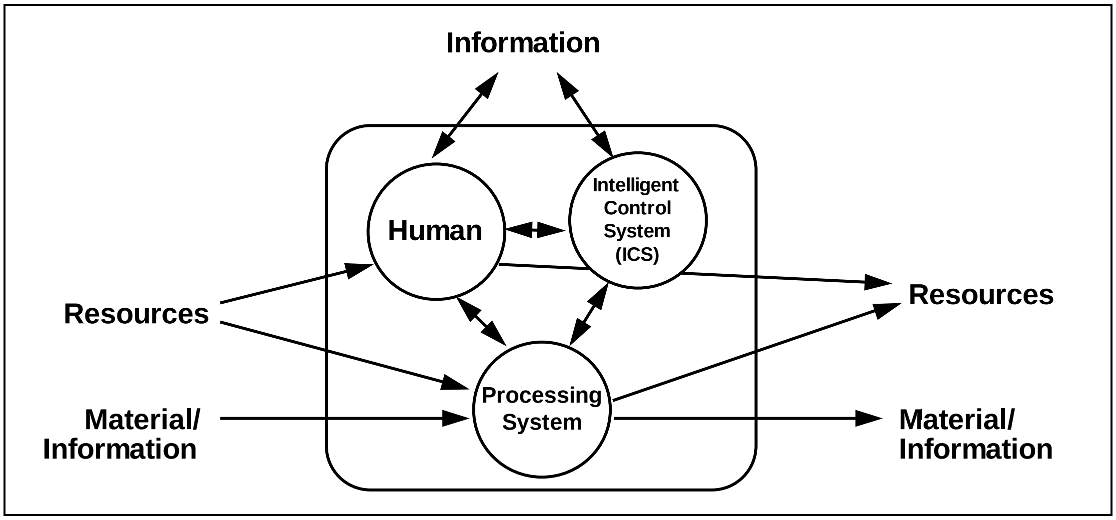

Figure 4 - Generic Activity Model of a Holon[11]

NOTE: Both the ISO GAM and the HMS Consortium definition of *holon* include *validation* of material and information objects. In future work on holonic systems, it will also be important to validate the *resources* which may flow into or out of the holon. That is, the holon needs to be sure that a resource is actually able to perform a function before attempting to use it to perform that function.

The functions of the Intelligent Control System (ICS) portion of Figure 4 can be partitioned as shown in Figure 5 into four major blocks:

- The Process/Machine Control (PMC) block, responsible for execution of the control plan for the process being controlled. This may include, in addition to traditional control algorithms, elements which add "intelligence" such as rule-based reasoning, fuzzy logic, and neural nets.
- The Process/Machine Interface (PMI) block, representing the physical and logical interface to the physical or logical process being controlled. This interface may itself include "intelligent" elements such as self-diagnosing sensors and actuators.

- The Human Interface (HI) block, representing the interfaces to the human resources which may enter or leave the holon, such as operators, supervisors, maintenance personnel, process engineers, etc. This block may also contain "intelligent" elements such as diagnostic aids, intelligent "front ends", etc.
- The Inter-Holon Interface (IHI) block, which provides for the exchange of information with other holons in the system, as well as capabilities for negotiating and cooperating with other holons to meet system goals. This may include, for example, cooperative planning and scheduling elements.

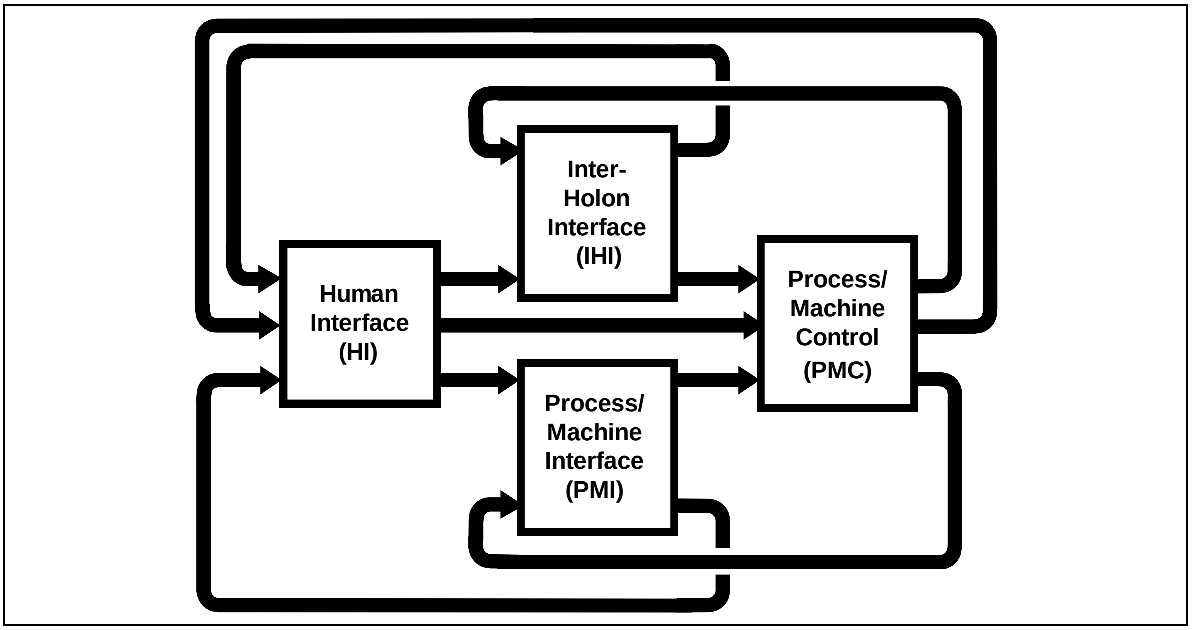

Figure 5 - Major functions of an Intelligent Control System (ICS)

Figure 6 illustrates how incremental introduction of the ICS model can be used to achieve the evolution of system architecture sketched in Figure 2, from centralized CIM to fully fractal holonic systems. At the initial stage, the "master" computer in the CIM hierarchy behaves as if it is in full control of its "slaves"; that is, it exercises its authority through a command/response interface that is functionally identical to the PMI of Figure 5. Similarly, in the centralized CIM architecture the "slave" units cooperate through an interface similar to the IHI in Figure 5; however, in this configuration their cooperation is limited to taking appropriate actions in response to commands received from the "master". Incremental modification of system architecture can then be applied in *kaizen* fashion as the opportunity arises:

i) A "proper hierarchical" system is constructed by appropriate decomposition of functions from the centralized CIM system; however, the master/slave relationship is maintained through the PMI/IHI connections.

ii) A "modified hierarchical" form is obtained by enabling some functions to be accomplished cooperatively via IHI-to-IHI connections, while maintaining some master/slave relationships via PMI/IHI connections.

iii) A "heterarchical" form is obtained with fully cooperative relationships in some parts of the system and master/slave relationships in other parts.

iv) Finally, a fully holonic form is realized when all relationships among system elements are cooperative, i.e., via IHI-to-IHI connection. The fractal nature of the system is realized by enabling each holon to belong to more than one "community" of holons, i.e., through more than one IHI.

NOTE: Figure 6 is not intended to dictate a "lockstep" sequence of phases in system evolution. The evolution toward fully fractal holonic architectures should be piecewise, as dictated by the opportunities for improvement in a *kaizen* process.

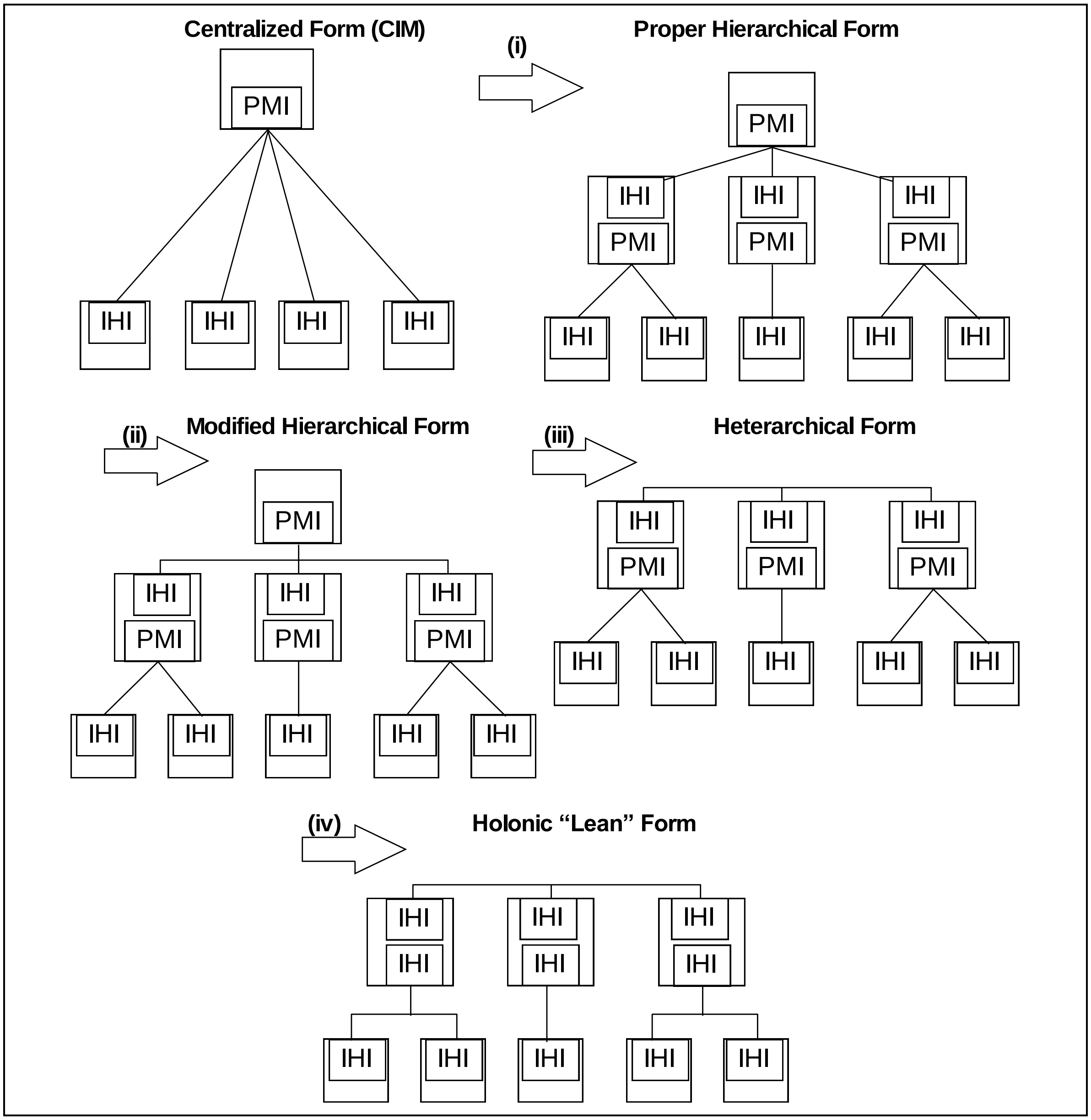

Figure 6 - Architectural evolution using the ICS model

A Systems Engineering Holon

The opportunities for applying HMS technology to new "greenfield" facilities will be relatively rare. The most profitable application for HMS technology will be in **incremental improvements** of existing processes. Therefore, any proposed HMS engineering process (and its engineering support environment) must support this **continuous improvement (*kaizen*).**

Figure 7 illustrates a system engineering process for the incremental introduction of HMS technology during the continuous improvement of manufacturing processes. The term "process" in this figure refers to the process to be improved. This will usually be the manufacturing process; however, it could (and should) also be the system engineering process itself.

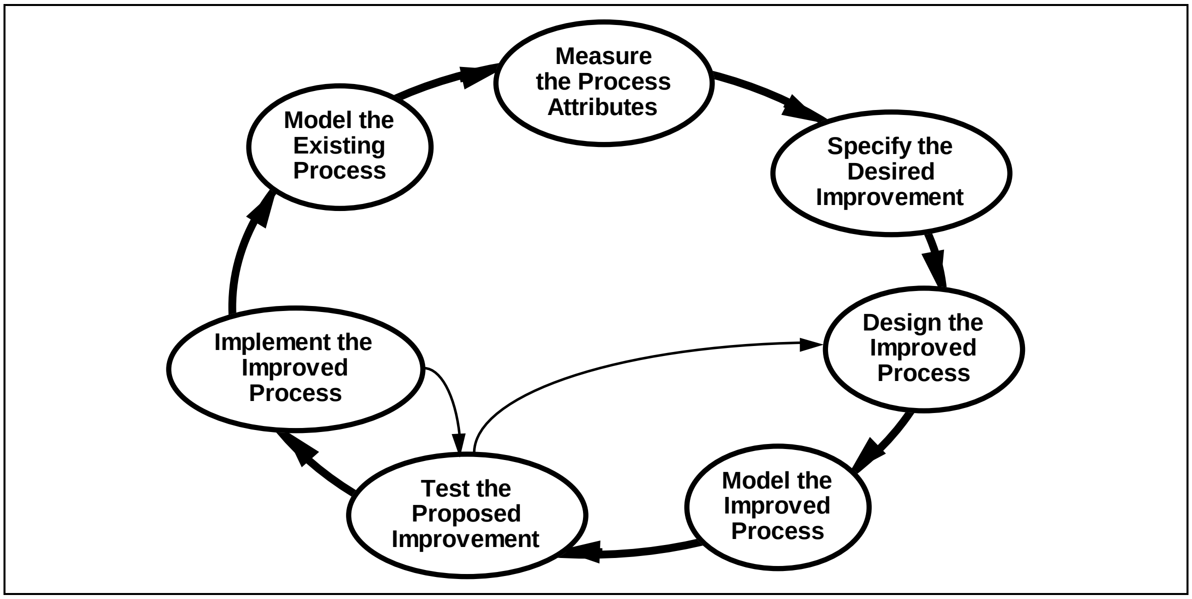

Figure 7 - An incremental HMS engineering process

Figure 8 envisions this system engineering process as the responsibility of a "system engineering holon" interacting with the manufacturing system.

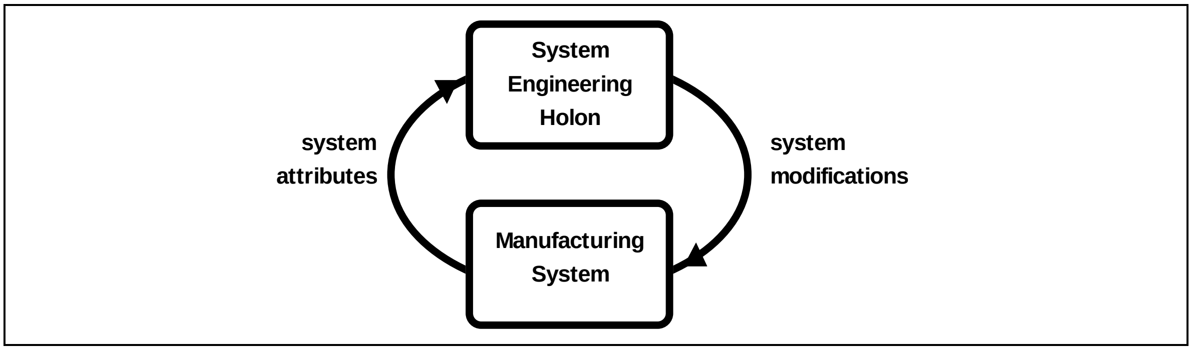

Figure 8 - HMS engineering context

Figure 9 illustrates a possible internal structure for a system engineering holon supporting an incremental HMS engineering process. Although this figure identifies the human as a "systems engineer", this could be any human with the responsibility of modifying the system to improve its quality attributes. This could include, for instance, process operators or maintenance personnel. The consequences for the human interface of the "intelligent support tools" are obvious.

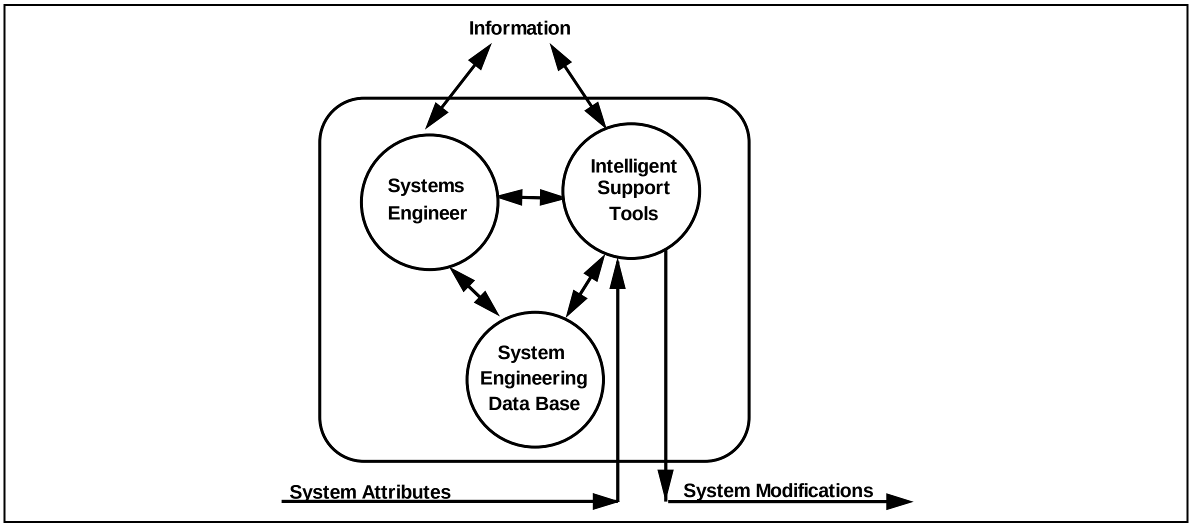

Figure 9 - A system engineering holon

The *intelligent support tools* and *system engineering data base* form the **support environment** for the system engineering process. Table 3 provides an illustrative list of the functionality required of this environment to support the incremental system engineering process shown in Figure 8.

TABLE 3
Requirements for a Holonic Systems Engineering Support Environment

System Engineering Data Base Human Interface Elements Inter-Holon Interface Elements Process/Machine Control & Interface Elements Product/Process/System Simulation Elements Product/Process/System Configuration Data Product/Process/System Historical Data
Intelligent Support Tools System Construction, Simulation, Testing, & Operation Data Access, Retrieval, & Analysis Function/Data Encapsulation & Reuse Library & Version Management

Standards Themes

Only limited implementation of holonic systems is possible at present. New *products* will be necessary to provide the functionality required for full realization of HMS technology. New or modified *standards* for the design of such products are required to assure their usefulness in holonic systems. Manufacturing and control system engineers will need *training,* as well as new, computer-assisted *tools,* in order to make effective use of the new technologies.

Figure 10 identifies the major functional elements and interfaces in the architecture of a holon designed with the HMS engineering methodology outlined above. Each of the functional elements represents an opportunity for HMS-compatible families of *products*. Each interface represents a critical point of *standardization* to assure ease of implementation, integration, and maintenance of HMSs.

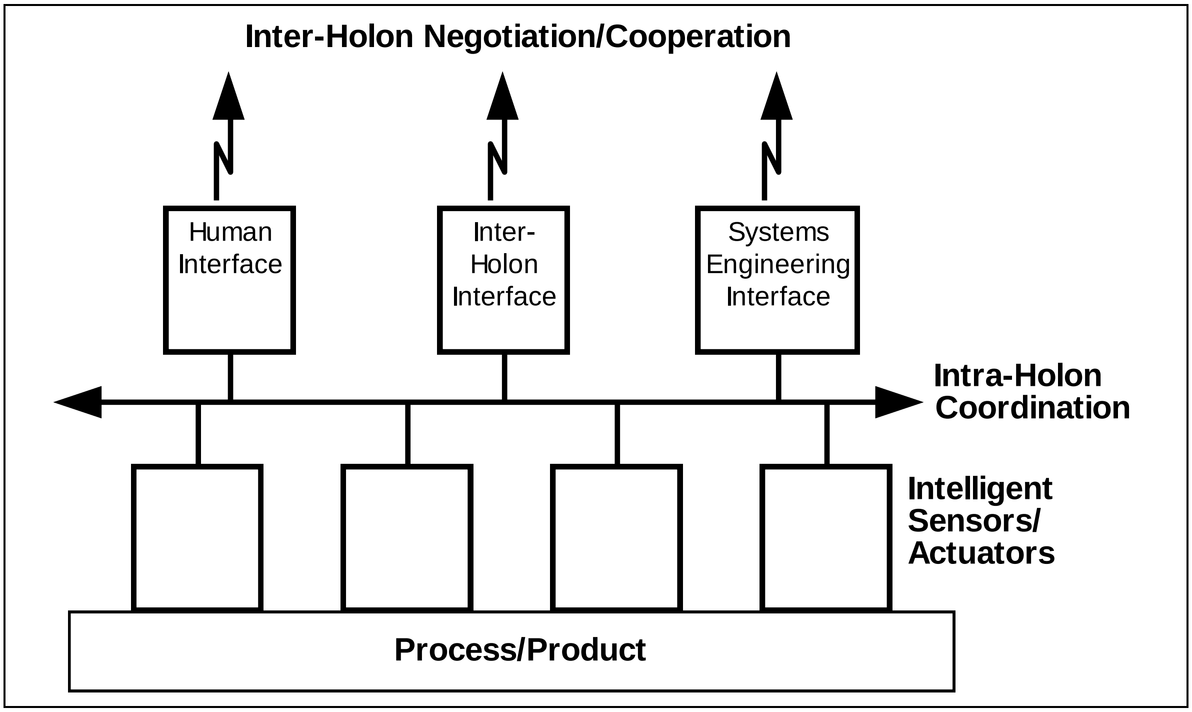

Figure 10 - Critical interfaces in HMS

The HMS Consortium has identified the following critical Standards Theme Tasks (STTs). Critical interfaces are addressed in STT2, STT3, STT4, STT6 and STT7, while STT5 addresses critical software factors. STT1 maintains an integrated picture of the standards identified in the other STTs. The work in these STTs will both identify requirements for and impose constraints on the design of HMS hardware and software products and their associated tools, training and support.

- **STT1 - International Standards Profile:** This is an ongoing task for the duration of the HMS program with responsibility to develop the framework for standardization of HMS technology, including selection of existing standards where appropriate and recommendation of new standardization work items where necessary.

- **STT2 - Inter-Holon Interface:** Develop standardized algorithms and application layer protocols for inter-holon negotiation and cooperation.
- **STT3 - Intra-Holon Interface:** Develop a full protocol stack specification, including "user layer", for intra-holon coordination.
- **STT4 - Human Interface:** Develop a standard specification for human interface building blocks and integration techniques to provide enhanced integration of human intelligence into HMSs.
- **STT5 - Intelligent System Elements:** Develop standard specifications for system building blocks and integration techniques to meet user requirements for "intelligent" behaviors for holonic system elements, including fuzzy logic, neural nets, rule-based and knowledge-based reasoning.
- **STT6 - Systems Engineering Interface:** Develop standard specifications to meet the interface requirements for the Systems Engineering Holon described above. This should include an "application protocol" for the management of the Systems Engineering Data Base, based on the international Standard for the Exchange of Product data (ISO 10303, "STEP").
- **STT7 - Mechanical Interfaces:** Develop standard interface specifications for mechanical elements of holons, such as machining, assembly and transport elements.

International Standardization Bodies

In order to provide for the global deployment and support of HMS, the relevant standards must also be international in scope. The two major bodies responsible for the development and maintenance of international standards are IEC and ISO:

- **IEC (the International Electrotechnical Commission)** was founded in 1906. It is responsible for preparing and publishing international standards for the electrical and electronics fields. The IEC is a non-governmental organization composed of national committees in 41 Countries. The work of the IEC is carried out by 88 Technical Committees and more than 100 sub-committees and several hundred working groups, each being responsible for developing standards for a well-defined sector of technology.
- **ISO (the International Standards Organization)** was formed in 1926 for the first meeting and reorganized in 1946. The ISO currently has over 150 technical committees, with more than 1,850 sub-committees and working groups. It has established approximately 7,100 International Standards. The work covers virtually every area except electrotechnical issues.

Table 4 lists current international standards which may be taken as a baseline for HMS standardization. Ongoing research in HMS should identify the options within these standards which should form part of the HMS Standard Profile, as well as necessary extensions and exceptions.

TABLE 4
Baseline HMS Standards

- **ISO TR 1000: International Standardized Profiles**
- **ISO TR 10314: Shop Floor Reference Model**
- **IEC 1131-3: Programmable Controller Languages**
  - Function Block Diagram (FBD)
  - Sequential Function Chart (SFC)
  - Structured Text (ST)
  - Encapsulation/Reuse Libraries
- **IEC/ISO 9506 (MMS)**

Table 5 lists some of the work in progress in various Technical Committees (TCs), Subcommittees (SCs), and Working Groups (WGs) of IEC and ISO which is of most relevance to HMS. The relationships between these work items, the baseline standards listed above, and the HMS Standards Theme Tasks (STTs) are discussed in detail in the HMS "Standardization Proposals" deliverable[12]. Of particular interest are the STEP and Function Block developments discussed below.

TABLE 5
Work in Progress on HMS-related Standards

IEC TC65/WG6: Function Blocks System, Device, Resource Models Distributed Application Model Event/Algorithm Execution Model
ISO TC184/SC4: STEP (ISO 10303/xxx) Product Data Exchange Exchange of Holonic System Design Data AP 212: Electrotechnical Plant
IEC SC65B/WG6: Fieldbus
ISO TC184/SC5/WG4: Programming Environments (MAPLE)

STEP (STandard for the Exchange of Product data)

STEP is a multi-part standard with the overall number ISO 10303 under development by a large number of Working Groups under Subcommittee 4 of ISO TC184. These parts include:

- A meta-language (EXPRESS) for the specification of product data description syntax
- A guideline for the development of "Application Protocols (APs)" in various domains

- APs for a number of domains, including electronic CAD and assembly, three-dimensional solid parts, sheet metal parts, etc.

Clearly, HMS will have to deal with STEP encoded data, not only for the exchange of raw geometric data about parts, but for analyzing and reasoning about structured data for the composition of assemblies from their component parts, derivation of machining tasks from feature-oriented part descriptions, etc. Future HMS systems will also use STEP encoded data to describe, implement and reason about the manufacturing system and its distributed control system themselves. Initial work in this area is in progress in the STEP AP 212 for Electrotechnical Plant.

Function Block Standardization

The Function Block standard currently in development in IEC TC65/WG6 may be particularly well adapted for use in HMS. This is projected to be a seven-part standard, and only the first part (General Requirements) has advanced to the status of a Committee Draft for Voting (CDV)[13]. Thus, it is too early to do more than indicate areas of potential applicability to HMS.

Figure 11 illustrates the model of an "industrial process measurement and control system" utilized by TC65/WG6. This may be considered to be a model of:

a) a single holon, in which the communication network(s) provide intra-holon coordination and each device is a sensor or actuator; or

b) a holonic system (which may itself be considered a holon), in which the communication network(s) provide inter-holon negotiation and cooperation, and each device is an Intelligent Control System (ICS) for a single holon.

Obviously, the choice of communication protocol will be strongly dependent on the level at which the model is applied. Also, in a fractal holarchy there will be a gradual transition from lower-level (real-time data exchange) to higher-level (more abstract negotiation and planning) communications as one moves "upward" in the holarchy. It is therefore a requirement that multiple protocols will have to be combined and interoperate harmoniously within the same network architecture.

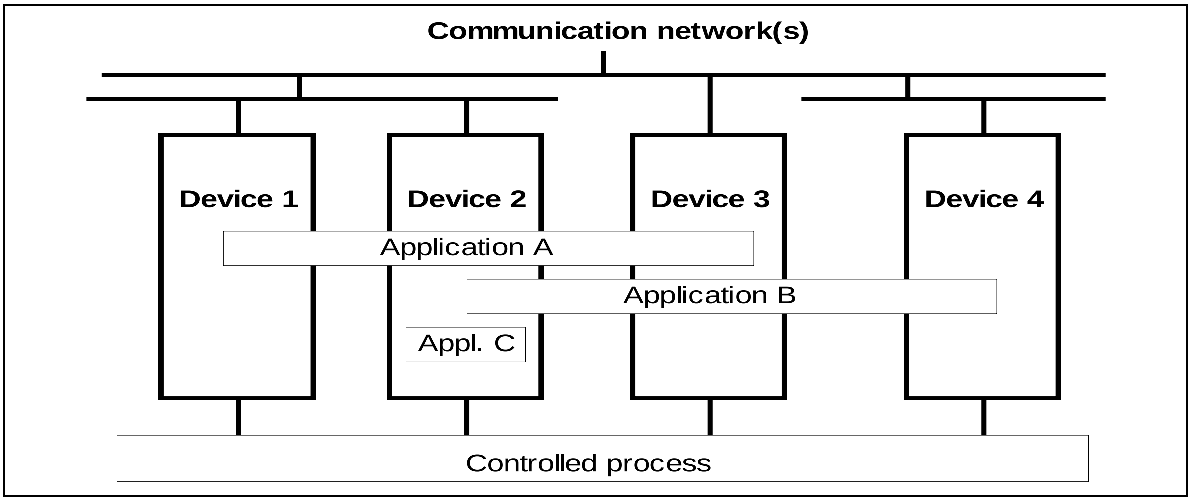

Figure 11 - Model of an industrial-process measurement and control system[14]

Figure 12 illustrates the device model used by IEC TC65/WG6. Again, the device may be viewed as a model of: (i) a holonic system, where each *resource* is an ICS associated with a single holon; or (ii) an individual holon, where each resource may provide one or more of the functions of an ICS as illustrated in Figure 5. Thus, the same requirements and constraints may apply to inter-resource communication as to inter-device communication as described above.

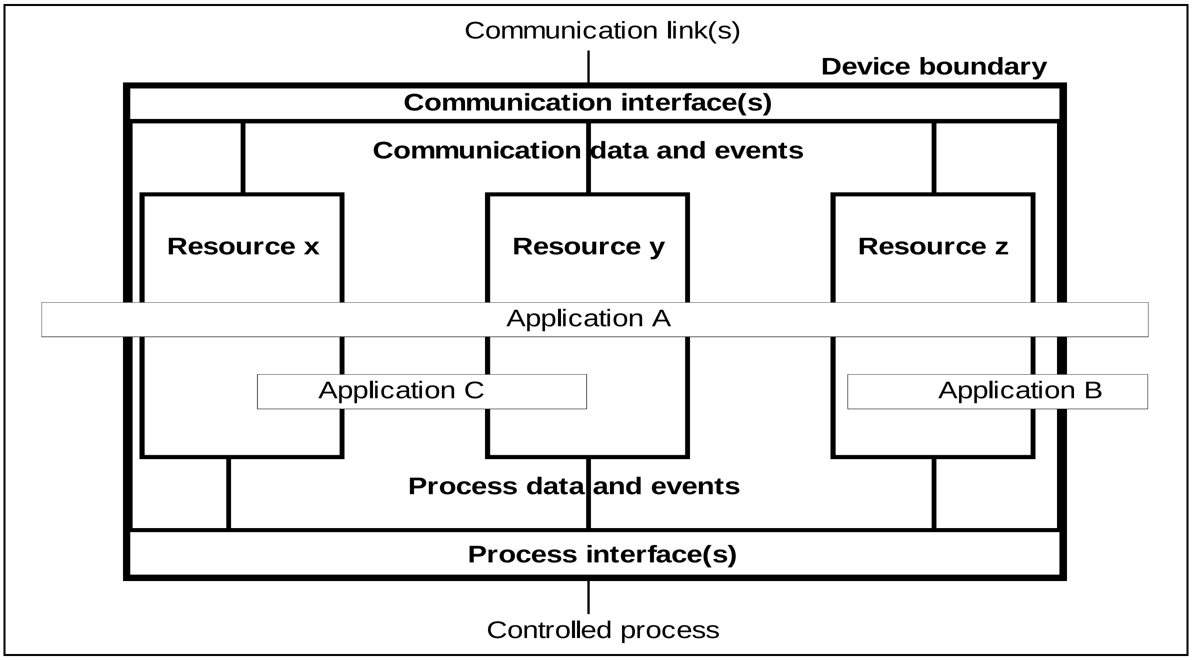

Figure 12 - Device model[15]

Figure 13 shows the IEC TC65/WG6 resource model. The functional capabilities of the resource, i.e., communications, process interface, and scheduling and execution of algorithms, are made available to a user application through *function blocks*. If the human interface is regarded as either a communications functionality or a specialized "process interface" functionality, all capabilities required for an ICS as shown in Figure 10 (except the Systems Engineering Interface) can then be supplied to an application via function blocks.

The TC65/WG6 model of an *application* consists of *event flow* and *data flow* among *function blocks*, as shown in Figure 14. Events flow between the upper portions (the *control blocks*) of the function blocks, and data flows between the lower portions. Flow of events and data is always in the left side and out the right side of the function blocks.

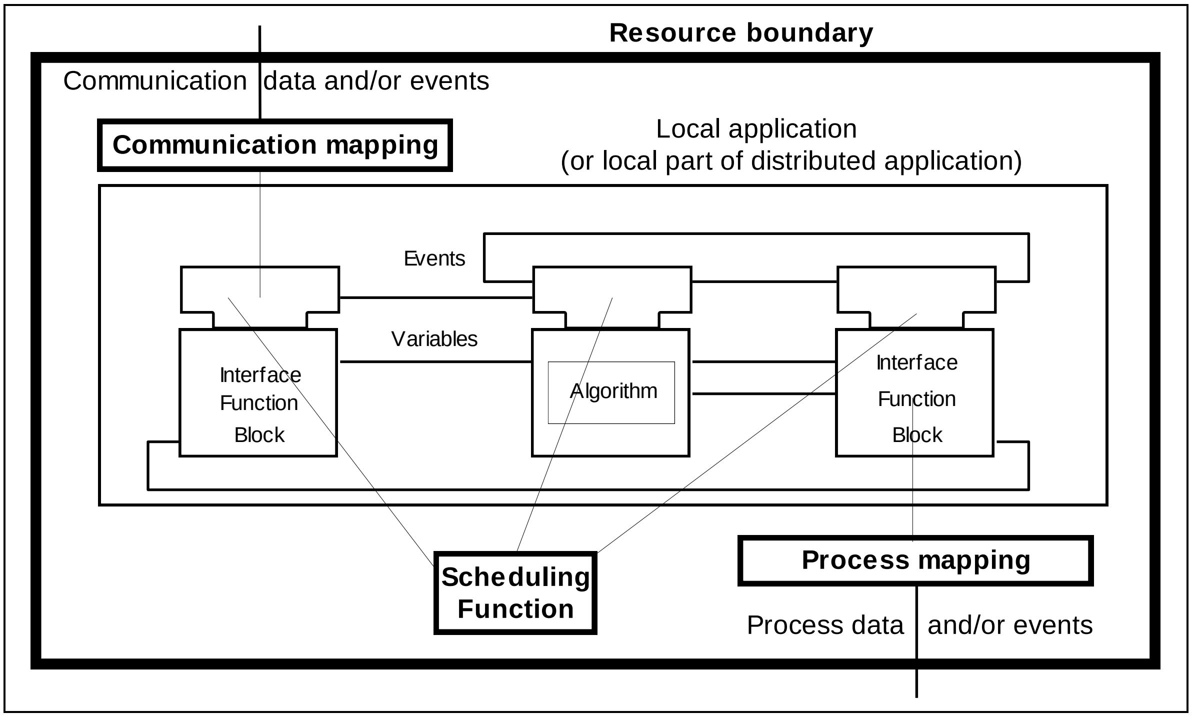

Figure 13 - Resource model[16]

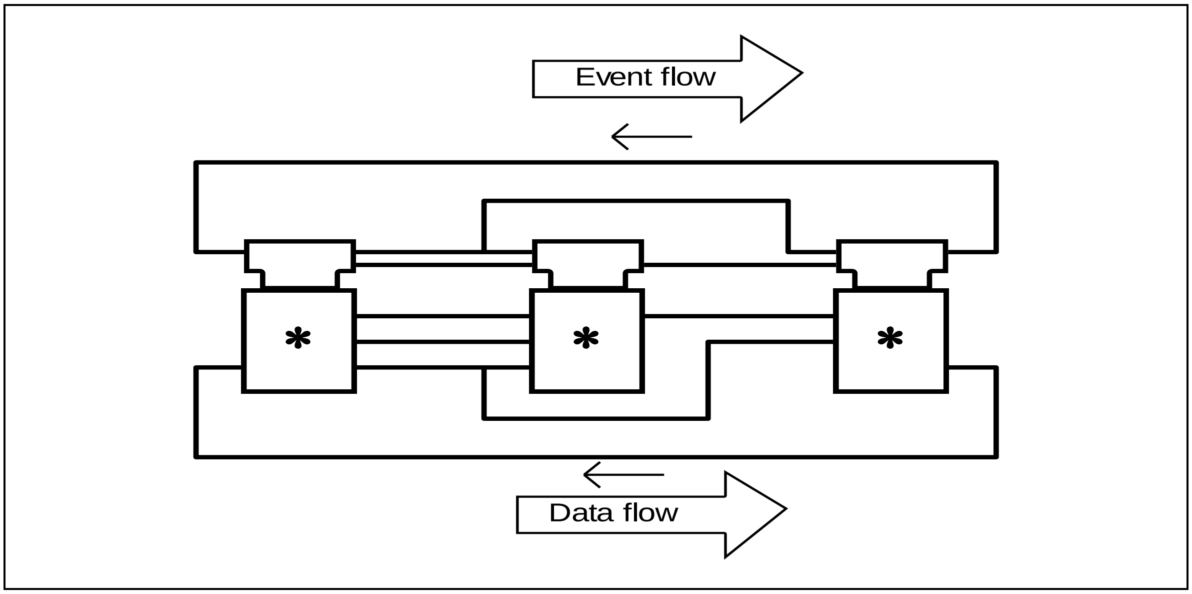

Figure 14 - Application model[17]

As shown in Figure 15, the TC65/WG6 model represents a *function block* as an *instance* of a *function block type*. The function block provides an interface between external data and events and encapsulated data and functional capabilities provided by the resource. The structure of the interfaces and internal data, associations between external events and internal processing, are specified in the function block's *type declaration.* Thus, the function block provides an object-oriented paradigm for construction of control applications.

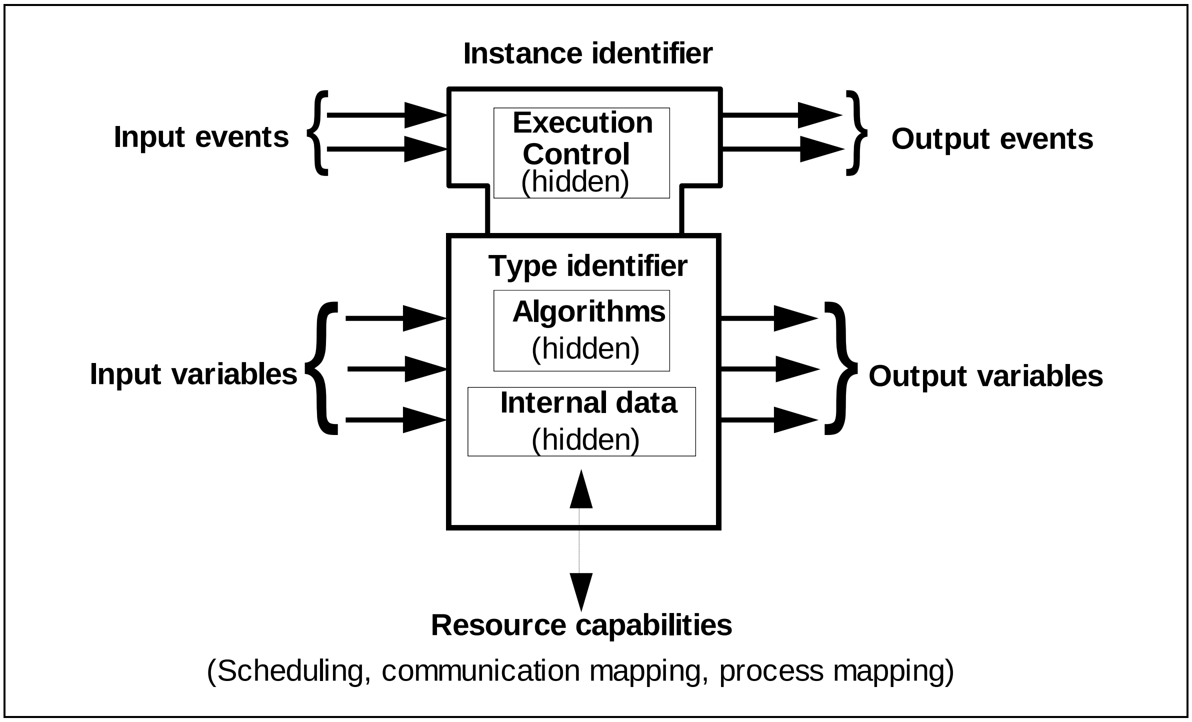

Figure 15 - Function block model[18]

A function block's algorithm may be written in one of the IEC standard languages for programmable control systems[19] or any other appropriate programming language. The IEC standard languages include the classical Ladder Diagram (LD), Sequential Function Chart (SFC) for state-machine (Petri net) oriented control, Function Block Diagram (FBD) itself, and a compatible, Pascal-like Structured Text (ST). Thus, the Function Block paradigm provides the necessary facilities for "top-down" system design via functional decomposition and "bottom-up" implementation via functional composition, as well as the encapsulation and reuse mechanisms required for incremental system improvement via *kaizen.*

In order to build holonic systems, function blocks must be developed that encapsulate capabilities which can provide **autonomy** such as:

- Encapsulated local data bases
- Local process/machine control
- Local optimization
- Local product tracking
- Self-scheduling
- Self-diagnosis
- Self-repair
- Self-configuration

As illustrated in Figure 16, function blocks must also be developed with **communication** and **negotiation** capabilities to enable of all the above functions to be accomplished in **distributed** and **cooperative HMS applications.**

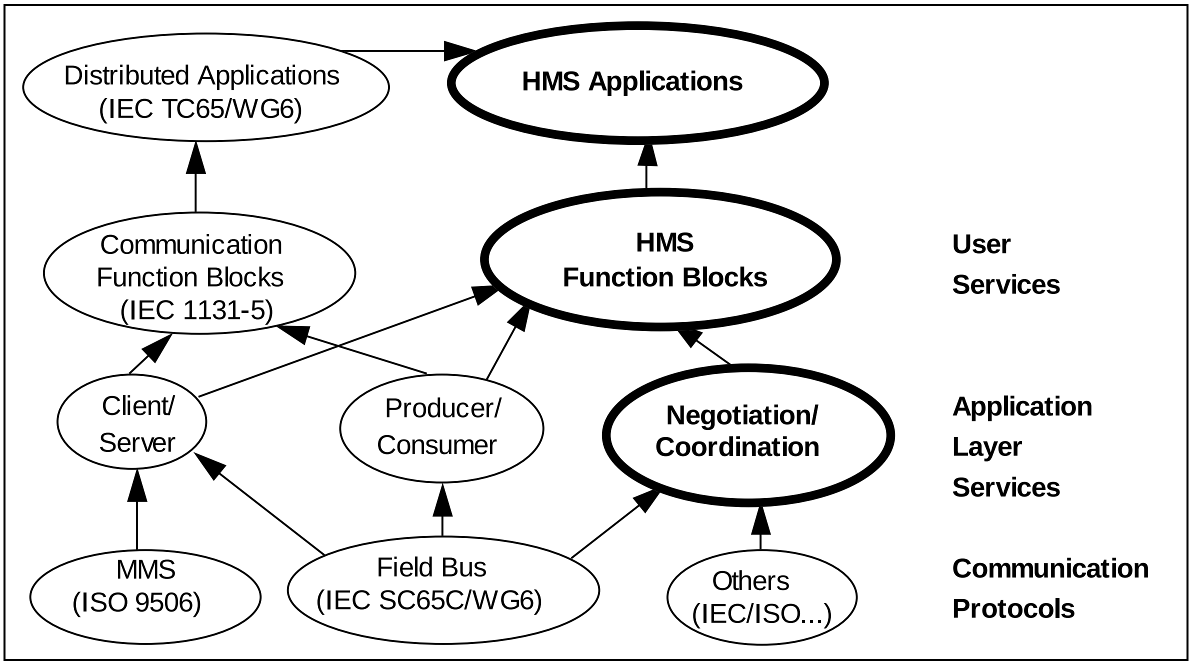

Figure 16 - Construction of HMS applications

Finally, consideration must be given to the **System Engineering Interface** in the development of the Function Block standard**.** In particular, TC65/WG6 or its successors must address the means for providing mechanisms for the dynamic reconfiguration of systems to meet the agility requirements of HMS. This should include mechanisms by which humans, other holons, or the holonic application itself can:

- dynamically create, modify,destroy, and relocate both instances and type definitions of function blocks;
- dynamically create and destroy connections among function block instances;
- dynamically activate and de-activate function block instances;
- perform version management of function block instances, classes, and applications.

Recommendations

There are many tasks to accomplish before full deployment of HMS technology becomes a reality. Users and vendors of control systems and automation technology could apply the following variation on the HMS system engineering process to take incremental steps toward full HMS deployment, realizing corresponding incremental benefits:

1. Develop an initial set of computer-assisted tools supporting the HMS engineering methodology.
2. Utilize the tool kit to perform conceptual designs of HMS solutions to existing problems in manufacturing systems.
3. Evaluate the proposed solutions using simulation and validation tools.

4. Attempt to install promising solutions in test beds using configuration and commissioning tools.
5. Evaluate the results to determine the requirements for:
    - improved tool kits;
    - improved control system elements and interfaces;
    - improved standards for HMS elements and interfaces.
6. Implement the indicated improvements, and repeat the process to achieve another increment of benefits.

Clearly, implementation of this process requires close collaboration among users, vendors and researchers in the proposed full scale IMS program. Each user or vendor will have to weigh the benefits and risks of various degrees of collaboration in order to arrive at an optimal twenty-first century manufacturing strategy.

ACKNOWLEDGMENTS

Many researchers have contributed to the understanding of architecture and standards for HMS. In particular, the author wishes to acknowledge the contributions of the following individuals:

E. Garcia-Herreros	SOFTING GmbH	Germany
M. Hoepf	FhG/IPA	Germany
J. Martin Prado	Tekniker	Spain
D. McFarlane	BHP	Australia
M. Mey	IFW/U. of Hannover	Germany
T. Moriwaki	Kobe U.	Japan
J. Oblak	United Technologies	USA
D. Seidel	IFW/U. of Hannover	Germany
T. Strasser	BHP	Australia
S. Tamura	Toshiba	Japan
P. Valckenaers	KU Leuven	Belgium
P. Wilbers	BHP	Australia

This document represents the author's synthesis and extension of the contributions of these and other researchers, and should not be construed as a consensus or as representing specific views of any particular individual or organization. Of course, all responsibility for any inconsistencies, errors, or omissions rests solely with the author.

REFERENCES

[1] A. Koestler, *The Ghost in the Machine*, 1971, ISBN 0-14-019192-5
[2] H. Suda, "Future Factory System Formulated in Japan," *TECHNO JAPAN 22*, 10 (October 1989), 15-25.
[3] H. Suda, "Future Factory System Formulated in Japan (2)," *TECHNO JAPAN 23*, 3 (March 1990), 51-61.
[4] HMS Consortium, *Glossary of Terms*,31 March 1994, p. 28.

REFERENCES - cont'd.

[5] Iacocca Institute, *21st Century Manufacturing Enterprise Strategy: An Industry-Led View,* ISBN 0-9624866-3-9, 1991, p. 7 (bold face added).
[6] P. Valckenaers, *WP3: Critical Factors, Chairman's Introduction,* HMS Consortium Document # HMS/WP6(Valckenaers)1214, 14 December 1993.
[7] H.J. Warnecke, *The Fractal Company,* Springer-Verlag, New York (1993), ISBN 0-387-56537-X.
[8] HMS Consortium, *HMS - Strategies, Vol. 0,* 31 March 1994, p.11.
[9] Dr. Ivar Jacobson, Objective Systems AB.
[10] ISO TR 10314/1, "Reference Model for Shop Floor Production Standards," 1989
[11] HMS Consortium, *Standardization - Areas,* 31 March 1994, p. 11.
[12] HMS Consortium, *Standardization - Proposals,* 31 March 1994, pp. 12-16.
[13] IEC TC65/WG6(PT1CDV)1, *Committee Draft - Function Blocks for Industrial-Process Measurement and Control Systems, Part 1 - General Requirements,* 1991-10-25.
[14] *ibid.*, p. 8.
[15] *ibid.*, p. 9.
[16] *ibid.*, p. 10.
[17] *ibid.*, p. 11.
[18] *ibid.*, p. 12.
[19] IEC 1131-3, *Programmable Controllers - Part 3: Programming Languages,* International Electrotechnical Commission, Geneva, 1993.