

Reference Architecture for Holonc Manufacturing Systems: PROSA

Hendrik Van Brussel, Jo Wyns, Paul Valckenaers, Luc Bongaerts, Patrick Peeters

Katholieke Universiteit Leuven, Mechanical Engineering Department,
Celestijnenlaan 300B, B-3001 Leuven, Belgium,
Tel: 32-16-32 24 80 ; fax: 32-16-32 29 87
e-mail: Hendrik.VanBrussel@mech.kuleuven.ac.be
<http://www.mech.kuleuven.ac.be/pma/pma.html>

**Computers In Industry, special issue on intelligent manufacturing systems,
Vol. 37, No. 3, pp. 255 - 276, 1998.**

PMA internal publication number: 97p077

Reference Architecture for Holonc Manufacturing Systems: PROSA

Hendrik Van Brussel, Jo Wyns, Paul Valckenaers, Luc Bongaerts, Patrick Peeters

Katholieke Universiteit Leuven, Mechanical Engineering Department,
Celestijnenlaan 300B, B-3001 Leuven, Belgium,
Tel: 32-16-32 24 80 ; fax: 32-16-32 29 87

Hendrik.VanBrussel@mech.kuleuven.ac.be

Jo.Wyns@mech.kuleuven.ac.be

Paul.Valckenaers@mech.kuleuven.ac.be

Luc.Bongaerts@mech.kuleuven.ac.be

Patrick.Peeters@mech.kuleuven.ac.be

<http://www.mech.kuleuven.ac.be/pma/pma.html>

Reference Architecture for Holonc Manufacturing Systems: PROSA

Biography of the authors

<Photo's included>

prof. dr. ir. Hendrik Van Brussel, born at Ieper, Belgium on 24 October 1944, obtained the degree of Technical Engineer in mechanical engineering from the Hoger Technisch Instituut in Ostend, Belgium in 1965 and an engineering degree in electrical engineering at M.Sc. level from Katholieke Universiteit Leuven, Belgium. In 1971 he got his Ph.D. Degree in mechanical engineering, also from K.U.Leuven. From 1971 until 1973 he was establishing a Metal Industries Development Centre in Bandung, Indonesia and he was an associate professor at Institut Teknologi Bandung, Indonesia. Consequently, he returned to K.U.Leuven where he is presently full professor in mechatronics and automation and chairman of the Department of Mechanical Engineering. He was a pioneer in robotics research in Europe and an active promotor of the mechatronics idea as a new paradigm in machine design. He has published more than 200 papers on different aspects of robotics, mechatronics and flexible automation. His present research interest are shifting towards holonic manufacturing systems and precision engineering, including microrobotics. He is Fellow of SME and IEEE and in 1994 he received a honorary doctor degree form the 'Politehnica' University in Bucarest, Romania and from RWTH, Aachen, Germany. He is also a Member of the Royal Academy of Sciences, Literature and Fine Arts of Belgium and Active Member of CIRP (International Institution for Production Engineering Research).

ir. Jo Wyns received the mechanical engineering degree in 1993 from the Katholieke Universiteit Leuven, Belgium. He is a candidate for the Ph.D. degree in mechanical engineering from the K.U.Leuven. Since 1993, he is with the Mechanical Engineering Department, division PMA, of the K.U.Leuven. His main research interests are in holonic manufacturing systems architecture, and dynamic resource allocation.

dr. ir. Paul Valckenaers received the applied mathematics engineering degree in 1983, the computer science engineering degree in 1985, and the mechanical engineering Ph. D. degree in 1993, all from the Katholieke Universiteit Leuven, Belgium. Since 1986, he is with the mechanical engineering department, division PMA, of the Katholieke Universiteit Leuven. His main research interests are in programming, scheduling and control of flexible production systems and design theory for the development of complex adaptive production systems. His current research activities focus on Holonic Manufacturing Systems (HMS).

ir. Luc Bongaerts received the mechanical engineering degree in 1992 from the Katholieke Universiteit Leuven, Belgium. He is a candidate for the Ph.D. degree in mechanical engineering from the K.U.Leuven. Since 1992, he is with the Mechanical Engineering Department, division PMA, of the K.U.Leuven, first as an IWONL bursary, from 1994 as an IWT bursary, and from 1996 as a K.U.Leuven bur-sary on the Concerted Research Action on Holonic Manufacturing Systems. From 1998, he is project engineer in the HMS/HANDS project. He is also involved in the MASCADA project and in the Esprit Working Groups IiMB and IMS-WG. His main research interests are scheduling and control of flexible and holonic manufacturing systems.

ir. Patrick Peeters received the mechanical engineering degree in 1996 from the Katholieke Universiteit Leuven, Belgium. He is a candidate for the Ph.D. degree in mechanical engineering from the K.U.Leuven. Since 1996, he is with the Mechanical Engineering Department, division PMA, of the K.U.Leuven. His main research interests are multi-agent systems for manufacturing control, flexible flow lines, and plant modelling and simulation.

Reference Architecture for Holonc Manufacturing Systems: PROSA

Hendrik Van Brussel, Jo Wyns, Paul Valckenaers, Luc Bongaerts, Patrick Peeters

Katholieke Universiteit Leuven, Mechanical Engineering Department,
Celestijnenlaan 300B, B-3001 Leuven, Belgium,
Tel: 32-16-32 24 80 ; fax: 32-16-32 29 87
e-mail: Hendrik.VanBrussel@mech.kuleuven.ac.be
<http://www.mech.kuleuven.ac.be/pma/pma.html>

Abstract. Future manufacturing systems need to cope with frequent changes and disturbances. As such, their control requires constant adaptation and high flexibility. Holonic manufacturing is a highly distributed control paradigm that promises to handle these problems successfully. It is based on the concept of autonomous co-operating agents, called 'holons'.

This paper gives an overview of the holonic reference architecture for manufacturing systems as developed at PMA-KULeuven. This architecture, called PROSA, consists of three types of basic holons: order holons, product holons, and resource holons. They are structured using the object-oriented concepts of aggregation and specialisation. Staff holons can be added to assist the basic holons with expert knowledge. The resulting architecture has a high degree of self-similarity, which reduces the complexity to integrate new components and enables easy reconfiguration of the system.

PROSA shows to cover aspects of both hierarchical as well as heterarchical control approaches. As such, it can be regarded as a generalisation of the two former approaches. More importantly, PROSA introduces significant innovations: the system structure is decoupled from the control algorithm, logistical aspects can be decoupled from technical ones, and PROSA opens opportunities to achieve more advanced hybrid control algorithms.

Keywords. reference architecture, holonic manufacturing system, shop floor control, CIM, IMS

1. Introduction

Market demand and environmental/societal pressures require effective manufacturing systems to adapt themselves at an ever-increasing pace. This creates the need for novel manufacturing control systems that are able to manage production change and disturbances both effectively and efficiently. To meet these new requirements, several new manufacturing paradigms are being investigated: bionic manufacturing [1], genetic manufacturing [2], the fractal factory [3], random manufacturing [4], virtual manufacturing [5], and holonic manufacturing [6].

The holonic manufacturing paradigm was developed in the framework of the Intelligent Manufacturing Systems (IMS) programme. In a feasibility study [7], conducted in 1994, six test cases were considered, one of which was 'Holonic Manufacturing Systems: system components of autonomous modules and their distributed control,' or HMS. The HMS project aimed at a better understanding of the requirements for future-generation manufacturing systems and at ways to build systems satisfying these requirements. A holonic manufacturing architecture shall enable easy (self-)configuration, easy extension and modification of the system, and allow more flexibility and a larger decision space for higher control levels.

Presently, PMA-KULeuven continues its research on holonic manufacturing systems by carrying out a nationally funded research project called GOA/HMS – Concerted Research Action on Holonic Manufacturing Systems. In this project, PMA combines its knowledge in flexible shop floor control [8], non-linear process planning [9], reactive scheduling [10], and machine controllers to develop a holonic architecture for production systems and to implement two prototypes or testbeds. The first testbed is a five-axis milling workstation which will be used to demonstrate the NC-workstation holon [11]. The PMA-flexible assembly system [12] serves as testbed for implementing the holonic shop floor control architecture.

This paper describes the holonic reference architecture for manufacturing systems developed in GOA/HMS. A reference architecture is defined as a set of coherent engineering and design principles used in a specific domain. It aims at structuring the design of a specific system architecture by defining a unified terminology, the structure of the system, responsibilities of system components, by providing standard (template) components, by giving examples, etc. [13].

After a brief introduction into the holonic manufacturing concept, the paper discusses the structure of the architecture: the components, their responsibilities, and their interactions. Section 4 motivates this design by comparing it to existing architectures. In the next two sections, more detailed aspects of the architecture are discussed: Section 5 lists the data, functions, and interaction behaviour of the system components. Section 6 describes the importance of the self-similarity encountered in this architecture. Finally, the section on different manufacturing viewpoints alerts the reader to the fact that depending upon his/her personal manufacturing background, the fundamentals of the described architecture may be perceived differently.

2. Holonic Manufacturing Concept

Over 30 years ago, Arthur Koestler proposed the word “*holon*” [14]. It is a combination of the Greek *holos* = whole, with the suffix *-on* which, as in *proton* or *neutron*, suggests a particle or part.

Two observations impelled Koestler to propose the concept of holon. (a) Complex systems will evolve from simple systems much more rapidly if there are stable intermediate forms than if there are not; the resulting complex systems in the former case will be hierarchic. (b) Although it is easy to identify sub-wholes or parts, ‘*wholes*’ and ‘*parts*’ in an absolute sense do not exist anywhere. This made Koestler propose the word *holon* to describe the hybrid nature of sub-wholes/parts in real-life systems; holons simultaneously are self-contained wholes to their subordinated parts, and dependent parts when seen from the inverse direction.

The HMS consortium translated the concepts that Koestler developed for social organisations and living organisms into a set of appropriate concepts for manufacturing industries. The goal is to attain in manufacturing the benefits that holonic organisation provides to living organisms and societies, i.e., stability in the face of disturbances, adaptability and flexibility in the face of change, and efficient use of available resources. The HMS concept combines the best features of hierarchical and heterarchical organisation [15]. It preserves the stability of a hierarchy while providing the dynamic flexibility of a heterarchy.

The HMS consortium developed the following list of definitions to help understand and guide the translation of holonic concepts into a manufacturing setting:

- **Holon:** An autonomous and co-operative 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 be part of another holon.
- **Autonomy:** The capability of an entity to create and control the execution of its own plans and/or strategies.
- **Co-operation:** A process whereby a set of entities develops mutually acceptable plans and executes these plans.
- **Holarchy:** A system of holons that can co-operate to achieve a goal or objective. The holarchy defines the basic rules for co-operation of the holons and thereby limits their autonomy.

More detailed definitions and historical background on Holonic Manufacturing Systems can be found in [7]. Information on HMS, the PMA project on HMS, and abstracts of related publications can be found on internet at <http://www.mech.kuleuven.ac.be/pma/research/hms>.

3. Structure of the HMS reference architecture

The structure of the HMS reference architecture is built around three types of basic holons: order holons, product holons, and resource holons. Each of them is responsible for one aspect of manufacturing control, be it logistics, technological planning, or resource capabilities respectively. These basic holons are structured using object-oriented concepts like aggregation and specialisation. Staff holons can be added to assist the basic holons with expert knowledge. These allow for the use of centralised algorithms and for the incorporation of legacy systems.

The name PROSA stands for Product-Resource-Order-Staff Architecture, which refers to the composing types of holons.

3.1 Basic holons

As well in the research community as in manufacturing companies, three relatively independent manufacturing concerns do exist: (i) resource aspects such as driving the machine at optimal speed and maximising its capacity, (ii) product and process related technological aspects such as which operations need to be performed to achieve a good quality product, (iii) logistical concerns about customer demands and due dates. From this analysis, our research group concluded that there are three types of basic holons, namely resource holons, product holons, and order holons (see Figure 1) [16]. A holonic manufacturing system (manufacturing holarchy) consists of resource holons, product holons and order holons.

- A *resource holon* contains a physical part, namely a production resource of the manufacturing system, and an information processing part that controls the resource. It offers production capacity and functionality to the surrounding holons [13]. It holds the methods to allocate the production resources, and the knowledge and procedures to organise, use and control these production resources to drive production. A resource holon is an abstraction of the production means such as a factory, a shop, machines, furnaces, conveyors, pipelines, pallets, components, raw materials, tools, tool holders, material storage, personnel, energy, floor space, etc.

In contrast with most traditional shop floor control architectures, such as PAC [17], the HMS does not separate the manufacturing system from the manufacturing control system. The HMS comprises both. A physical manufacturing resource is incorporated inside a resource holon.

- A *product holon* holds the process and product knowledge to assure the correct making of the product with sufficient quality. A product holon contains consistent and up-to-date information on the product life cycle, user requirements, design, process plans, bill of materials, quality assurance procedures, etc. As such, it contains the “product model” of the product type, not the “product state model” of one physical product instance being produced. The product holon acts as an information server to the other holons in the HMS.

The product holon comprises functionalities which is traditionally covered by product design, process planning, and quality assurance.

- An *order holon* represents a task in the manufacturing system. It is responsible for performing the assigned work correctly and on time. It manages the physical product being produced, the product state model, and all logistical information processing related to the job. An order holon may represent customer orders, make-to-stock orders, prototype-making orders, orders to maintain and repair resources, etc. Often, the order holon can be regarded as the workpiece with a certain control

behaviour to manage it to go through the factory, e.g. to negotiate with other parts and resources to get produced.

The order holon performs tasks traditionally assigned to a dispatcher, a progress monitor, and a short term scheduler.

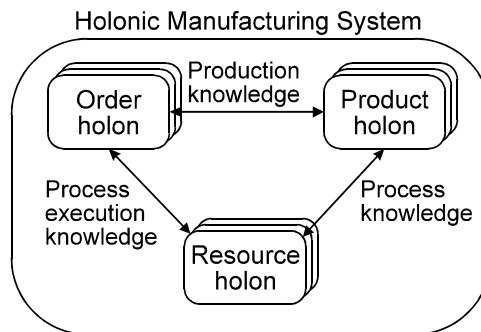


Figure 1: Basic building blocks of a HMS and their relations

As shown in Figure 1, these three types of holons exchange knowledge about the manufacturing system. Product holons and resource holons communicate process knowledge, product holons and order holons exchange production knowledge, and resource holons and order holons share process execution knowledge.

- *Process knowledge* contains the information and methods on how to perform a certain process on a certain resource. It is knowledge about the capabilities of the resource, which processes it can perform, the relevant process parameters, the process quality, possible outcomes of a process, etc.
- *Production knowledge* represents the information and methods on how to produce a certain product using certain resources. It is knowledge about the possible sequences of processes to be executed on the resources, data structures to represent the outcome of the processes, methods to access information of process plans, etc.
- *Process execution knowledge* contains the information and methods regarding the progress of executing processes on resources. It is knowledge about how to request the starting of processes on the resources, making reservations on resources, how to monitor the progress of execution, how to interrupt a process, the consequences of interrupting a process, suspending and resuming processes on resources, etc.

Modelling

Figure 1 shows the three basic holons and their relationship in an informal drawing. For the remainder of this text, more formalised drawings are shown. These drawings use UML (Unified Modelling Language [18]), a standard for drawing object-oriented diagrams. UML emerged out of the combination of the Booch Method [19] and OMT [20]. An object-oriented modelling tool is chosen because the concepts of HMS are closely related to the concepts of object-oriented design. For those not familiar with these diagrams, a short explanation of the syntax is in place.

For the moment, only the class diagrams of UML are used. Using UML, Figure 1 is redrawn as in Figure 2. Every rectangle represents an entity type in the system. In this case, it represents a holon type. Every line represents a relation. Depending on the symbol on the line (arrow, diamond, or no symbol), the line refers to a different kind of relation. The aggregation relation or "has-a" relation is represented by a line with a diamond. On Figure 2, the holonic manufacturing system has product holons, resource holons, and order holons. The specialisation relation or "is-a" relation is represented using a line with an arrow. This relation is not present on Figure 2. The "association"-relation is represented using a normal line. E.g. on Figure 2, the order holon type has an association with the product holon type.

Both ends of the line usually have numbers associated with them. This is called the cardinality (0..*, 1..*, 0..1, 0, 1, or unspecified), and it represents the number of this type of holons that are involved in this relation. Figure 2 shows that holonic manufacturing system contains at least one resource holon and one product holon, and it may contain zero or more order holons. When containing zero order holons, the HMS is idling. Figure 2 also shows that multiple (0..*) order holons can be producing the same kind

of product. However, every order holon refers to exactly one product holon. If the order would be allowed to process several products simultaneously, technological constraints might occur between these products. Technological constraints are to be handled by the product holon, so in such case an aggregated product holon is to be composed which tackles the technological constraints and acts as one single product holon towards the order holon.

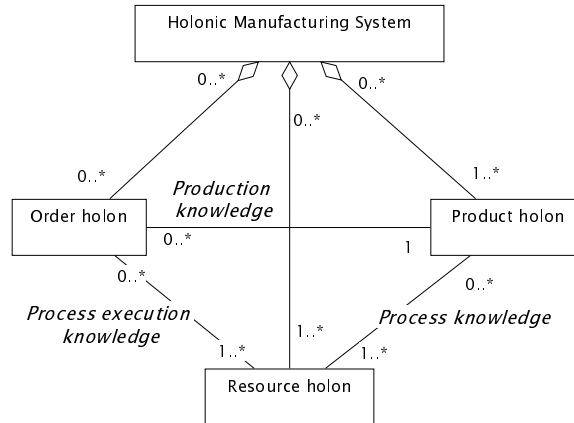


Figure 2: UML version of Figure 1: Basic building blocks of a HMS and their relations

3.2 Aggregation

Interaction between a large number of low-level agents results in a complex system behaviour which is difficult to understand, to control and to predict [21]. Structuring the agents in a hierarchy is the appropriate solution to tackle this complexity [14].

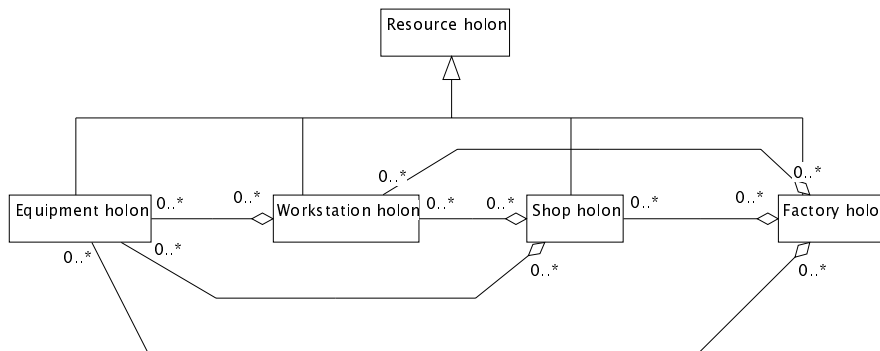


Figure 3: Aggregation of resource holons

Therefore, aggregated holons are defined as a set of related holons that are clustered together and form on their turn a bigger holon with its own identity. As such, an aggregation hierarchy is formed, which is open-ended at the top and at the bottom. Depending on the study scope of the observer, holons are split up into their sub-holons or treated as a whole.

The aggregation hierarchy is not necessarily a tree-shaped one: holons may belong to multiple aggregations, e.g. a tool can be shared between several workstations. Aggregated holons are no static sets of holons, but can dynamically change their contents depending on needs of the system. Aggregated holons may emerge out of the self-organising interaction of holons or they may be designed up front. The number of hierarchical levels depends on the specific needs of a certain system, and is not dictated by the architecture.

Figure 3 indicates an aggregation structure for resource holons. All holons on this diagram are resource holons (the arrow indicates a specialisation relation - see section 3.3). Equipment holons can be grouped into a workstation holon, workstation holons are grouped into a shop holon, shops are grouped into a factory. Since a holon may belong to multiple aggregations, the cardinality indicates 0..* saying that an equipment may belong to zero or more workstations, and a workstation contains zero or more equipments, etc. Moreover, a holon may belong to different levels of aggregation at the same time: an equipment holon may belong as well to a workstation holon, as to a shop holon, as to a factory holon. Note that this aggregation is not static, therefore Figure 3 can only be a snapshot of the architecture.

Similar to the resource aggregation, an aggregation of order holons and of product holons can be represented. A customer order can be divided into separate batch orders, and further into work orders. A product can be divided into subassemblies, and subassemblies consist of several operations.

3.3 Specialisation

Specialisation separates the holons with respect to their characteristics. Already a specialisation took place by separating the basic holons into three kinds: order holons, product holons and resource holons. In a specific architecture, these basic holons may still be too abstract to reason about. Specialisation can then be used to differentiate between the different kinds of resource holons, order holons, and product holons.

Figure 4 gives a specialisation representation of resource holons. Both assembly shops and milling shops are kinds of shop holons. Shop, workstation, and equipment are all resource holons. Similarly, specialisation can be used to structure the order and product types. Order holons can represent customer orders, stock orders or maintenance tasks. Products can be divided into several product families. Since all these specialised holons will require specialised behaviour, they may require different research and implementation approaches. For instance, the control of a robot will require to focus on issues such as calibration and tool exchange; while conveyors, CMM stations, or assembly shops will require efforts in other areas.

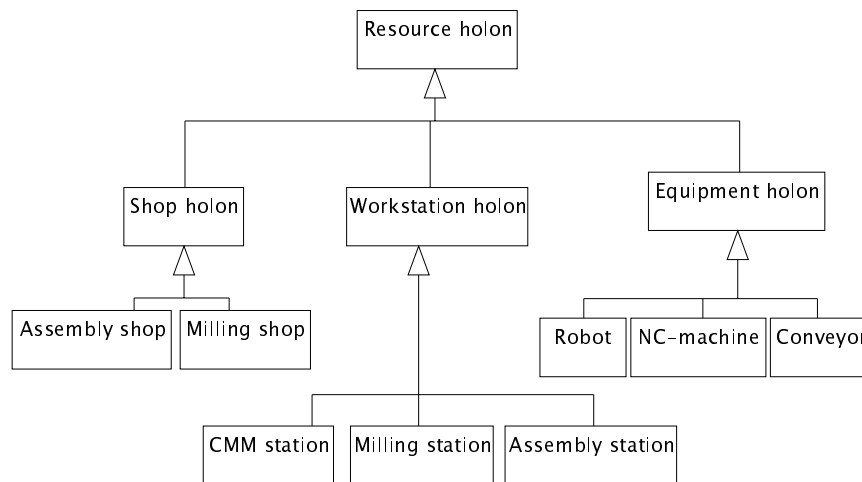


Figure 4: Specialisation of resource holons

3.4 Staff holons

The architecture foresees in the possibility to provide staff holons to assist the basic holons in performing their work [16]. They consider some facets of the problems of the basic holons and provide them with sufficient information such that these can take the correct decision to solve the problem. The basic holon is still responsible for taking the decision, and the staff holon is considered as an external expert that gives advise. The name "staff holon" is inspired by the difference between line functions and staff functions in human organisations [22]. Also in a human organisation one of the main goals for the

introduction of staff functions is to reduce the work load and work complexity of line functions by providing them with expert knowledge.

The concept of staff holons allows for the presence of centralised elements and functionality in the architecture. This is useful for problems which are too hard to solve in a distributed manner, and it allows a smooth migration from current hierarchical shop floor control systems to a holonic architecture. However, it does not introduce a hierarchical rigidity into the system, since the final decision is still to be taken by the basic holon.

In fact, the concept of basic holons, enhanced with staff holons giving advice, decouples robustness and agility from system optimisation. Due to its distributed basic architecture, the HMS delivers robustness and agility and is simple to extend and reconfigure. When staff holons are added, the basic holons will follow this advice as well as possible. When, due to disturbances and changes in the system, the hierarchical staff holons perform badly, the advice may be ignored by the basic holons, which again take autonomous actions to do their work. On the other hand, when disturbances are absent, an HMS can be configured such that the basic holons do follow (in a hierarchical way) the advice of the staff holon. This configuration is determined by a meta-controller, which defines the basic rules for co-operation of the holons present in the holarchy [23].

The centralised scheduler of a shop, as shown in Figure 5, is an example of a staff holon. It has an overview of all resources and all orders, it generates an (optimal) schedule, and it gives this schedule as advice to the individual order holons [10] and to the resource holons. By including a centralised scheduler, the system can exhibit hierarchical control behaviour as present in hierarchical control architectures like PAC [17]. Since the order holons take these schedules as advice instead of as commands, the centralised scheduler does not introduce rigidity in the system as in traditional hierarchical control systems. The cardinality in Figure 5 explains that there is at most one scheduler for a set of orders and a set of resources. Other examples of staff holons are on-line shop floor control holons, (central) process sequence planners, CAD-systems, and even MRP-systems. More elaborated examples of the use of staff functions are given in [10] and in [24].

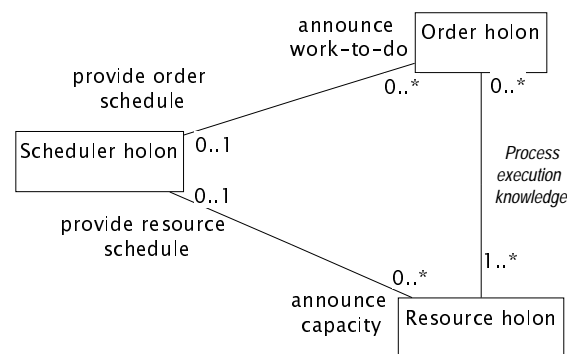


Figure 5: A centralised scheduler is a staff holon to the order holons and resource holons

This architecture leaves open opportunities for behaviours in between fully hierarchical and fully heterarchical control. As mentioned in previous paragraph, the system can dynamically switch between hierarchical and heterarchical control. However, scenarios can be thought of in which mixed behaviour may be more optimal. For instance, the system could apply hierarchical control for a bottle-neck resource to maximise throughput, and apply heterarchical control on some other resources to reduce operational cost on them, or to give the workers the ability to replan their work. The structure of the meta-controller is not further discussed here.

4. Comparison with other architectures

This section compares the structural characteristics of PROSA with the characteristics of the main competing reference architectures. This comparison serves as a motivation why PROSA is designed to consist out of three types of basic holons extended with the concept of staff holons.

From the comparison below, it is concluded that PROSA is covering all aspects of hierarchical and heterarchical control architectures. It covers all relevant functions and it can incorporate a wide range of

control algorithms from centralised to distributed approaches. As such, PROSA can be regarded as a generalisation of the two former approaches.

However, PROSA also introduces several important innovations: the system structure is decoupled from the control algorithm, logistical aspects are decoupled from technical ones, and PROSA allows to incorporate more advanced hybrid control algorithms [23] [25].

4.1 Existing architectures

In literature, two families of manufacturing control approaches exist: the hierarchical and the heterarchical control approach. This section briefly discusses the typical assumptions these approaches make about the manufacturing system and its environment. These assumptions constrain the applicability of the architectures and limit the evolution of the manufacturing control systems to follow unforeseen trends.

Dilts et al. give a detailed overview of the evolution from centralised control, over hierarchical and modified hierarchical, to heterarchical control. They give a detailed discussion on the characteristics, advantages, and drawbacks of each of these control concepts [15].

Hierarchical architectures

The natural perception of hierarchy in the structure of complex systems is the motivation to build hierarchical manufacturing control systems. The established hierarchy is used as the basis for structuring as well as for controlling the system. Commands flow top-down, and feedback information flows bottom-up. The break-down of high-level commands into smaller commands for subordinates is often based both on technological (process oriented) and logistical considerations.

The more recent examples (MSI [26], PAC++ [27], FACT [28]), sometimes called modified hierarchical control structures, provide also in a peer-to-peer communication between entities. This allows the lower levels to exchange data (MSI), to better synchronise their progress (PAC++), or to solve specific disturbances (FACT).

All hierarchical control architectures require a fixed structuring while the system is running, and assume deterministic behaviour of the components. These implicit axioms are the basis of the main drawbacks of hierarchical architectures:

- Modifying the structure, e.g. for adding, modifying, or deleting resources, is difficult. Firstly, it requires a shutdown of the system. Secondly, all data structures at higher levels need updating in order to allow accurate prediction of lower-level behaviour.
- The incorporation of unforeseen modifications, such as new data structures (e.g. process plan with alternatives) or new technology, is almost impossible [29].
- Prediction at higher levels is used to trigger the starting of resources. Disturbances, such as machine breakdowns, invalidate the planning and therefore propagate to the higher levels. In some cases, the schedule is already invalid before it is completely generated [30].
- Additionally, hierarchical control architectures almost automatically imply a top-down development methodology, which again introduces additional constraints into the solution.

Heterarchical control

Hatvany introduced the concept of co-operative heterarchies as an alternative to the dogmatic hierarchic control structures [31]. Heterarchical manufacturing control systems have a flat structure and are composed out of independent entities called agents. These agents typically represent resources and/or tasks, and have local intelligence to adapt to unforeseen situations. Instead of a master-slave relationship as in hierarchical systems, information and commands are exchanged by the use of a negotiation protocol in which for instance resources can refuse to accept certain jobs [15].

The “contract net protocol” is the best known example of such negotiation mechanism [32] [33]. It consists of five steps: task announcement by a manager, potential contractors evaluate task announcements from several managers, potential contractors bid upon the selected task, the task manager awards the contract to one of the bidding contractors, finally manager and contractor communicate to execute the contracted task. This protocol can be applied to the manufacturing context in several ways:

- managers and contractors are both resources using subcontracting to distribute the workload,
- managers are parts offering work and contractors are resources bidding to get this work assigned,
- managers are resources offering capacity and contractors can be parts bidding to use this capacity.

This yields a simple and fault-tolerant system, since none of the agents need a priori information about the other agents [34] [29]. As a consequence, several disturbances and changes can easily be

handled. When a resource is malfunctioning, it just doesn't take part in the market; when a resource is able to perform new operations, the associated agent starts responding to bids for such tasks; new resources only require the creation of an additional agent in the market; etc.

Again the basic axiom of this control paradigm gives rise to its principle drawbacks which obstruct a widespread use of this kind of control systems in industrial environments. The independence of agents prohibits the use of global information. Therefore, central scheduling or resource planning is impossible:

- The global system performance, e.g. throughput, is very sensitive to the definition of the market rules, and to the fine tuning of the rules (e.g. relative importance of transport times).
- The control system cannot guarantee a minimum performance level in case the system goes outside the working scope for which the rules were tuned. Only an average overall performance can be predicted when the system is inside its nominal working scope.
- Prediction of the behaviour of individual orders is impossible. The flow time of an order highly depends on the nature and status of other orders in the system.

Additionally, the advantages of the automatic handling of machine break-downs is only appropriate when alternative machines are available. Due to the absence of a process planning agent, alternative machines in practice often means identical machines. The most successful applications of heterarchical control need therefore to be sought for in applications which have a homogeneous set of resources, for instance distributed computing applications, or mobile robot applications.

4.2 Comparison

Three basic holons form a necessary set

In other manufacturing control paradigms, such as hierarchical control and heterarchical control, the functions covered by the order holon, product holon, and resource holon are also present in varying degrees. Often several functions are assumed to be performed as static up-front activities. The list of arguments below justify an architecture containing three basic holons as separate, active entities which execute the necessary manufacturing control functions.

The functionality of the product holon, i.e. the generation of process information, is traditionally considered as an off-line up front activity. However, in the case some disturbances like a machine breakdown or a missing tool, a technical replanning of the process can be more effective than a logistical replanning.

Distinguishing separation between order holon and resource holon is more generic than combining these two responsibilities in one entity as done by several heterarchical systems. The more advanced market mechanisms have both order (or part) agents and resource agents; scenarios with only resource agents implicitly have one resource agent responsible for the finished part. In the latter case, the resource agent is playing both the role of task executor and of task manager. Since both roles are independent, it is best to separate them.

The advantage of independent order agents, is that the different agents can have different objectives for optimisation [35].

Three basic holons form a sufficient set

Based on the fact that heterarchical control systems have sufficient functionality to run production, it is concluded that the three basic holons are sufficient to drive a production system. Moreover, the three basic holons cover for all critical manufacturing functions as present in hierarchical control system: scheduling of operations, control of inventory levels, co-ordination of machines to execute operations, maintenance planning, monitoring, development of control procedures, process planning, product design, quality control [36] [37].

Assumed that all algorithms could be effectively realised in a distributed implementation, the three basic holons are sufficient to construct a manufacturing control system.

Structure is decoupled from control algorithms

Decoupling is one of the main issues in the design of complex systems [38]. The decoupling of the control system structure and the control algorithm is therefore the most important characteristic of the PROSA architecture. The type of resource allocation algorithms is not determined by the PROSA structure.

Any existing heterarchical control algorithm can be implemented using order and resource holons. When including staff holons, also the hierarchical control algorithms can be implemented inside this structure. The presence of staff holons enables the inclusion of centralised functions such as a

centralised scheduler. Also modified distributed control algorithms requiring centralised elements are possible. A first example is Duffie's real-time distributed scheduling system [30]. All agents develop their own local plan, which are then evaluated in a time-scaled simulator of the real system. A central evaluator analyses the results of the simulation and feeds back a global performance measure to the agents. A second example is to use a centralised scheduler to give the distributed market mechanism an initial solution to start from [39] [10]. [39] combines an off-line scheduling algorithm based upon Lagrange relaxation together with a distributed on-line control based on a market mechanism. [10] uses a centralised reactive scheduler which sends the generated schedule to a set of order agents and resource agents.

Centralised staff holons are only giving advise to the basic holons. This reduces the potential impact of a performance degradation of these holons, e.g. due to outdated data models or changes in the lower levels. The advisory role of the staff holons makes it possible for several alternative staff holons to work in parallel to each other.

Technical aspects are decoupled from logistical aspects

Both technical process-related aspects as well as resource allocation aspects are crucial for the efficient operation of manufacturing systems and for a correct reaction to disturbances. By separating the process-related aspects from the logistical resource allocation aspects, both can evolve independently.

Implementation quality

The decoupling in the PROSA reference architecture should enable an intensive reuse of sub-systems in a wide range of manufacturing systems. This will lead to higher-quality implementations of these sub-systems, resulting in increased reliability, increased performance, reduced installation and maintenance costs, and a higher development pace.

The decoupling of structure from control allows to reuse structural modules in a different logistic chain, just by only replacing the control algorithms; the decoupling of technical from logistical aspects allows to reuse control algorithms in logistically similar factories just by replacing the product holons; default functions, such as the deadlock prevention mechanism [40][41], can be reused in all system, but can eventually be overwritten by system-specific solutions if required.

5. Detailed model of the basic holons

From the examples given in the previous sections, the reader may already have derived data, functions, and interaction behaviour of the holons. This section more explicitly describes the data held by the holons, the functions which each holon needs to fulfil, and interactions between the holons. The data and functions give a more detailed insight in the responsibilities of the three basic holons. The interaction behaviour gives simple examples of the process knowledge, production knowledge, and process execution knowledge (see Figure 2).

This enumeration of data, functions, and behaviour, is not restrictive, nor prescriptive: the system is not limited to this enumeration, nor do all these have to be present in every holonic system. It is a descriptive indication of which kind of data, functions, and behaviours can be expected.

The given examples are biased by the authors' background in discrete manufacturing research. For instance, process planning is a typical discrete manufacturing topic which applies to machining and assembly, it applies less to flow production, and has a totally different meaning in process industry. This leads to another aspect that needs attention: vocabulary is problem domain specific, the same word may have different semantics depending on the domain. An example of such word is "scheduling". In our examples it refers to short-range planning to assign resources to tasks.

5.1 Data managed by the basic holons

The data model shows the data held by the holons. Since the holonic reference architecture complies with the object-oriented paradigm, data do not exist on their own. Data always belong to the holon that maintains the data. Figure 6 is an extension of the model of Figure 2 and represents the data of the basic holons using the UML-syntax. The data are enumerated in the middle box inside the rectangle representing the holon type.

The resource holon maintains data on its capabilities (list of products), its running tasks, its sub-resources, and a log of its activities. The product holon holds a process plan, a product description, and the quality requirements. The order holon keeps track of the state of the physical product, the progress of the task, and historical data of the tasks.

5.2 Functions performed by the basic holons

Functions are represented together with the holons which performs the function. Similar to the data model, the functional model is also an extension of the model of Figure 2. On Figure 6 the functionalities of the holons are represented in the bottom box of the rectangles representing the holon types. This diagram is not to be interpreted as a “normal” class diagram of an object-oriented program which would show the interface functions of the holons. Instead, it describes the functionalities which are covered by these holons.

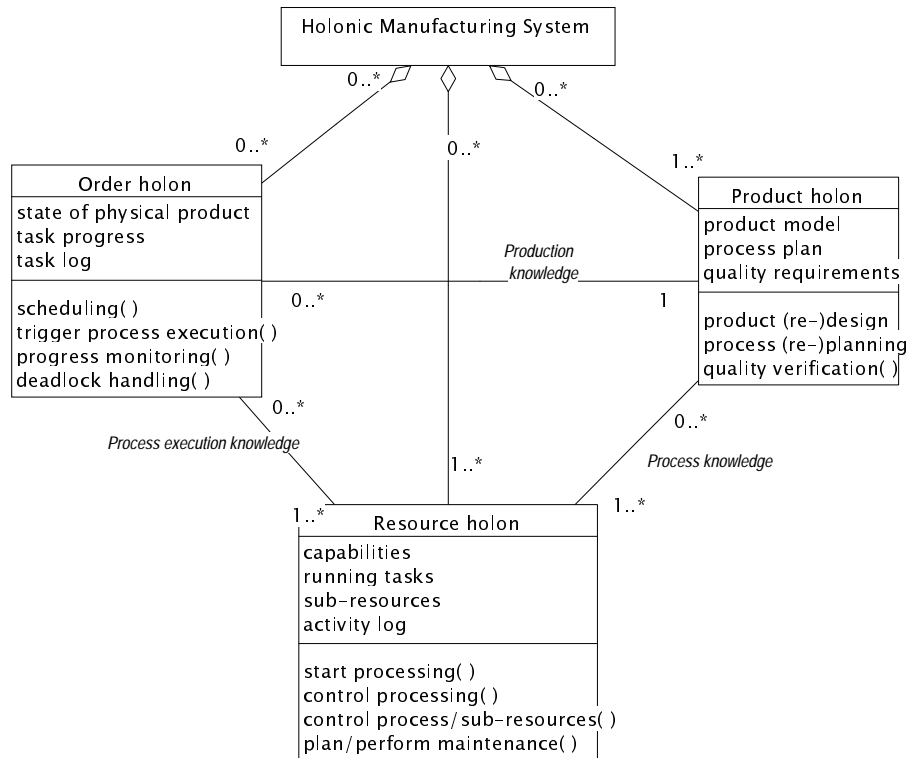


Figure 6: Mapping of data and functions to the three basic holons

The resource holon is able to start processing, this implicitly means that it is authorised to accept or reject a task which it was assigned. It controls the process execution (suspend, restart, abort), manages the sub-resources, controls the process, process monitoring, plans and performs maintenance, etc. [24].

The product holon covers the tasks of product (re-)design, process (re-) planning, and quality verification based on the result of tasks. The order holon does the scheduling [10], deadlock handling [41], progress monitoring, and triggering the starting, suspending, resuming, aborting, or stopping of a process on a resource [42].

5.3 Interaction behaviour of the basic holons

This section describes the behaviour of a holon by looking at its interactions with other holons in order to fulfil its goals. It gives simple examples of the process knowledge, production knowledge, and process execution knowledge (see Figure 2). In the examples, a holon interacts with its environment. Depending on the interaction, the holon uses its knowledge about the process, the production, and process execution to interact with the other holons, and to interpret their response.

The interaction behaviour is described by giving some example scenarios. Depending on the viewpoint of the reader (see section 6), other scenarios can be added.

The scenarios are modelled using UML’s collaboration diagram, or object interaction diagram. Where the class diagram (see Figure 2) shows the relation between holon types, this collaboration diagram shows the interaction between individual holons. In every rectangle, the name (optional) and type of an individual holon are given. If multiple rectangles are drawn, it indicates that multiple holons

with this role are present in the system. The lines represent links between the holons, the arrows represent interactions. The numbering of the arrows refers to the sequence in which the interactions take place. A full arrow (e.g. arrow 2 on Figure 7) refers to a simple interaction where the initiator waits for the completion of the task before continuing its processing. A half arrow (e.g. arrow 1 on Figure 7) refers to an asynchronous interaction where the initiator does not wait for completion of the task. An arrow with a circle indicates on important data transfer (e.g. arrow 2 on Figure 8).

Below, three scenarios are described. These explain how an order holon is created in the HMS, how an order holon interacts with product holon and resource holon to drive production, and how the adding of a resource holon influences the product holon and the order holon. Other scenarios can be added.

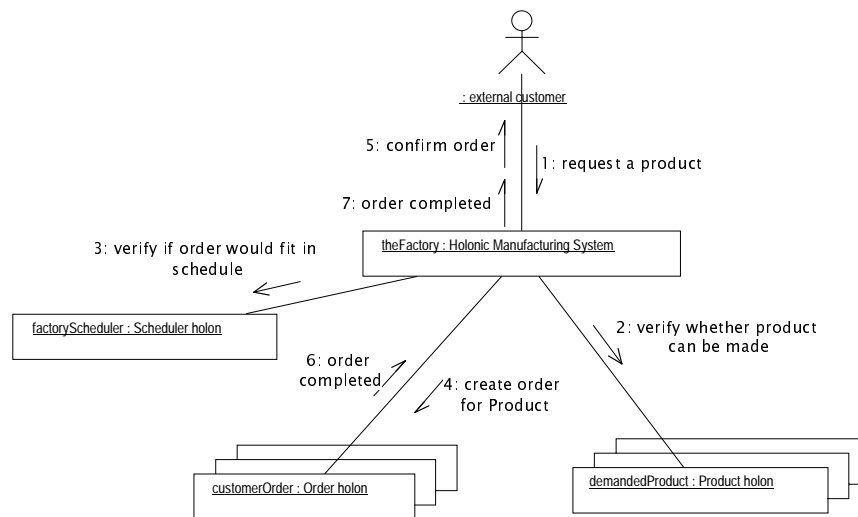


Figure 7: Launching an order in the holonic manufacturing system

Launch of an order

The scenario on Figure 7 illustrates the behaviour of a holonic manufacturing system when a customer requests the production of a product (arrow 1) to “theFactory”. The HMS verifies whether it is able to produce this kind of product (arrow 2), and whether it is able to fit it into its schedule (arrow 3). Next, it creates the appropriate order holon (arrow 4) and confirms the acceptance of the order to the customer (arrow 5). When the order holon completes its task (arrow 6), the HMS informs the customer (arrow 7).

Order drives production

The collaboration diagram of Figure 8 shows how an order holon interacts with its surrounding holons to execute its task. The order is initiated by an external customer, which can be a higher-level holarchy. The order holon requests a process plan from the product holon for the task it has to perform. It announces the scheduler (if this staff holon is present) of the tasks it needs to perform and request a schedule. If this schedule does not satisfy the order holon (e.g. because the order holon is a rush order), it negotiates with other order holons to get a better deal. Finally, the order holon starts the first of its tasks by requesting a resource holon to start a process. The customer is informed of the progress the order makes.

Adding a new resource

When a resource holon is added to a holarchy, it announces its presence to the other holons in the system (see Figure 9). The product holons take this event as a trigger to verify whether they can improve their process plans. The new resource holon could be better in performing some processes, or it could provide new processes to the system. Both cases lead to adding alternative operations to the existing process plans. If a process plan gets updated, the product holon announces this to the running order holons which are producing this product. This way, the order holon is able to use this new resource holon directly.

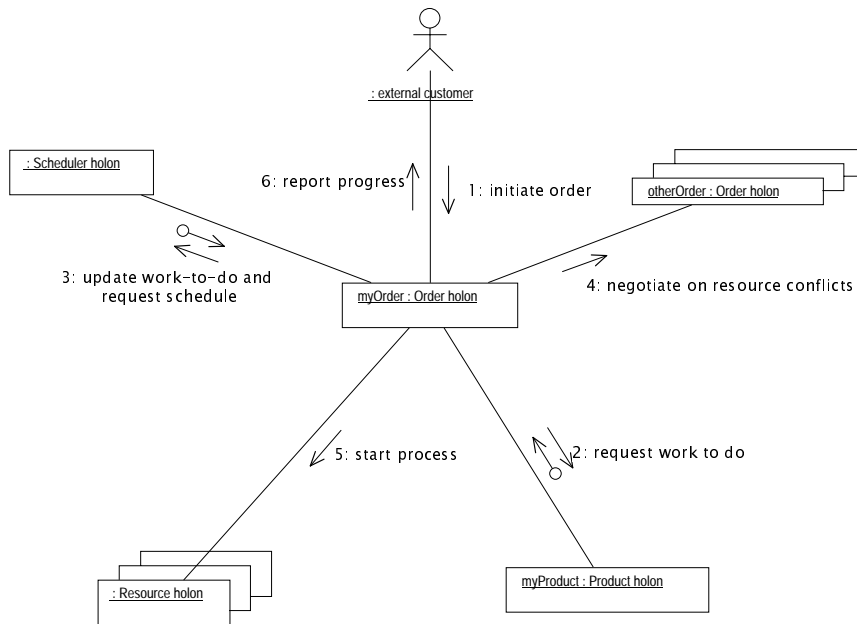


Figure 8: Interaction of order holon with other holons to drive production

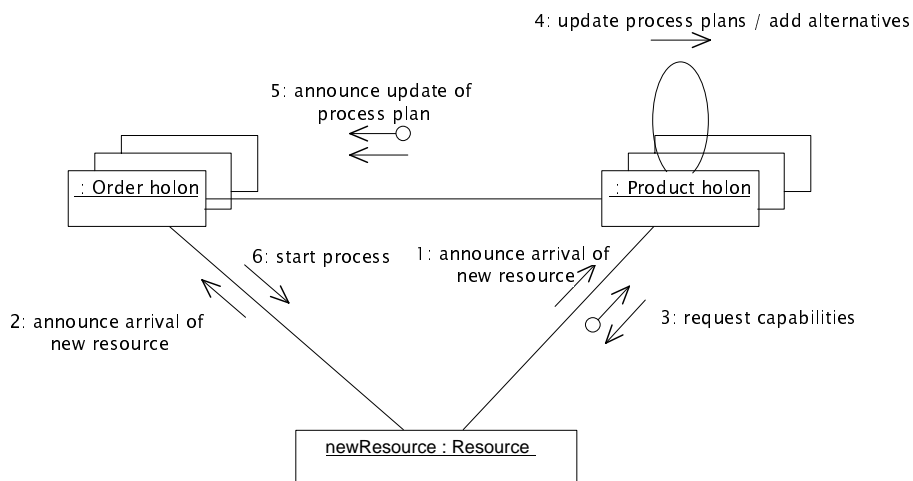


Figure 9: Adding a resource holon triggers an update of the process plans in the product holons

6. Self-similarity in holonic systems

Self-similarity in PROSA is an important characteristic which partly determines the reconfigurability of the control system. Homogeneous system components reduce the complexity of the overall system, which simplify the development and integration of new holons into the system. The HMS architecture contains self-similar components, meaning that holons of the same type have similar interfaces and similar behaviour. This allows holons to be internally different, while not imposing additional complexity to the others.

The HMS architecture shows “horizontal” and “vertical” self-similarity. Horizontal self-similarity relates to self-similarity across different specialisations on one level of aggregation. Vertical self-similarity refers to self-similarity across different levels of aggregation: higher-level resource holons work similar to lower-level resource holons.

Note that the term self-similarity is often used in relation with fractal structures. The presented HMS architecture is a fractal system in the original meaning of the word. A similar fractal structure, but then applied to the factory on an organisational level, is also proposed in the fractal factory as described by Warnecke [3].

6.1 Horizontal self-similarity

Using the specialisation mechanism described in section 3.3, the HMS architecture models the heterogeneous nature of a manufacturing system in a homogeneous way. All order holons, product holons and resource holons are self-similar across different specialisations.

All holons inherit, in the object-oriented sense, a common interface and behaviour from the basic resource, product, or order holon (see Figure 4). So, a holon which communicates with an order holon can always use this common interface and it does not need to be aware of the fact that this order might be a special kind of order holon (e.g. rush order). The diversity is expressed by adding aspects to the interface and to the behaviour. The same applies to product holons and resource holons.

All holons have at least the data and functionalities which are given in Figure 6. Therefore, the resource holons have at least the following functions in common: query the list of capabilities of the resource (list of its product holons), the list of running tasks, the list of sub-resources, request the activity log, query the status of a running task, and functions to start and stop the production of a product. Whether the resource holon is a conveyor holon, robot holon, or NC-machine holon, they all have this interface. Each one of these more specialised holons may add aspects to the interface and to the behaviour. For instance, a robot holon could add interface functions for configuring the workspace, while these functions do not make sense for a conveyor holon.

The next sections give examples that illustrate the advantage of this self-similarity: special cases can be handled similar to the nominal case. The natural heterogeneity of manufacturing system components is reduced by modelling them as homogeneous as possible. The main interface functions are common to all holons.

Heterogeneity of orders

Most shop floor control systems are designed to handle standard orders only, and need to be bypassed to produce orders which require different logistical or technical treatment. A well known example is a rush order with an early due date, which should get priority over make-to-stock orders.

Another example is a so called first-of order which is responsible for producing the first (set of) instances of a new product. In most factories, the product model and process plans still have to undergo a final validation phase. Because of unreliable process plans, such first-of order requires longer operation times, possibly needs repeating of failed processes, and needs human supervision and human intervention.

Although this first-of order holon has special characteristics, the other holons in the system treat it just as a normal order holon. The non-validated product holon will specify in its process plans that it needs additional resources (e.g. a human supervisor), and more time to be produced (e.g. because of running the processes at lower speed). The order holon will take this into account when generating the schedule. The resource holons produce according to the specifications of the process plan, and are unaware of the fact that this is a special order.

Heterogeneity of products

Product heterogeneity is present in almost all factories. Factories usually produce several products: variations of the same product, a new generation of a product, spare parts, high-volume mass production, high-quality product, etc.

Suppose a factory makes a product A which represents 90% of the factory throughput, and products B1, B2, B3, etc. are low-volume, high-precision products which allow a high profit margin and give a strategic advantage over competitors. The product holon of product A focuses on efficient use of resources. This implies process plans which specify optimal cutting speed, foresee in alternative operations which can be used in case of a malfunctioning resource, and minimise the need for human assistance during processing. On the other hand, the product holons for B's focus on precision and reliability. This results in process plans which specify high tool accuracy, safe cutting speeds, and include no alternative operations because this could lead to quality variation.

Both product holons are by their nature totally different, they have different representations of process plans, and different requirements. Even though, the order holons and resource holons do not need to adapt their way of using these product holons. The order holons get a list of possible next steps to execute, and the resource holons get specifications on how to execute their process.

Heterogeneity of resources

The heterogeneity of resources is the most obvious one. The HMS architecture provides a homogeneous interface to all these resources. Several resource holons may be able to produce the same product holon. However, internally these resource holons may apply different processes to produce this product holon.

For instance, a 3-axes milling workstation holon and a 5-axes milling workstation holon are both capable of milling prismatic parts [11]. They will however generate and execute different NC programs to mill such part. Another example is a conveyor belt holon, an AGV-fleet holon, and a human worker holon, which are all capable of transporting a pallet from one location to another location. The time it takes to perform a specific transport job, the price, and the availability of each of the transport resources are different. These parameters are taken into account by an order holon to select one of the transport resources to perform the job. However, the order holon does not need to know how this transport job will be carried out to be able to select a resource.

6.2 Vertical self-similarity

[14] mentions the concept of a “whole” which becomes “part” of a bigger entity: holarchies become holons in a higher holarchy. The holonic factory is itself a resource holon which can be part of an holonic extended enterprise. On the other hand, the resource holons of the factory can be holarchies themselves. The HMS architecture is applicable to any aggregated resource mentioned in section 3.2, it can be applied to a factory, a shop, a workstation, etc.

This self-similarity delivers homogeneity in the internal operating of holarchies on different levels of aggregation. As a consequence, reconfiguring the system becomes easier because a resource holon can be integrated in any holarchy without needing to change the functioning of the resource holon, nor the functioning of the holarchy. If the other holons want to use this newly added resource holon, they need to update their model of the holarchy, their process plans, and their decision rules accordingly.

Since a resource holon does not need to be adapted to a specific functioning of a holarchy, it may be member of several holarchies at the same time. The traditional hierarchical levels in a factory do not need to be respected: an equipment holon can directly belong to the factory holarchy, as well as to a workstation holarchy (see Figure 3). Especially for the human resource holon, which is a resource with a broad functionality, this belonging to multiple holarchies, even at multiple levels of aggregation is a desirable characteristic. The same human can be member of a workstation holarchy to clamp the workpiece, while this same human worker can also belong to a transport holarchy for transporting pallets, and to the factory holarchy to give training to new machine operators.

Note that this vertical self-similarity is only illustrated for resource holons. This is because the HMS architecture is designed for production systems, which are by their nature always resources. Whether this HMS architecture of order holons, product holons and resource holons can also be applied to the internal working of a product holon or of an order holon is an issue of further debate.

7. Different viewpoints on the basic holons

From previous presentations and discussions about PROSA with other researchers, we learned that, depending upon their personal manufacturing background, the fundamentals of the described architecture may be perceived differently. Depending on the reader’s viewpoint, the basic holons have a different ontology. From some viewpoint, a specific holon seems to be a passive information server. Another viewpoint focuses on the way this holon generates and maintains the information, so the same holon is known to be a complicated component which requires contributions from several research communities.

This section alerts the reader to this fact by explaining explicitly the four most common viewpoints taken by people with a manufacturing background: the external viewpoint, the process viewpoint, the manufacturing control viewpoint, and the concurrent engineering viewpoint. Other viewpoints are possible but are outside the scope of this paper. For instance, a computer scientist may take a multi-agent system viewpoint trying to analyse and design the communication and co-operation aspects in the HMS.

7.1 External viewpoint

This viewpoint looks from outside at a HMS and its components. This is the viewpoint of the customer, and of the factory manager, who wants to use the holarchy to produce goods. The customer needs to know which goods he can order. The factory manager also want to know which goods are currently

being produced, and which resources are used. This viewpoint does not cover the internal operation of the holonic system.

The product holons represent things which can be made by the holarchy. The customer orders one or more products which gives rise to an order holon to fulfil this task. The order holon triggers the resource holon to start production. If the HMS represents the entire factory, the product holons refer to items present in the company's catalogue, the order holon refer to items present in the order book, and the resource holons represent the departments, shops, machines, people, etc. If the holarchy represents a shop, the product holons represent half-finished products made by the shop. If the holarchy is a workstation, the product holons represent operations that can be performed by this workstation. Order holons represent tasks to create these products, and the resource holons represent resources which may be used to perform these tasks.

From the external viewpoint, the holonic manufacturing system contains product holons for the products it can make, order holons for the tasks it is performing, and resource holons for the resources it may use to perform these tasks.

7.2 Process viewpoint

This is the viewpoint of people focusing on production processes, process control, process parameters, process monitoring, quality measuring, etc. Given a certain description of a product to make, they are interested in controlling the process such that the best performance is achieved with respect to process gain (e.g. cutting speed) and process quality (e.g. cutting surface quality). This viewpoint does not incorporate the logistics, nor the existence of multiple products, multiple process plans for a product, or multiple resources which can perform the process.

Seen from this viewpoint, the order holon triggers the resource holon when to start executing a specific process. The product holon is an information server which sends the correct process description, parameters and constraints to the resource holon. The resource holon executes its task with the best possible performance. In the process viewpoint, most research is situated in the resource holon wanting to push the limits in executing the task (e.g. higher cutting speed, better surface quality, more accurate welding), or developing new processes to execute a task (e.g. rapid-prototyping). The product and order holon are considered to be a database of task descriptions and a trigger signal to start a task.

For example, for a feature based milling process as described in [24], the product holon specifies the feature to be processed: pocket shape, size, and tolerances. The resource holon executes this task as well as possible by doing a proper task refinement, selecting appropriate tools and cutting speeds. The resource holon carries out a tool length measurement and corrects its cutting paths correspondingly in order to get more accurate results. Additionally, the resource monitors tool breakage and vibrations in order not to damage the workpiece when something goes unexpectedly wrong.

7.3 Manufacturing control viewpoint

The manufacturing control viewpoint is generally the viewpoint taken when focusing on scheduling, balancing of flows, shop floor control, multi-agent negotiation, contract-net protocol for resource allocation, reactive scheduling, disturbance handling, etc. Given a number of tasks and a set of resources, the aim is to assign tasks to the resources in order to optimise certain performance criteria like throughput, work-in-process, lead-time, or tardiness. This viewpoint does not consider aspects of process control, or process planning.

The product holon is an information server which gives the order holon a list of possible tasks to execute and a list of resources on which these tasks can be executed. The order holons query the resource holons for available capacity. Next, the order holons negotiate among themselves on the allocation of the available resource capacity to the tasks. For this scheduling task, they may call in the advice from a centralised scheduler staff holon as described in section 3.4.

For example, several order holons are launching tasks on resources. The resource holons report back the progress of executing the task. If a resource breaks down, it informs the order holon for which it was working. This leads to a re-negotiation among the order holons so that the most important orders can continue, and the disturbance only affects the less important orders.

7.4 Concurrent engineering viewpoint

Product development, production preparation, process planning with alternatives, etc. are parts of the concurrent engineering viewpoint. This viewpoint is concerned with generating recipes on how to use the resources to create products which maximally satisfy the customer and put minimal constraints on

the resource usage. The first goal refers to the fact that the system should generate a product which is desired by the customer, including aspects like tolerances and quality specifications. The second goal refers to representing the product design and production plan such that a maximal set of production options stay open. This allows the order holons to use alternative resources or alternative processes to optimally balance the production flow.

The resource holon provides the product holon with information about available processes and process capabilities. The order holon informs the product holon about the result of processes executed on the resources. It is the order holon which holds the state information of the physical parts being produced, but it is the product holon which informs the order holon of what processes may be executed next. The product holon may give the order holon different information, depending on the result of previous production steps and the status of the resources.

For example, a product holon may internally work with a non-linear process plan, which takes into account alternative operations [9]. If the order wants to execute an operation on a resource, but the resource is temporarily unavailable, the order will ask the product holon for another operation which it can execute in the mean time.

8. Conclusions

The HMS architecture PROSA consists of three types of basic holons: resource holons, product holons and order holons. Every basic holon type focuses on different responsibilities of the manufacturing system. The holons exchange process knowledge, production knowledge, and process execution knowledge respectively. Aggregation is used to focus on different levels of holons. Specialisation is used to focus on different functionalities of holons. Staff holons are optional elements which may assist the basic holons in performing their task. Staff holons permit the incorporation of centralised solutions. This is useful for problems for which there does not exist a distributed solution, and it allows easy migration from current hierarchical systems to holonic solutions. Since staff holons are only giving advice to the basic holons, they do not introduce hierarchical rigidity into the system. A descriptive model of the data, functions, and behaviours of the holons, shows more detailed aspects of the architecture.

After comparing PROSA with existing manufacturing control approaches, it is concluded that PROSA is covering all aspects of both hierarchical and heterarchical control architectures. As such, PROSA can be regarded as a generalisation of the two former approaches. More importantly, it also introduces significant innovations: the system structure is decoupled from the control algorithm, logistical aspects can be decoupled from technical ones, and , and PROSA allows to incorporate more advanced hybrid-control algorithms.

The resulting architecture has a high degree of self-similarity, which reduces the complexity to integrate new components and enables easy reconfiguration of the system. Due to the horizontal self-similarity, special cases of orders, products, and resources can be handled similar to nominal cases. The vertical self-similarity avoids the need for strict hierarchical levels. It allows for individual resource holons to belong to several holarchies at several hierarchical levels.

The last section emphasises the influence the reader's personal background has upon the understanding of PROSA. When looking at this HMS architecture from different viewpoints the holons have a different ontology. The four most common viewpoints are the external viewpoint, the process viewpoint, the manufacturing-control viewpoint, and the concurrent-engineering viewpoint.

Future work needs to focus on establishing a methodology to go from this generic reference architecture to a specific system architecture for one specific manufacturing system. A further research concern of great importance is the development of the holarchy meta-controller, containing the "rules of the game."

9. Acknowledgement

This paper presents research results obtained through work sponsored by the Concerted Research Action (GOA) on holonic manufacturing, supported by the Office of the Prime Minister, Science Policy Programming, through the K.U.Leuven Research Council.

10. References

- [1] N. Okino, "Bionic Manufacturing Systems", in *Flexible Manufacturing Systems, Past, Present, Future*, Ed. J Peklenik, CIRP, Faculty of Mechanical Engineering, Ljubljana 1993, pp. 73-95.
- [2] K. Ueda, "A Genetic Approach toward Future Manufacturing Systems", in *Flexible Manufacturing Systems, Past, Present, Future*, Ed. J Peklenik, CIRP, Faculty of Mechanical Engineering, Ljubljana 1993, pp. 221-228.
- [3] H-J. Warnecke, *Die Fraktale Fabrik, Revolution der Unternehmenskultur*, Springer, Berlin, 1992 (in German).
- [4] K. Iwata, M. Onosato, "Random Manufacturing System: a New Concept of manufacturing systems for production to order", *Annals of the CIRP*, Vol. 43/1/1994, 1994, pp. 379-384
- [5] F. Kimura, "A Product and Process Model for Virtual Manufacturing Systems", *Annals of the CIRP*, Vol. 42/1/1993, 1993, pp. 147-150.
- [6] H. Van Brussel, "Holonc Manufacturing Systems, the vision matching the problem", Proc. of First European Conf. on Holonic Manufacturing Systems, Ed. IFW-Hannover, Hannover, December 1, 1994.
- [7] P. Valckenaers, H. Van Brussel, L. Bongaerts, J. Wyns, "Results of the Holonic Control System Benchmark at the KULeuven", Proc. of the CIMAT Conference (Computer Integrated Manufacturing and Automation Technology), 10-12 October 1994, Troy, NY, Rensselaer Polytechnic Institute, pp.128-133.
- [8] P. Valckenaers, *Flexibility for Integrated Production Automation*, PhD. thesis K.U.Leuven, 1993.
- [9] J. Detand, *A Computer Aided Process Planning System Generating Non-Linear Process Plans*, PhD Thesis, KU Leuven, 1993 (ISBN 90-73802-23-7).
- [10] L. Bongaerts, P. Valckenaers, H. Van Brussel, J. Wyns, "Schedule Execution for a Holonic Shop Floor Control System", Pre-prints of the Advanced Summer Institute (ASI) 95 of the N.O.E. on Intelligent Control of Integrated Manufacturing Systems, Lisboa, Portugal, Ed. Alfamicro, June, 25-28, 1995
- [11] P.I. Tanaya, J. Detand, J.P. Kruth, "Holonc Machine Controller: a study and implementation of holonic behaviour to current NC controller", Co-operation in Manufacturing: CIM AT WORK (Preprint), Kaatsheuvel, The Netherlands, August 28-30, 1995, pp. 375-396.
- [12] H. Van Brussel, P. Valckenaers, F. Bonneville, "Programming, Scheduling and Control of Flexible Assembly Systems", *Manufacturing Systems*, Vol. 23, No. 1, 1994, pp. 25-36.
- [13] J. Wyns, H. Van Brussel, P. Valckenaers, L. Bongaerts, "Workstation Architecture in holonic manufacturing systems", Proc. of the 28th CIRP International Seminar on Manufacturing Systems, May 15-17, 1996, Johannesburg, South Africa, Ed. Z Katz, Rand Afrikaans University, pp. 220-231.
- [14] A. Koestler, *The Ghost in the Machine*, Arkana Books, London, 1989 .
- [15] D.M. Dilts, N.P. Boyd, H.H. Whorms, "The Evolution of Control Architectures for Automated Manufacturing Systems", *Journal of Manufacturing Systems*, Vol. 10, N. 1, 1991, pp. 79-93.
- [16] L. Bongaerts, J. Wyns, J. Detand, H. Van Brussel, P. Valckenaers, "Identification of Manufacturing Holons", Proc. of the European Workshop for Agent-Oriented Systems in Manufacturing, Berlin, Germany, 26-27/9/1996, Eds. S. Albayrak (TUBerlin) and S. Bussmann (Daimler-Benz AG), pp. 57-73
- [17] Bauer, Browden, Browne, Duggan, & Lyons, *Shop Floor Control Systems - From design to implementation*, Chapman & Hall, USA, 1991.
- [18] Rumbaugh, Booch, & Jacobsen, *The Unified Modelling Language (UML) ver. 1.0*, Rational Co., California, USA, 1997.
- [19] G. Booch, *Object Oriented Design with Applications*, Bejamin/Cummings, Redwood City, California, USA, 1991.
- [20] J. Rumbaugh, Blaha, M., Premerlani, W., Eddy, S., & Lorensen, W., *Object Oriented Modelling and Design*, Prentice Hall, Englewood Cliffs, NJ, USA, 1991.
- [21] M. Waldrop, *COMPLEXITY, The emerging science at the edge of order and chaos*, VIKING, Penguin group, 1992.
- [22] J. Hommes, T. Geraads, *Van arbeidskunde naar bedrijfskunde*, H.E. Stenfert Kroese BV, Leiden-Antwerpen, 1980 (in Dutch).
- [23] L. Bongaerts, L. Monostori, D. McFarlane, B. Kádár, "Hierarchy in Distributed Shop Floor Control", Proc. of the First International Workshop on Intelligent Manufacturing Systems IMS-EUROPE 1998, Ed. EPFL, Lausanne, 1998, pp.97-113.

- [24] J-P. Kruth, J. Detand, P.I. Tanaya, T. Van Ginderachter, J. Wyns, "An NC Holon architecture", Proc. of Machine tool national conference CNMU96, Bucharest, Romania, Ed. Politehnica University of Bucharest, October 24-25, 1996.
- [25] L. Bongaerts, H. Van Brussel, P. Valckenaers, "Schedule Execution using Perturbation Analysis", Proc. of the IEEE International Conference on Robotics and Automation (ICRA98), Leuven, Belgium, 1998.
- [26] M.K. Senehi, T.R. Kramer, S.R. Ray, R. Quintero, J.S. Albus, "Hierarchical control architectures from shop level to end effectors", in *Computer control of flexible manufacturing systems*, eds. Sanjay B. Joshi and Jeffrey S. Smith, Chapman & Hall, 1994, pp. 31-60.
- [27] P. Gullander, P. Klingstam, P. Martensson, "Generic modelling of cell control systems and production resources", Proc. of the 7th Int. Conf. on Flexible Automation and Intelligent Manufacturing (FAIM'97), University of Teesside, Middlesbrough, UK, 1997.
- [28] A.L. Arentsen, *A generic architecture for Factory Activity Control*, PhD thesis, University of Twente, Enschede, 1995.
- [29] N.A. Duffie, R. Chitturi, J. Mou, "Fault-tolerant Heterarchical Control of Heterogeneous Manufacturing System Entities", *Journal of Manufacturing Systems*, Vol. 7, No. 4, 1988, pp. 315-327.
- [30] N.A. Duffie, V.V. Prabhu, "Real-Time Distributed Scheduling of Heterarchical Manufacturing Systems", *Journal of Manufacturing Systems*, Vol. 13, No. 2, 1994, pp.94-107.
- [31] J. Hatvany, "Intelligence and Cooperation in Heterarchic Manufacturing Systems", *Manufacturing Systems*, Proc. of the CIRP Seminars, Vol. 14, No. 1, 1985
- [32] R.G. Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver", *IEEE Transactions on Computers*, Vol. C-29, No. 12, 1980, pp. 1104-1113.
- [33] R.G. Smith, R. Davis, "Frameworks for Cooperation in Distributed Problem Solving", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-11, No. 1, 1981, pp.61-70.
- [34] N.A. Duffie, R.S.Piper, "Nonhierarchical Control of Manufacturing Systems", *Journal of Manufacturing Systems*, Vol. 5, No. 2, 1986, pp. 137-139.
- [35] G.Y. Lin, J.J. Solberg, "Autonomous control for open manufacturing systems", in *Computer control of flexible manufacturing systems*, eds. Sanjay B. Joshi and Jeffrey S. Smith, Chapman & Hall, 1994, pp. 169-205.
- [36] F.P.M. Biemans, *A Reference Model for Manufacturing Planning and Control*, PhD thesis, University of Twente, Enschede, 1989.
- [37] U. Rembold, B.O. Nnaji, A. Storr, *Computer Integrated Manufacturing and Engineering*, Addison-Wesley, 1993.
- [38] N. Suh, *The principles of Design*, Oxford University Press, New York, 1990.
- [39] S.E. Ramaswamy, S.B. Joshi, "Distributed Control of Automated Manufacturing Systems", Proc. of the 27th CIRP Seminar on Manufacturing Systems, Michigan, 1995, pp. 411-420.
- [40] J. Wyns, P. Valckenaers, H. Van Brussel, "Resource allocation in the holonic manufacturing system at K.U.Leuven", Proc. of Conf. on Control of Industrial System, Belfort, France, 1997, Vol 3, pp. 353-358.
- [41] J. Wyns, P. Valckenaers, H. Van Brussel, L. Bongaerts, "Implementation of Resource Allocation in the Holonic Workstation Architecture", Proc. of the 1st Europe-Asia congress on Mechatronics, Besancon, France, October 1-3, 1996, pp. 680-685.
- [42] J. Wyns, T. Van Ginderachter, P. Valckenaers, H. Van Brussel, "Integration of resource allocation and process control in holonic manufacturing systems", Proc. of the 29th CIRP International Seminar on Manufacturing Systems, May 11-13, 1997, Osaka, Japan, pp.57-62.