ON RESOURCE SERVICE ACCESS IN A HOLONIC MANUFACTURING SYSTEM

BY

DORU PĂNESCU and CARLOS PASCAL

Abstract. This paper treats some problems concerning the access to the resources when the holonic approach is used in manufacturing systems. The proposed solution makes use of a separate component that should solve distinct problems on accessing the services provided by the operational units existing in a manufacturing environment. It solves the configuration phase and then ensures the information collecting as needed for system monitoring and evaluation. The interfaces integrated with the respective component are described, too. For the holonic architecture performance analysis, a Petri net based model is used. The conclusions regard the way a centralized component added to a heterarchical manufacturing control scheme can determine certain advantages.

Key words: holonic manufacturing system, configuration, resource access, Petri nets.

2000 Mathematics Subject Classification: 68T42, 93C5; 93C65.

1. Introduction

Present manufacturing systems require higher and higher performance of the control systems as regards the adaptability to the market changes, just in time operation and quality standards. Holonic Manufacturing Systems (HMSs) represent a control architecture that aims at satisfying these characteristics, by making use of certain ingredients: a heterarchical structure, the use of autonomous, intelligent and cooperative components – the holons, and specific implementation approaches, mainly derived from multi-agent systems [1],...,[3]. A frequently used mechanism in the holonic method is a service based operation: the resources available for the manufacturing process provide various services and through an appropriate coordination scheme the right combination
is to be chosen in order to satisfy the system’s goals. This kind of functioning implies specific aspects and this contribution regards a methodology to deal with some of them.

Both at the initializing time and then in operation the HMS can be supported by a distinct component facilitating the access to the services offered by holons. Though many holonic approaches suppose a service based structure, they do not give the design and implementation details, but only refer to a standard architecture (e.g. CORBA [4], [5]). Thus the proposed solution is to include a distinct part into the HMS, named Resource Service Access Component (RSAC) that should be a service oriented constituent, too. The RSAC has to cope with: the users’ access to the HMS (allowing goals and specification forwarding, the interaction with the holonic agents’ knowledge bases), the initial configuration, keeping a model of the HMS’s services and some analyzing facilities. To obtain all these, the RSAC must have the capacity to monitor and collect the needed information, by making the necessary connections with the other entities of the HMS. In the remainder of this paper further details are given regarding the RSAC integration into the holonic structure and its operation, with a focus on the configuration phase. Moreover some aspects regarding the HMS modelling and analysis facilitated by the RSAC are discussed.

2. The Resource Service Access Component Role and Integration

The way the new proposed component can be integrated into the holonic architecture is explained in Fig. 1. The connections must be made with a few types of entities; besides the holons (these are of three kinds – order, product, and resource holons according with the most often used holonic methodology, the so called PROSA architecture [6], [7]) the driver applications and the HMS infrastructure have to be considered, too. These are the software components that interface the physical operational devices (e.g., robot controllers, PLCs) with the agents that form the deliberative part of the corresponding resource holons [2], [8]; this means that, for example, a robot resource holon contains the physical manipulator and its corresponding controller that is coupled by the means of an application driver with an agent representing the robot in the HMS. The driver application’s role is to abstract and provide services for the external control of the physical part.

In Fig. 1 the thin arrows indicate the connection of various entities to the services provided by the holonic infrastructure, connection that is necessary so that a component should be able to publish the services it offers, while the thick arrows designate the services’ use. At the initial configuration time the names and addresses of the infrastructure services must be provided. Then the RSAC is connected to the infrastructure and it can access the services of the other holonic parts. After that moment, the main purpose of the RSAC is to
collect information from the holons and driver applications, which may be
delivered at the operator’s request and used in the HMS performance analysis. It
thus results that the RSAC plays its role at certain phases of the HMS
functioning: at the initial configuration stage, when a dynamic re-configuration
is needed (the holonic structure can provide an autonomous re-configuration
mechanism when necessary – for example, at a device fault [9]), and when the
user asks for assistance. These situations are described in the next sections.

3. The Holonic System Configuration

The correct service oriented operation of the HMS is highly dependent
on the proper initial configuration, when the RSAC has to be involved. The
importance of this operation is increased by the way the holonic structure has
not an a priori settled layout, but this can vary according to the resources being
available. In fact, this is one of the advantages of the HMS that is created by the
holonic loose organization, which further determines the need of a specific
configuration phase. Thus the HMS formation by incorporating the existing
manufacturing parts is to be conducted according to the three levels of
configuration shown in Fig. 2, taking into consideration the following steps:

Fig. 1 – The RSAC integration in the HMS using existing services.

Fig. 2 – Levels of configuration for the holonic structure.
1. the holonic infrastructure configuration (level 0);
2. the resource holons’ configuration that supposes two distinct stages:
   − establishing the configuration parameters for the driver applications
     (level 1);
   − settling the configuration parameters for the communication between
     the holonic agents and their corresponding driven physical devices (level 2);
3. the configuration for the holons’ integration, a process that further
   includes the product and order holons as a stage of the level 2 of configuration.

The order for different configuration operations is marked in Fig. 2
through the numbers attached to the arrows. As about the first level of
configuration (level 0), the holonic infrastructure comprises all the hardware
and software entities and their connections. In an HMS this includes the
computer networks, various industrial controllers, the operational parts
(manipulators, machine tools, conveyors, etc.) and the attached software
modules (both common industrial software components, and specific holonic
units – e.g., the components needed for the agent’s creation). In the holonic
approach this infrastructure can be locally formed (e.g. for a single
manufacturing cell) and then the process is repeated in a recursive manner for
higher levels (for example including several cells in a manufacturing
department). This is a benefit of the recursive character of the holonic
organization. Besides the configuration tasks common for an industrial
computer network, in an HMS the primary holonic structure must be configured
and started. As it is a service based approach, it supposes the initiation of
different services; in the CORBA scheme, the naming, trader and event services
must be established [4]. These services are configured and initiated on a
networked computer, which then can offer configuration parameters for other
services and levels. A compulsory set of services regards those needed for the
agents’ knowledge bases updating.

Another distinct operation of the infrastructure configuration concerns
the establishing of the absolute spatial and temporal references. These are
needed because the HMS is a distributed system, including holons that have
only partial knowledge on the entire environment. Though the holons are
characterized by autonomy, the manufacturing goals have to refer a unique
reference and this must be known even by the low level holons in a holarchy,
namely the resource holons. Thus, in the infrastructure configuration phase a
reference frame is established that will be used to assess the positions and
orientations of the various resources (robots, conveyors, storage devices). Afterward
this reference frame has to be used as a configuration parameter for
the resource holons of the above mentioned devices, when the level 2 of
configuration is done. A temporal reference is to be also settled because the
common case is when the manufacturing goals have attached specific time
constraints, which can be managed only by using such a reference.
As about the level 1 of configuration, because it implies the driver applications, two types of operations are necessary: first, the configuration parameters for the connection between the driver application and the physical device it serves are needed, and second, the integration of the driver application into the holonic structure is to be obtained (this reflects the double connection of the driver application). The first operation is dependent on the way the physical entity allows to be connected and can interact; for example, a driver application communicates with an industrial robot controller though a specific protocol. Then, as a second operation, the information of the level 0 of configuration (the names and addresses of the necessary services) is used to incorporate the application driver into the holonic scheme by publishing its own services. Besides the services allowing the physical device configuration and control, the driver application must dedicate a service for the configuration of the communication channel. With this service a user can access an interface in order to be able to modify certain parameters of the driver application. In this way, it results that in the holonic scheme the driver application can be in one of the following states:

− initial configuration: it is the state allowing the connection process towards the physical device and the publishing of the services for the piece of equipment control;
− active: it is the state when the driver application is ready to accept the connection towards the holonic agent representing the deliberative component of the corresponding resource holon, being prepared to offer dynamic (on-line) resource configuration services;
− agent-active: it is the state reached after an agent is connected to the driver application and is able to control the resource;
− inactive: it is a state when the driver application suspends the services regarding the physical device, while its own configuration services continue to be active.

It is to note that when the driver application is in the agent-active state part of the configuration services are hidden towards the operator, in order to prevent the conflicts with the agent’s actions; even so, one can use the level 2 of configuration to indirectly interact with the physical entity.

As about the level 2 of configuration it benefits of the previous two levels allowing the holons’ formation and their integration in the holonic architecture. Considering first the resource holons, they require two types of configuration operations: the establishing of the intra and inter holonic communication [5], and respectively the setting up of the initial facts on the driven physical device and its environment, as needed for the agent’s knowledge base. By that time, the driver application which is already configured supports the intra-holonic link between agent and resource. Thus, after the level 2 of configuration completion even a dynamic (on-line) resource
configuration can be done by the HMS when needed, through the corresponding holonic agent. The other type of configuration, regarding the agent’s knowledge base filling in and updating, is based both on general reference information (spatial and temporal) and specific resource data (e.g., a robot working area and payload, the contents of a storage device). By conducting the configuration process in this way an indirect benefit is obtained: an agent pattern can be reused or cloned when a further device of the same type is added (a new robot can use the agent pattern from a previous robot resource holon).

After these configuration phases the resource holons’ agents publish their services, among them two being mandatory: a dynamic configuration service that allows an on-line configuration of the holonic agent and the driver application, and a service for publishing the holon’s state. To conclude, as for the driver application, one can remark the states of the resource holon’s agent:

− initial state, when the connection between the agent and the driven physical device is established and the holonic agent’s knowledge base is updated using the results of the low levels configuration phases;

− configuration state, when the operator is allowed to modify the configuration parameters of the agent and its associated device;

− simulation state, when the agent offers its services only towards the operator so that this can make tests on the decisional holon’s functioning;

− active state, when the agent is fully active so that all the holon’s capabilities are operational into the HMS.

The agent enters the configuration or simulation states at the operator’s request. If such a demand appears when the holonic agent is within the active state, then the request is regarded as a goal, as the only possibility of interaction with an active agent of the HMS. In consequence, as usual for the goals’ treatment, the agent will switch its state only after completing the current goal.

For the product and order holons, because they are not connected with a physical entity, the configuration is simpler. The configuration parameters regarding the holonic agent operation and its knowledge base are treated in the same way as for the resource holons. For example, in the case of a product holon an important aspect of the configuration phase (comprised in the level 2 of configuration) refers the updating of the agent’s knowledge base with the production plans. For the order holons a common issue of the configuration phase is the connection with a user interface allowing the operator to transfer goals.

After a holon is introduced and configured in the HMS, and subsequent to its services publishing (which is implicitly carried out in the configuration phase, as already explained), it waits for those manufacturing goals which it can be involved in. The various services provided by different holonic agents will be used in the goals’ management and solving. About the process of holonic services’ backing off or alteration this must follow a certain sequence.
the holon (through its agent) should withdraw or modify the service and then accordingly update its bids in the current cooperation activities (this are handled according with the Contract Net Protocol [10], [11]). Both at the configuration stage and in operation the RSAC needs certain interfaces, described in the next section.

4. Interfaces of RSAC

The RSAC contains a collection of interfaces which enable an operator to monitor and interact with the HMS’s components. It is the role of the RSAC to centralize and group these interfaces and expose them whenever the operator needs links to particular entities. The RSAC manages the selection of a proper interface and the connection process. This feature is enabled only for those components that have already published their services, specially the configuration one. As a consequence of this restriction, the configuration for starting up an entity in the HMS is not present in the entity’s interface. Thus, a driver application or an agent is included into the holonic structure using a configuration script that is running on the host computer. Three interface classes are illustrated below.

An interface of a driver application for an industrial robot appears in Fig 3. It contains means to connect the driver application to a robot controller (label 1 in Fig. 3), to interact with the selected controller by copying, running, stopping programs (label 4), open/close the robot’s motors (label 3), and change its gripper’s position/orientation (label 10). Moreover, it maintains and shows the controller and robot’s states (labels 2 and 7), the program variables (6), significant events (8), and also information about the driver application (9). Similar interfaces exist for the rest of the physical devices. It is clear that these interfaces depend on the services that the driver applications expose.

Distinct interfaces are used for different types of holons. Fig. 4 contains an appropriate presentation layer for an agent of a storage resource holon. In the considered manufacturing system [8], a storage device can contain a stack of pallets with raw and processed parts. At the configuration time, the user has the possibility through the interface to specify the corresponding driver application, to modify the agent’s knowledge regarding the contents of the storage device (label 7 in Fig. 4) or to view different information when the agent is active, such as: its internal events (8), state of its resource (labels 6 and 7), and the contents of the messages interchanged with other holons in accordance with the CNP coordination scheme (1, 2, and 3). Furthermore, the user can interact with the agent through the interface as being another agent, by proposing goals, selecting offers, according contracts (see labels 4 and 5). This feature is useful to analyze the agent’s behaviour when it receives a specific message; the obtained information depends on the services.
exposed by the agent. The interfaces of the product and order holons have similar characteristics, except for the resource information that is not exposed, because they control a holarchy.

In addition to the interfaces dedicated to various holonic entities, there is a primary RSAC interface (Fig. 5) that pictures the whole HMS’s state. Thus, on the left side the operator has information about the existing entities (holons and driver applications – marked with label 1 in Fig. 5); from here there is an option to open the interface for each entity and to access or attach some Petri net models (these will be briefly introduced in the next section). The information regarding the active/inactive services is present in the right side (label 2). The inactive services are those that were once available in the HMS; the RSAC keeps a history of their evolution and publishes their last state. In this way, an operator can track the system’s progress via this interface, which is updated by appropriate events (label 3 in Fig. 5), any information being directly obtained from the infrastructure components.
Fig. 4 – The resource holon interface for a storage device.

Fig. 5 – The main RSAC interface.
Some other interfaces of the RSAC (not detailed here) show useful information obtained after the corresponding processing, like: the services’ coverability, the collaborative planning process graph, the interaction diagram for a set of entities, and the generic models for specific holons. Some issues regarding these features are treated in the following section.

5. Models and Performance Analysis

Through the way of providing a collection of information from a distributed structure the RSAC and its attached interfaces aim at obtaining features that are common to hierarchical structures. The RSAC takes certain messages exchanged between agents (it can be notified at the occurrence of specific events) and in this way it can provide a global analysis of the HMS’s performance (both in terms of reaching a global optimum and avoiding situations of malfunctioning), by making use of local pieces of knowledge. It is important not to turn the RSAC into a simple collector component that gathers all the events and corresponding messages, as the amount of information provided by the distributed system can increase a lot. Thus, the analysis made available by the RSAC is to be triggered at the operator’s request (using the RSAC interface) or at the occurrence of some events: a bottleneck, a planning process for a manufacturing goal that lasts too long (it can be caused by an infinite loop), a component’s failure. The information collecting and analysing is also launched at the moment when the HMS starts an autonomous, dynamic reconfiguration. In such a case the RSAC begins to follow the agents’ steps made in the reconfiguration process, for further examination.

The HMS is a goal-driven system and the RSAC can record the goals’ decomposition process, until the elementary operations are identified and ready to be done by the contractor holons. Some of the holons’ messages exchanged during this goals’ processing phase contain information about the real-time behaviour of the holonic system, and this can be further used in obtaining models of the HMS. One such model regards a holarchy formation, which is represented as a tree, having as the root node the manager holon that has started the planning process, while the intermediate nodes denote the contractor-manager holons, and the leaves are represented by those holons being only contractors (Fig. 6). This holarchy model is extracted from a detailed representation of the planning and coordination processes, including the possible contractors and subcontractors. A holon is represented by one or more nodes, due to the fact that more possibilities exist to reach the manufacturing goal. In Fig. 6, an arrow marks a possible collaboration and a filled node represents a holon with no offer. As Fig. 6 shows, two holarchies can be formed for the same manufacturing goal. The first is based on collaboration among the holons H₁, H₂, H₃, H₄, and H₆, and the second one is formed by the holons H₁, H₃, H₄, and H₆ (H₃ regards a distinct type of services, while H₂ and H₄ make
bids on the same sub-goal). In this case the holon $H_1$ has the main role to select one bid from $H_2$ or $H_4$, which determinates the final model of the holarchy. Such a graph lets us analyze the decisions made by each manager holon about its contractors’ bids. Thus, by using an apriori known manufacturing scenario, with a determined solution, this can be compared with the result obtained by the HMS, and the holonic scheme performance be appreciated.

![Fig. 6 – The model of a holarchy extracted from the coordination process.](image)

Another kind of model can be constructed at the holons’ level, by using Petri nets [12]. The present contribution considers such models to be connected with the RSAC, increasing its capabilities. Thus, each holon is represented by a generic model under the form of a Petri net (having the same pattern for all the types of holons – [12]), based on the states and events that appear when the holon activates according to the CNP. In the proposed model the Petri net positions are interpreted as agent’s states (idle, planning and execution) and the transitions define the message handling events – the agent receives or sends certain messages regarding goals, contracts, bids, etc. This generic model is illustrated in Fig. 7, where an ordinary Petri net appears in black colour, this being in fact modelled through a Coloured Petri net (CPN) [13]. Using a link between the RSAC and the CPN software environment used for the model construction [13] (this allows the model’s connection to external events), and a proper extension of the Petri net model (this is marked with the red colour in Fig. 7), it is possible to analyze each holon’s interaction behaviour. This coupling between the RSAC and the Petri net models supports the detection of the irregularities’ occurrence in the HMS, namely those that can cause abnormal operation. For example, when an agent is in the idle state and the RSAC indicates a goal sending event, then a monitor function of the CPN tool signals the irregularity in the agent’s behaviour, as a goal can be only sent from the planning agent’s state. This is happening when the agent contains some incorrectly designed plans or its planning process enters into an infinite loop caused by a certain combination of manufacturing goals [12].
Another analysis can be done on the holon’s active services without involving the interchanged messages. This implies only the initial information from each holon according with the type of services that it can offer in the HMS and the possible needed services at the time being. A comparison can be made having as result the services’ coverability, indicating if certain goals are not attainable at the actual state of the system.

It is to remark that by incorporation a centralized component, which is used in the configuration and analysis phases, certain weaknesses of the HMS can be first revealed and then lessened. For example, by using the RSAC and its interfaces a quantitative performance analysis was already obtained [14] regarding the number of sub-goals needed and interchanged between holons in order to solve a manufacturing order and reach the optimal solution.

6. Conclusions

This contribution regards a topical subject of the manufacturing control architectures, i.e. the resource utilization in a distributed environment. Considering as main points the holonic method and the service oriented operation, some aspects missing from literature are covered: the configuration process, an appropriate organization to facilitate the access to the resources’ services and the HMS’s performance analysis. As it appears in this paper these subjects are inter-related. Through a well designed configuration stage, that uses a well devised component – the RSAC – the services provided by the manufacturing resources can be efficiently combined in a heterarchical scheme.
as the holonic one, and then the same component can be productively used in system’s evaluation.

Without restricting the distributed character of an HMS and its holons’ autonomy, the RSAC as a centralized component is required to provide a way for accessing all the entities within the HMS. This is necessary for the initial configuration, when the user should benefit from a software component that can guide the initiation process and provide the needed connections. Then, at the system running the RSAC represents a means that distinguishes the HMS’ decisions for the operator. At critical moments, as for example when a reconfiguration process takes place, the RSAC can allow the switching between an autonomous and an operator’s guided operation.

The consideration of a single component gathering in a controlled manner the information needed to obtain the HMS model is another point supporting the proposed solution. It is clear that a complex manufacturing system possesses a hybrid character, as both continuous and distributed processes are being involved. This contribution pleads for a Petri net model, as this can be efficiently implemented and interfaced with different types of processes for an on-line analysis. In conclusion, the RSAC and the performance assessment it can offer represent a further step towards an efficient access to the services that the resources can provide in a manufacturing system.

Received: May 26, 2010

“Gheorghe Asachi” Technical University of Iaşi, Department of Automatic Control and Applied Informatics
e-mail: dorup@tulasi.ro

REFERENCES


**UNELE ASPECTE PRIVIND ACCESUL LA SERVICIILE RESURSELOR ÎNTR-UN SISTEM HOLONIC DE FABRICAȚIE**

(Rezumat)

Lucrarea tratează câteva probleme legate de accesul la resurse într-un sistem de fabricație atunci când se folosește o arhitectură de control de tip holonic. Soluția propusă este aceea de a utiliza o componentă distinctă, destinată rezolvării unor probleme specifice privind accesul la serviciile pe care le oferă compozițiile operaționale existente într-un mediu de fabricație. Această componentă soluționează faza de configurare, iar apoi asigură colectarea informațiilor necesare pentru monitorizarea și evaluarea comportării sistemului de fabricație. Sunt descrise interfețele utilizate, iar pentru analiza performanțelor se apelează la un model de tip rețea Petri. Concluziile obținute privesc felul în care adăugarea unei componente centralizate într-o schemă de control distribuită poate determina avantaje certe.