

Article

Not peer-reviewed version

The Methodology for Data Collection from a Holonic Manufacturing System

[Branislav Micieta](#), [Vladimira Binasova](#)^{*}, [Martin Gaso](#), Adam Kabat

Posted Date: 21 October 2025

doi: 10.20944/preprints202510.1597.v1

Keywords: data collection; holonic systems; process level; manufacturing systems; industrial engineering



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

The Methodology for Data Collection from a Holonic Manufacturing System

Branislav Micieta, Vladimira Binasova *, Martin Gaso and Adam Kabat

University of Zilina, Faculty of Mechanical Engineering, Department of Industrial Engineering, Univerzitna 8215/1, 010 26 Zilina, Slovakia

* Correspondence: vladimira.binasova@fstroj.uniza.sk; Tel.: +421-041-513 2727

Abstract

The main objective of the paper is to design a data collection and processing system for a holonic system as part of an intelligent manufacturing system. The purpose was to design and experimentally verify a communication interface capable of collecting, evaluating data and communicating with other holons. These holons represented information systems and process-level control systems. The defined algorithm for the design of data collection from a holonic manufacturing system was verified on selected parts of the manufacturing holonic system. The design of a manufacturing data collection system contains a comprehensive sequence of steps and a list of necessary supporting equipment. Partial objectives include an analysis of the methods of obtaining production data from centralized and distributed control systems, identification of differences in the method of collection and control of centralized and distributed control systems. The core of the paper is devoted to the identification of key functional parts of the systems and the design of requirements for information manufacturing systems, the design of holons and their information protocols in intelligent distributed manufacturing systems and in decentralized control systems. The main output is the design of an algorithm for collecting and evaluating data from holons, which also includes a unified communication protocol for a holonic production system and the technical and software equipment for solving the task.

Keywords: data collection; holonic systems; process level; manufacturing systems; industrial engineering

1. Introduction

Over the past two decades, intensive research has been conducted in both academia and industry to improve the performance of manufacturing systems and their ability to flexibly respond to changing customer needs [1–3].

In addition to traditional goals such as increasing productivity, improving quality, and reducing costs, new requirements for the functioning of these systems have recently emerged. These include, for example, the ability to respond immediately to disturbances, increased resilience to failures, and the ability to adapt hardware or software components to current conditions [4]. Activities associated with the collection of production data require knowledge and practical experience from several areas of automation and informatics [5,6]. It is important to understand the principle of operation, the functionality of existing systems and their role in the management system of an enterprise that is considering the deployment of these information systems to support efficient and flexible production. To aid understanding, it is necessary to explain the principles of obtaining data from a centralized or distributed management system.

The holonic management approach can subsequently use these parts of communication and data collection in principle [7]. It represents a different approach to evaluating and managing production,

which can use standard information or industrial protocols. The subject and content areas of the analysis of the current state in this paper are dedicated to (Figure 1):

Automation area - understanding the principles of operation of industrial communication standards [8], buses and control systems. Their possibilities and scope of optimal use for specific applications in conjunction with intelligent sensors, which are becoming increasingly affordable [9].

The area of data network design and engineering, topology and hierarchical arrangement of nodes. This information and knowledge will ensure optimal connection of information systems with production units not only within the company [10], but also from the point of view of collecting and sharing information between production layouts located at distances exceeding the range of the local data network.

The area of communication interfaces of information systems, their possibilities of use and their mutual integration. This is a rapidly growing area with the aim of making functionality and data available for cooperation with other systems in the network [11]. This area is described in more detail in the proposed methodology and addresses the design of communication between information systems together with the process level.

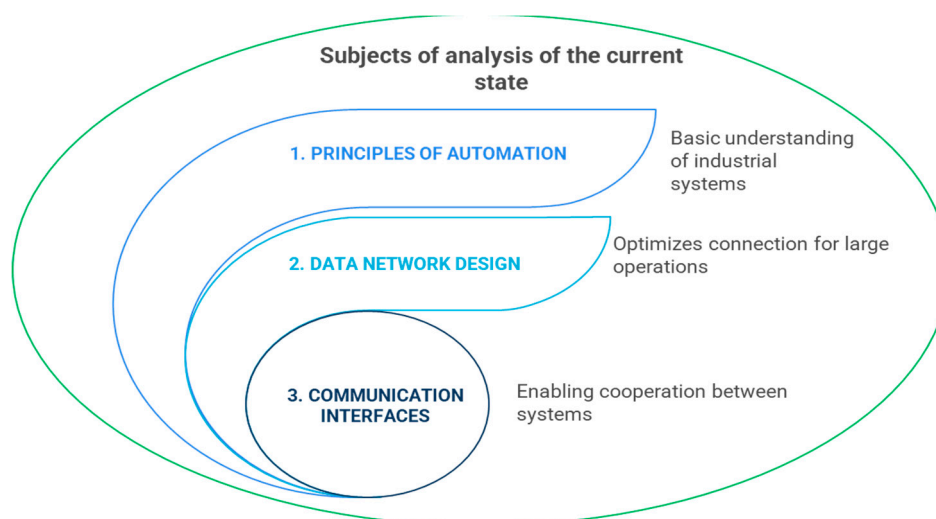


Figure 1. The subject and content areas of the analysis of the current state.

The idea of using the holonic concept in manufacturing systems emerged around 1990 in the IMS(s) program as a solution to cope with the increasing intensity of changes that were affecting the entire economic world, including the manufacturing sector. The HMS Consortium (HMSC), consisting of researchers from Australia, Canada, Europe, Japan and the United States, was established within the IMS program to develop tools for implementing the holonic concept in manufacturing areas and thus achieve the advantages of holonic organizational structures, such as "tolerance to failures, adaptability to changes and efficient use of available production resources" [12–15]. The HMS Consortium introduced a series of working definitions for individual entities of a holonic system:

1. A holon is an autonomous and cooperative block in a manufacturing system for the transformation, transport, storage and/or verification of information or physical objects. A holon consists of an information part and often also a physical, process part. A holon can be part of other holons [16].
2. Autonomy is the ability of an entity to create and manage the implementation of its plans and/or strategies.
3. Cooperation is the process by which a group of entities jointly develop acceptable plans and implement these plans [17].
4. Hierarchy is a system consisting of holons that can cooperate to achieve a desired goal or purpose (Figure 2).

5. A holonic manufacturing system is a holarchy that connects the entire range of manufacturing activities from order, through design, production, and marketing, to achieve an agile manufacturing enterprise [18,19].

Hsieh [20] defined a holarchy formation problem to lay a foundation to propose models and develop collaborative algorithms to guide the holons to form a holarchy that coherently moves toward the desired goal state with minimal costs.

Figure 2 illustrates a scenario in which a production process is to be formed in HMS with seven product holons h_1, h_2, \dots, h_7 and ten resource holons r_1, r_2, \dots, r_{10} to accomplish a task with timing constraints. Holonic processes are production processes dynamically created based on the collaboration of product holons. Each product holon has an internal process flow. Execution of the internal process of a product holon may rely on the outputs from the internal processes of one or more upstream product holons. For example, product holon h_5 and h_6 depends on either h_1 or h_2 to provide the type-one parts and depends on either h_3 or h_4 to provide the type-two parts. Product holon h_7 depends on either h_5 or h_6 to provide the type-three parts [20].

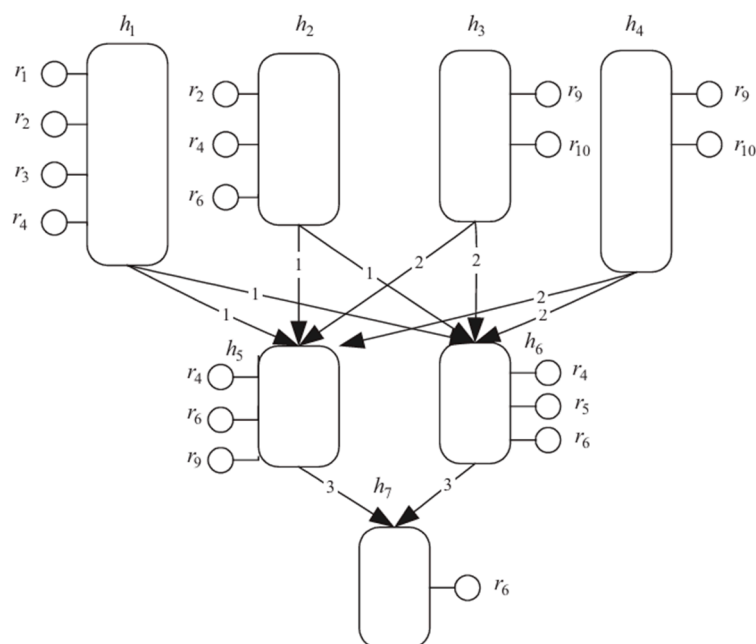


Figure 2. Holonic process formation [20].

A holonic system is composed of autonomous, independently operating controllers, so-called holons, operating in real time and physically connected to other controlled systems. The entire production process (e.g., a production line) is controlled by such small, distributed units, ideally without a central control unit. [21,22]. Each holon has only a certain part of the global information about the structure, capabilities, and goals of the production system. This is sufficient for independent functioning, and for effective cooperation through the exchange of messages [23].

Holons handle the necessary regulatory processes locally and independently, only in critical situations (e.g., when the production equipment is overloaded) do they inform other holons about the situation by sending messages. This initiates the activity of the holons, which leads to a change in the situation (to prevent partial failures, or, for example, to reconfigure the production system) [24].

The holonic approach is much more flexible than the classical approach in terms of the broader range of tools used for control, design, and management of manufacturing systems as well as entire enterprises [25]. They are robust, fault-tolerant, easily configurable, etc.

2. Materials and Methods

The main goal of the research was to design a data collection and processing system for a holonic system as part of an intelligent manufacturing system. The purpose was to design and experimentally verify a communication interface capable of collecting, evaluating data and communicating with other holons. These holons will represent information systems and process level control systems. The

algorithm for the design of data collection from a holonic production system on selected parts of the production holonic system is verified.

The design of the production data collection system will contain a comprehensive sequence of steps and a list of necessary support equipment, whether technical or software. The partial goals necessary to meet the main goal are:

- Identify differences in the method of collection and management of centralized and distributed control systems.
- Design of software and technical equipment of a holonic production system.
- Identification of key functional parts of the systems and design of requirements for information production systems.
- Purpose and role of production information systems in the holonic production system of an enterprise.
- Design of holons and their information protocols in intelligent distributed production systems and in decentralized control systems.
- Design of an algorithm for collecting and evaluating data from holons.
- Design of a unified communication protocol for a holonic production system.
- Design of technical and software equipment for solving the task.

The latest trend [26–29], which is still in the experimental stage, is the concept where a product, or rather a semi-finished product, can be a holon and is able to communicate with the production equipment about its requirements. It can tell it which operation it needs, whether it should be processed on the given equipment or not. The equipment can respond, for example, that it has no free capacity, and the semi-finished product must find another way.

The product and the production line can therefore communicate with each other. The technology that makes this possible is, for example, RFID (Radio Frequency Identification). The latest RFID tags allow not only reading, but also writing, and can thus communicate with the production equipment in both directions. The multi-agent approach used in production management can also be used for planning and scheduling. There is therefore a direct link between planning, scheduling and actual management, and these systems can communicate with each other without any problems.

For example, a system that controls production detects a malfunction or detects a new device and passes on information to the planning system that the production line capacity has changed. The planning system can then re-plan production accordingly. The whole thing forms a single multi-agent system. The fact that planning and scheduling can be directly linked to operational control is one of the greatest advantages of multi-agent systems [30].

3. Results

Based on a comprehensive analysis of the principle of operation of data collection systems and the definition of scientific problems in the subject area, an algorithm was designed (Figure 3). The algorithm represents a methodological sequence of steps that must be performed when designing the introduction of holonic control into an existing or new production system. Areas related to the design of a holon ensuring interaction between elements of the production system will be defined in subsequent subsections, which will clarify the requirements for technical and software equipment necessary to meet the main goal.

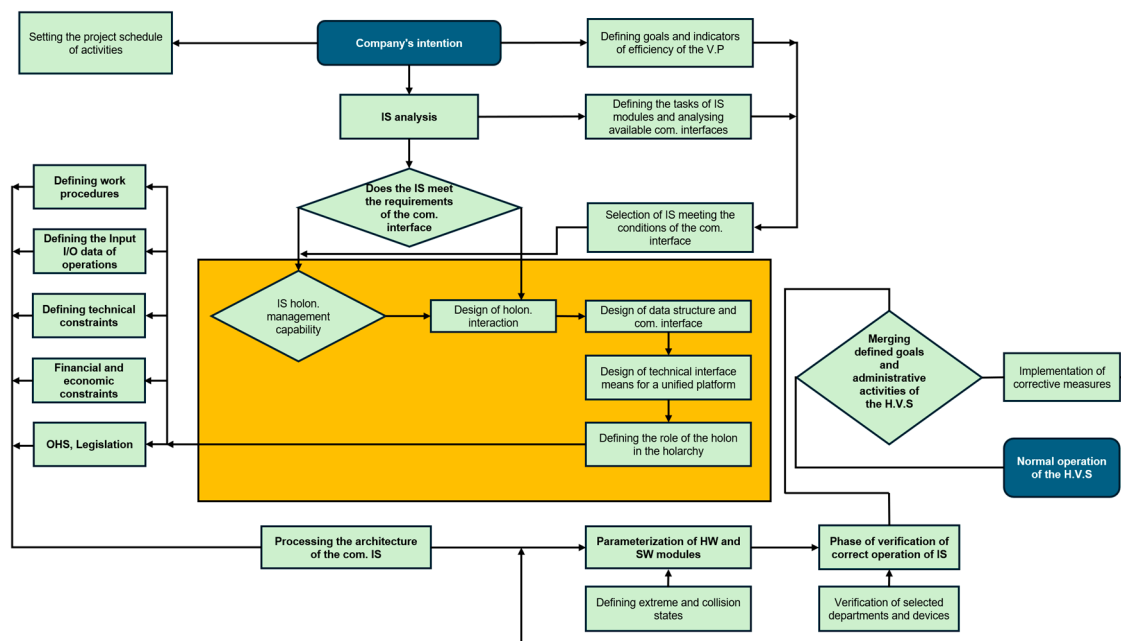


Figure 3. Algorithm for implementing holonic control in production.

Information systems for supporting efficient and flexible production have their defined position and tasks within the vertical integration of information systems. The systems necessary for the holonic production system and their software will be the subject of the chapter. In the chapter, we will identify and design the necessary modules and their functionality. This functionality forms the condition that is set for the fulfilment of information systems for the holonic production system.

The following subchapters present the operating principle and structure of enterprise information systems in the context of mutual integration possibilities with respect to the holonic management approach.

3.1. MES Systems

MES systems are primarily developed for operational planning and production management, and their purpose is to provide operational information for immediate management and optimization of production processes. MES systems help production managers to better use information for launching or optimizing the production plan. They are also suitable for deployment in production, where a functional ERP (Enterprise Resource Planning) enterprise system is already deployed to streamline the management and optimization of the company's production processes. It is therefore a directly integrated computer system that accumulates methods and tools necessary for improving and optimizing production.

Unlike classic information systems, they work with current data in real time, which allows them to flexibly respond to both non-standard conditions in production and immediate business requirements and to adapt the production process to be as efficient as possible. Regarding enterprise groups of systems, MES act simultaneously as a source and a recipient of information. Production information systems are an effective tool for monitoring, managing and evaluating the production process in all its complexity. They were developed to fill the communication gaps between the production planning system (MRP -Material requirements planning) and the MCS (Manufacturing Control Systems) used to run the equipment on the production platform.

MES functionality covers a wide range of features that a system can contain. These manufacturing information systems generally implement functional areas according to the requirements of the MESA association (Figure 4). Several functions may overlap when implementing the system in specific conditions, and conversely, some functions may not be included in the final version at all. The resulting solution always depends on the needs and requirements of the customer. The MESA association (The Manufacturing Execution Systems Association) therefore conducted a study on users and offered the following list of benefits of using a computer-controlled manufacturing information system:

- reduction of production cycle time, reduction of time required for implementation,
- reduction or elimination of input data processing time,
- reduction of work in process [31],
- reduction or elimination of administrative work,
- improvement of product quality,
- strengthening of the growth of operating technicians,
- improvement of process planning,
- improvement of customer service [32].

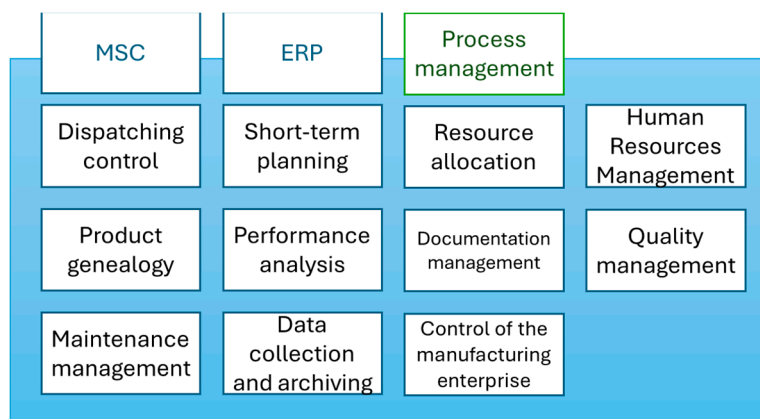


Figure 4. Functional model of MES systems according to MESA.

Visualization of technology statuses serves for an immediate overview of the operating status (Figure 5) of individual machines, production lines and equipment. It includes information on operating conditions in production, technological parameters affecting production quality, data related to products (machine cycles, numbers of pieces), records of downtimes or equipment shutdowns during machine setup. This allows the creation of short-term (e.g. daily) production schedules considering the sequences of production operations and their distribution between individual production equipment.

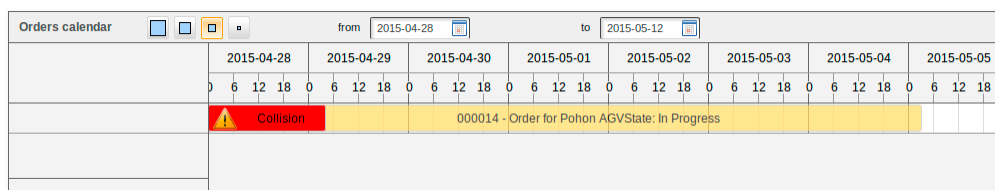


Figure 5. Example of a parallel overlapping operation in the Qcadoo MES environment.

Resource allocation ensures that all necessary production resources (machines, tools, labour, materials, energy, etc.) are available for the start of production (in the correct time-place-quality configuration).

Allocation of production units according to assigned work orders and schedules, coordination of production between lines, ensuring the necessary number of raw materials and energy, and monitoring the status of the production cycle.

Documentation management is a program module for managing all records and documents (schemes, production procedures, schedules, change protocols, procedures, work orders, etc.), information on the progress and results of production, comparison of the assignment with reality, and sending instructions to the equipment operator and at the same time providing procedures for control systems. Log export options in the MES Qcadoo environment are shown in Figure 6.

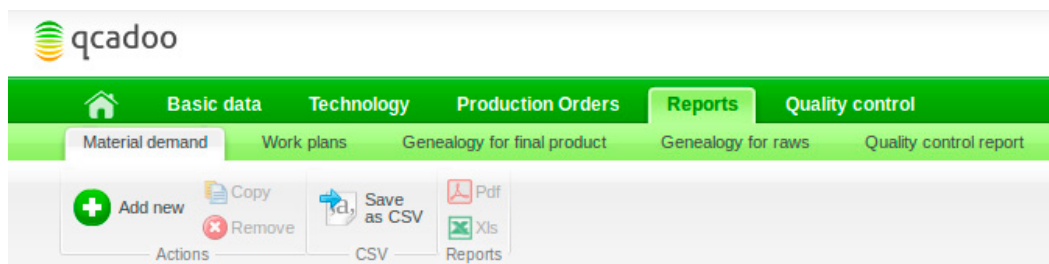


Figure 6. Log export options in the MES Qcadoo environment.

Tracking each product, batch or series throughout the entire production cycle and preserving the actual conditions under which they were produced (records of individual production steps, materials used, procedures, the course of key technological variables, etc.). Performance analysis monitors and calculates key production indicators, compares the results currently achieved in production with their short-term history and predicts estimates of economic outputs. This functionality can also be processed by an ERP system that has advanced performance analysis. Human resources management keeps records and informs about personnel qualifications (education, certificates, special knowledge and skills). It tracks indirect activities in the preparation of materials, machines and tools as a basis for calculating costs based on activity (e.g. how much time someone spends solving a downtime, changing a machine).

The functional area of maintenance planning and management monitors and manages activities carried out with the aim of maintaining production assets in such a technical condition as to prevent unplanned production interruptions. It provides periodic preventive maintenance schedules and allows you to manage maintenance according to the actual condition of the equipment. Control of the production process is provided by operator functions. This functionality can only be provided in the case of connecting the technology control system with a specific module that allows sending commands that can be processed by the control system. Quality management ensures real-time analysis of data captured from the production equipment to monitor the quality of the manufactured product and identify unwanted deviations in a timely manner. It uses SPC / SQC methods of continuous statistical search for differences between the required "ideal" and actual process parameters and the search for the causes of these differences. Quality management (Figure 7) also often includes off-line analyses of the information system.

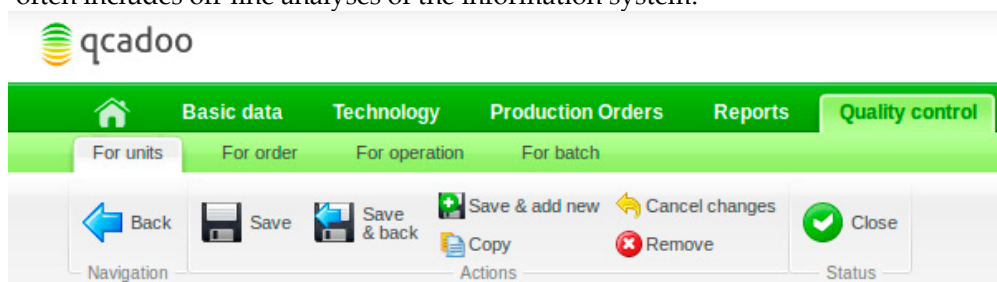


Figure 7. Quality control module in MES Qcadoo.

Data collection and archiving is the basic building block of every MES system. It ensures continuous collection of production data in real time, their long-term archiving and availability for further processing. An integral part is data protection against loss and misuse. MES thus ensures the implementation of individual functions, for example:

- operational planning and optimization of production series,
- process control and electronic production recording,
- technological data collection and archiving,
- data analysis, balancing and creating protocols,
- production monitoring and production operation history,
- tracking of materials in production, operational inventory.

There is a constant exchange of information between the MES level and the higher and lower levels during system operation. The problem with the exchange of information between system levels is caused by:

- different levels of abstraction, work procedures (Figure 8), documents, control signals,
- different processing periods (MES systems operate in real time and the period is in milliseconds),
- different data structures (different types of documents, design drawings and other records in databases),
- different levels of accuracy (measured quantities, control signals, forecasts, plans, estimates),
- different processing approaches (managers, production workers, machines and equipment).

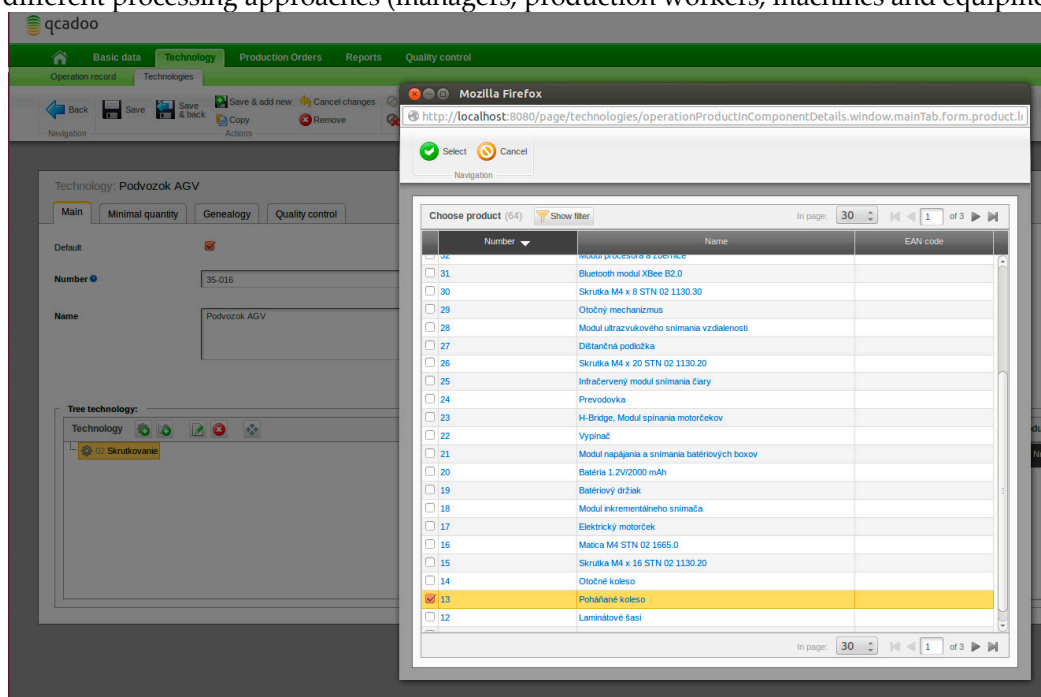


Figure 8. Workflow management in MES QCADoo.

The MES system modules according to the MESA association can be divided into main (gray) and support functions (orange), as shown in Figure 9.

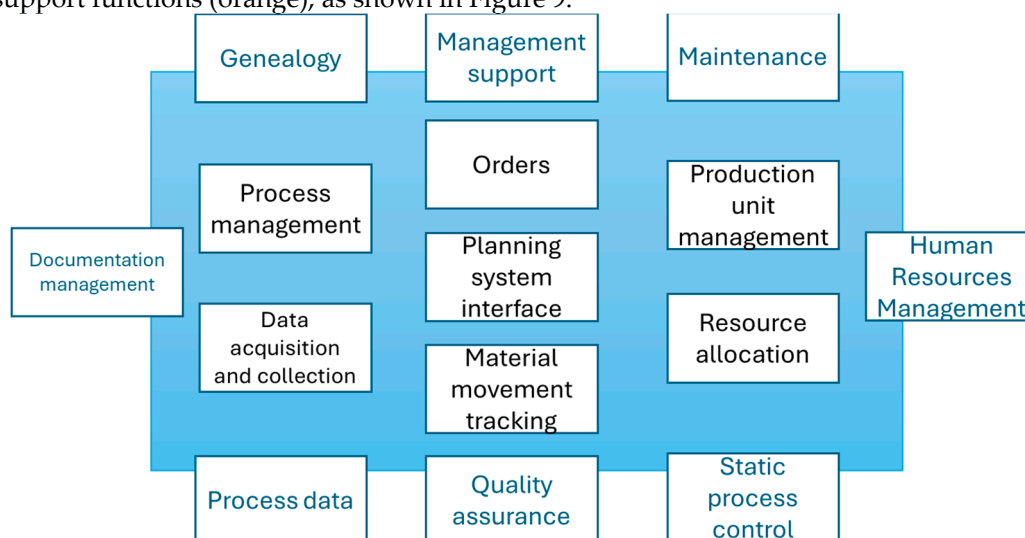


Figure 9. Division of MES modules into main and support functions.

3.2. Design of Interaction Holon

The proposed holon architecture respects the FIPA CNIP protocol. To implement the holon, it is necessary to choose the appropriate software and hardware. There are proposed the basic algorithm

of the holon, which can work in combination with the input-output circuits of the system to ensure communication via the industrial bus of the control system, or data transfer from the sensor (Figure 10). The proposal is focused on universality. The holon can also work at the level of the operating system of the controlled device (Figure 11). This means that the software - the control program and the communication interface are identical. The holon can be implemented at the level of:

- Sensor,
- PLC,
- Information system.

The following subsection will deal with the design of means that meet the condition of compatibility with the identified levels.

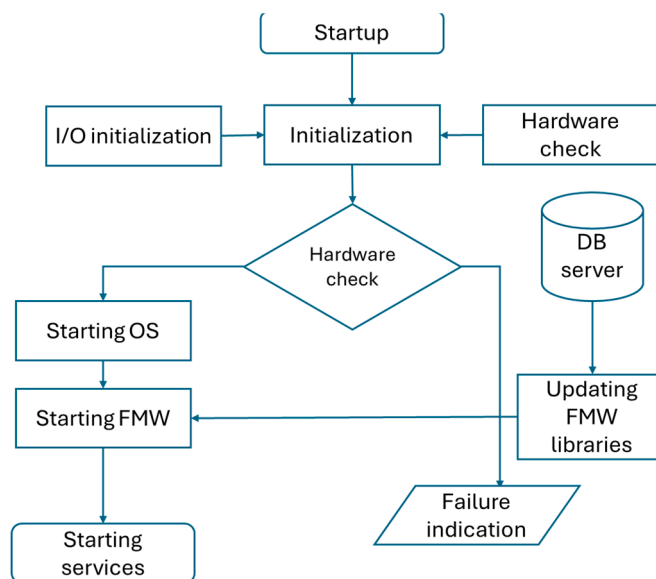


Figure 10. The principle of operation of a separate interaction holon.

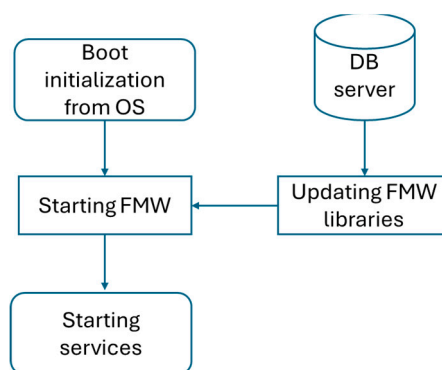


Figure 11. The principle of operation of the interaction holon in the software version.

3.3. Design of Technical Interface Means for a Unified Platform

When designing the software, there is emphasized meeting the requirements of holon compatibility with the environment in which the holon will be integrated. The role of the holon software is also the ability to work at the level of the device's operating system, which will expand its control properties with elements of the holonic principle. When selecting, we also focused on the case when there is a need to create a separate physical holon that can communicate, control and process signals from sensors. The software must therefore meet the conditions for the universality of the operating system platform used. Another condition is to implement a quick change of the control program, use program modules implemented in different programming languages and be able to integrate them into a functional unit.

To ensure the cooperation of holons, the correct choice of the architecture of mutual communication is essential. The most common way of implementing software communication identified during this research appeared in the literature and Internet resources SOA architecture, which is also often implemented. The abbreviation SOA comes from the English phrase Service - Oriented Architecture and in recent years has become a synonym for a new concept of connecting business needs with the possibilities of current information technologies in the enterprise. It represents a concept whose creation was required by the rapid development of IT and information systems in the 1990s, as well as the solution of related problems in the transfer and processing of information in heterogeneous systems. The SOA concept (Figure 12) provides a guide to solving integration problems either within a single enterprise or between several enterprises. The core is the use of services in processing requests for information exchange between heterogeneous systems. At the same time, the concept of service in IT terminology can be understood in the same way as this concept is understood in everyday life (order, etc.). In a narrower sense, a service should be understood as a suitable application (component) that meets the following requirements in particular:

- Clear localizability in a certain environment (e.g. on the Internet or in a certain organizational unit) [33].
- Support for globally accepted standards (XML, SOAP, JMS, JCA, etc.), which ensures the usability of the service without the need to develop and use proprietary communication protocols.
- Autonomy – no additional software needs to be installed to use the service.
- Self-description – information is also available at the location where the service is localized about what functionality the service provides and under what conditions this functionality can be used.

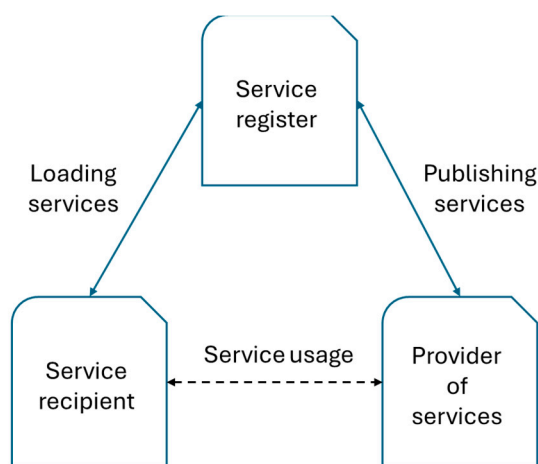


Figure 12. Conception SOA.

In SOA implementations, there are cases where meeting communication requirements is unnecessarily complicated and time-consuming – for example, obtaining additional status information and information about additional resources. We therefore propose using REST architecture as a replacement for SOA architecture for holons whose communication layer is formed by separate holons, such as in the form of a PC.

REST (Representational State Transfer) is a way to easily create, read, edit, or delete information from a server - holon using simple HTTP calls. REST represents an interface architecture designed for distributed environments. REST was designed and described in 2000 by Roy Fielding (co-author of the HTTP protocol) as part of his dissertation *Architectural Styles and the Design of Network - based Software Architectures*. In the context of the work, the most interesting chapter 5 is in which Fielding derives the principles of REST based on known approaches to architecture. The REST interface can be used for uniform and simple access to resources. The resource can be data, as well as

application states (if they can be described by specific data). REST is therefore, unlike the better-known XML - RPC or SOAP, data-oriented, not procedural.

The biggest difference between SOAP and REST is therefore mainly in how you communicate with the server - what standards and how they use them. On the contrary, the method that will serve this request on the server is written in a different programming language and can be very similar in both cases. A REST framework is needed that will call the correct method based on the URL. An interesting feature of REST is that in the response that comes from the server, references to other data can be written - represented by URLs. In this way, further data can be obtain based on the first response. This is undoubtedly a feature that is difficult to achieve in SOAP services, and it is also a step back to the classic web, where hyperlinking is a common thing.

Basic principles of REST communication architecture - the state of the application and its behaviour is expressed by the so-called RESOURCE (key resource), each resource - holon must have a unique identifier (URL). HATEOAS (hyper-medial as the Engine of Application State) - represents the state of the application and is specified by the URL. Other possible states can be obtained using links that the client receives in the response from the server. A unified approach is defined for obtaining and manipulating resources in the form of four CRUD operations (Create, Read, Update, Delete). Resources can be represented (XML, HTML, JSON, SVG, PDF), the client does not work directly with the resource, but with its representation [34].

3.3.1. Communication protocol

- Client/Server - used to define responsibility.
- Stateless - each request must contain all the information necessary to execute it.
- CACHE - each request can be explicitly marked as cacheable or non-cacheable, which allows transparently increasing performance by adding a cache between the client and the server. Code-On-Demand - client functionality can be extended by code sent by the server (for example, JavaScript).
- Layering - allows the stacking of layers providing services to increase variability (cache, transformation, load balancing, etc.) [35].

There are of course other approaches to solving distributed architectures such as RPC (Remote Procedure Call). In general, the difference can be described between REST and RPC is on two levels:

- semantics of operations and what is distributed,
- semantics of operations in REST is final and consists only of CRUD (create, read, update, delete) on a given resource - holon.

In contrast, the semantic structure of RPC corresponds to the methods that are called. In REST, the state (data represented by a resource) is distributed, as opposed to the message that is distributed in RPC.

REST architecture represents services that are about a smaller number of standards and their more efficient use. The basic standards are HTTP, URI and XML (or JSON, or XHTML, etc.). Knowledge of the above is a basic prerequisite and a necessary condition for using and communicating with REST services. The basic idea is that the URI defines the data you want to work with and the HTTP operation, i.e. what we want to do with the data. So far, we have encountered two operations GET and POST, the HTTP standard has several of them. Let's mention: PUT, DELETE, HEAD and OPTIONS. The set of these operations allows us to design a holonic structure for managing other holons. The GET and POST operations have gained their popularity mainly because they are still used on the standard web (GET to get data from a page and POST to send data from a form). When designing the concept, the Python language was chosen because of its multi-platform nature.

3.3.2. Python Programming Language Design

Python is a multi-paradigm language similar to Perl. Python supports object-oriented, structured, and functional programming. It is a dynamically typed language, supporting many high-

level data structures. Although Python is often referred to as a "scripting language", it is used to develop many large software projects such as the Zope application server and the Mnet and Bit-Torrent file-sharing systems. It is also widely used by Google. Python proponents prefer to call it a high-level dynamic programming language, because the term "scripting language" is associated with languages that are used only for simple shell scripts, or with languages like JavaScript: simpler and for most purposes less capable than "real" programming languages like Python [36].

Another important feature of Python is that it is easy to extend. New built-in modules can be easily written in C or C++. Python can also be used as an extension language for existing modules and applications that need a programmable interface. Supported operating systems:

- Linux,
- BSD,
- Mac OS X,
- Windows.

The support of the operating systems was key in choosing the Python language. Industrial computers support the Windows operating system in most cases. On the other hand, information systems are largely run on servers built on the Linux operating system for reasons of stability and security.

Program module ensuring interaction

To meet the requirement of holon communication on a unified data layer, there is chosen an extension for the Python language. This extension ensures communication of the holon control program with the control programs of other holons in the holarchy through the REST architecture. The REST architecture can be implemented by its own extension or by using existing ones. From the perspective of the sophistication of the existing ones, the development of its own architecture is inefficient. The source code is accessible and therefore possible extensions or modifications are feasible [37].

Bottle is a fast, simple and lightweight WSGI micro web-framework for Python. It is distributed as a module in a single file and does not use any other than standard Python libraries [38].

Key features that were considered in the selection:

- Routing: Requests for prompt execution of a function through mapping with support for clean and dynamic URL addresses.
- Templates: Fast and built-in core templating system with support for other Python extensions such as: mako, jinja2 and Gopard.
- Transactions: Convenient access to choosing the form of input or output data, adding files, cookies, headers and other metadata for HTTP purposes.
- Server: Built-in HTTP server with support for modules: fapws3, Bjoern, Gae, CherryPy, or any other compatible with the ability to work with WSGI HTTP server. Example of Python syntax and use of the Bottle module when loading data from the information system database.

The proposed architecture can also be used for devices designed for the Internet of Things.

3.3.3. Design of the Technical Equipment of the Interaction Holon

When choosing the technical equipment and implementing the interaction holon, it is important to pay attention to the support of the software that there is defined in the previous chapter. When the interaction holon cooperates in the case of connection with industrial automation control systems, it is necessary to consider the number of types of communication interfaces and its durability in the industry.

During experimental verification, there is worked with a credit card-sized computer called Raspberry PI. Specifications of the proposed experimental holon interaction:

- 900MHz quad-core ARM Cortex-A7 CPU,
- 1GB RAM,
- 4 USB ports,
- 40 GPIO pins,

- Full HDMI port,
- Ethernet port,
- Combined 3.5mm audio jack and composite video,
- Camera Connection Interface (CSI),
- Display Interface (DSI),
- Micro SD card slot,
- 3D VideoCore IV graphics core.

Currently, other models from various manufacturers based on the same architecture that allows for the implementation of holonic control and data collection are appearing on the market. There are several modules for the devices, such as communication extensions, input/output modules. In the experiment, there is used the standard hardware configuration of the Raspberry PI model 2 computer.

Software architecture design (Figure 13) of the proposed solution using a computer from the point of view of software. The holon operating system is based on the Linux kernel, which can work with input-output ports (GPIO) that can read a digital or analogue signal.

There is an I2C communication line to which it is possible to connect peripherals that expand the original hardware configuration capabilities of the device. The operating system runs the WSGI service, which is part of the software that was specified and provides REST API communication with other holons in the network.

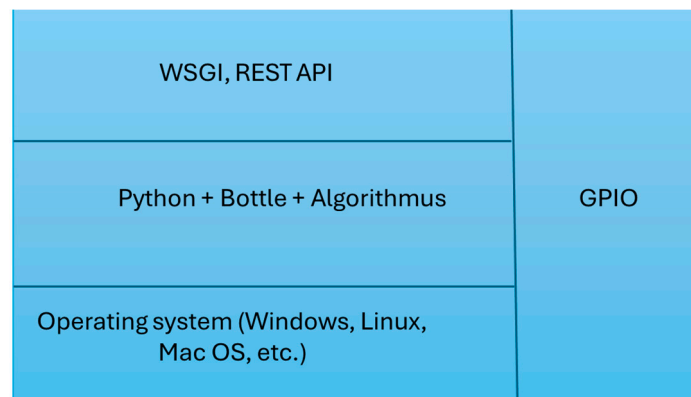


Figure 13. Internal software architecture of the interaction holon.

Defining the roles of a holon in a holarchy - mutual interaction is a fundamental property of holonic control. Holons exchange information with each other. Several protocols have been introduced in this area, covering querying, voting, negotiation and auction execution, and have also been standardized by the FIPA (Foundation for Intelligent Physical Agents) organization. This organization was founded in 1996 with the aim of introducing a whole set of standards that define agent systems and their communication. The established standards include:

- FIPA Agent Communication Language (ACL), which defines the language in which agents communicate. The structure of the given message and its ontology are defined [39].
- FIPA Contract Net Interaction Protocol (CNIP), which is based on CNP and is one of the most widely used auction protocols. [40].

CNP was designed in 1980. It is a high-level communication protocol for distributed control and cooperation in task execution. Any holon can initiate a negotiation process and contact other holons requesting that they provide a given operation (or service). This makes it a “manager” or “moderator” that can contact other holons and mediate mutual communication. It can contact:

- All holons (broadcast).
- Selected holons (multicast) [41].
- A unique holon (unicast) [42].

FIPA CNIP is based on small modifications of CNP, namely the addition of rejection and confirmation communication messages (messages requiring the execution of actions - acts). Holons that accept this request are together able to generate n responses. Each initiated conversation (part of communication with a certain goal) is marked with a conversation-id parameter, which is global, non-zero, and assigned to the initiator. Holons that are part of this communication are forced to mark the ACL message with this identifier. This allows the holon to distinguish individual conversations and reflect on them in a state of inactivity.

Defining communication via ACL messages - at any point in the conversation, the recipient of an ACL message (Figure 14) can inform the sender that he did not understand the content of the message, namely with a not-understood message [43]. This message can terminate the entire IP, which implies that all commitments made during this conversation are invalid. In the case where the IP concerns multiple participants, each response must be evaluated separately and some of these messages may also be not-understood. For this reason, it may not be appropriate to terminate the entire IP, since participants can continue communicating with their subprotocols [44].

Canceling mutual holon communication - at any step, the initiator can cancel the IP, namely Fipa-cancel-Meta-Protocol (Figure 15), which contains the cancel message, and whose conversation-id is the same as the conversation that the initiator wants to cancel. The participant informs the initiator whether the cancellation was successful, namely with the information - done, or that the IP termination failed, namely with the message failure.

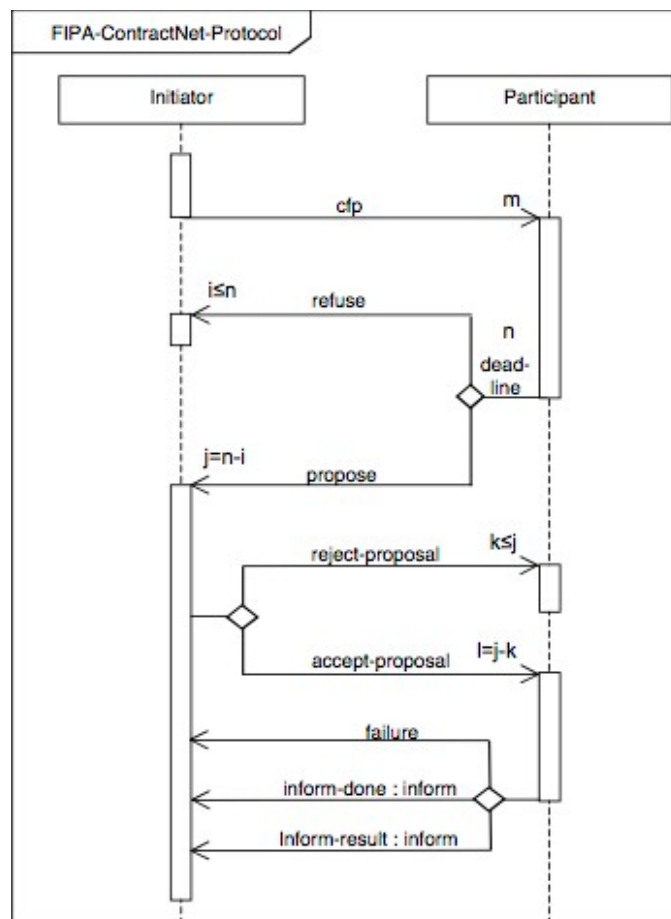


Figure 14. Defining the structure of the ACL message.

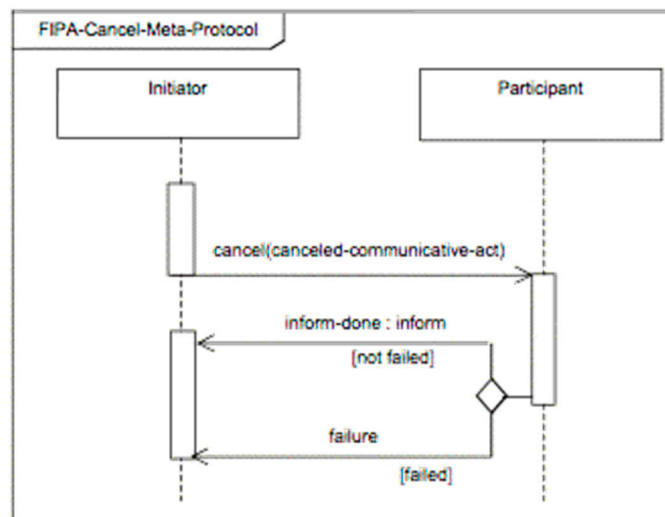


Figure 15. FIPA Cancel Meta-Protocol.

Experimental verification of the proposed methodology was carried out in the conditions of the ZIMS workplace. The subject of the experiment was the implementation of information software within the common data network, which met the conditions set out in the algorithm. In another experiment, the functionality of the proposed interaction holon was verified.

4. Conclusions

The essence of the ZIMS workplace (Figure 16) is research, development and experimentation in the field of intelligent manufacturing systems. The effort to connect the design and implementation parts of intelligent manufacturing systems. By applying PLM (Product Lifecycle Management) systems, products are being designed from the very beginning of their design, through the technologies used for production, designing production and logistics processes to the virtual commissioning and the actual implementation of automated equipment such as industrial robotics, industrial automation and logistics AGV systems. The ZIMS laboratory is mainly focused on long-term research and not on direct commercial projects.

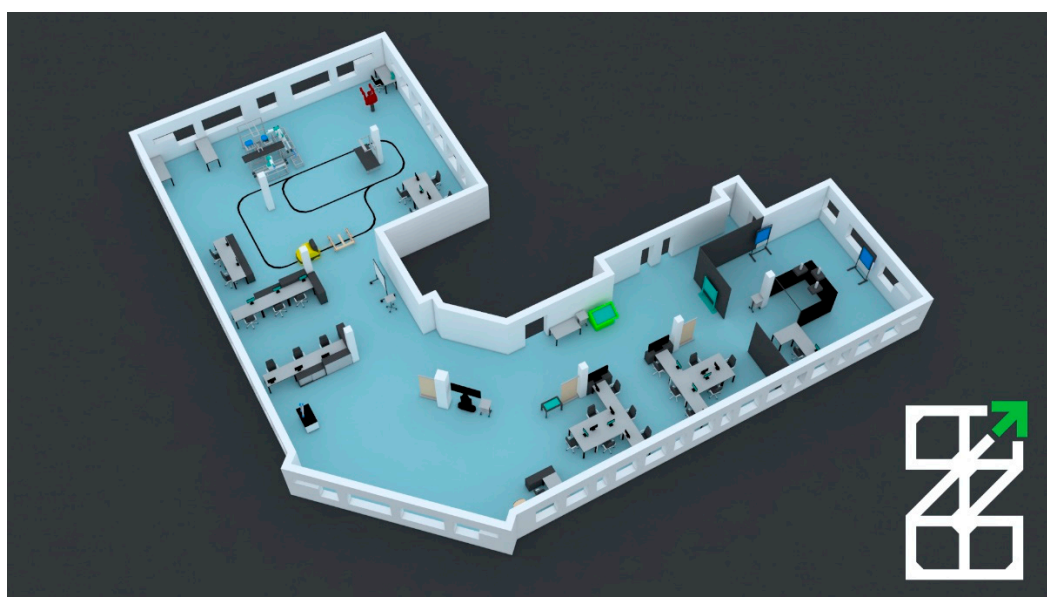


Figure 16. ZIMS Laboratory.

All systems communicate via the Ethernet network, and their functionality is available via the Internet interface. The network architecture is designed so that it is possible to access the ERP system

from the Internet. This decision resulted from the real need to enable access to selected data outside the private network of the company. For example, recording business opportunities of external employees of the company. SCADA/HMI systems are available within the internal network. and MES. The architecture of the resulting production system, the operation and purpose of the systems is shown in Figure 17. The resulting architecture integrates the process level integrated by the proposed unified platform with the information systems of the holonic production system.

Specifically, these were the following systems:

- ERP System OPEN ERP7 (Figure 18),
- MES Qcadoo (Figure 19),
- SCADA/HMI ControlWeb.

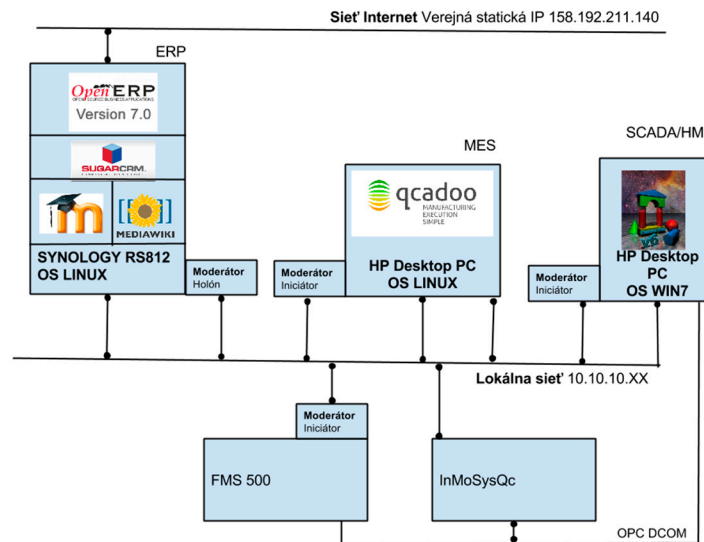


Figure 17. Architecture of experimental verification of information systems.

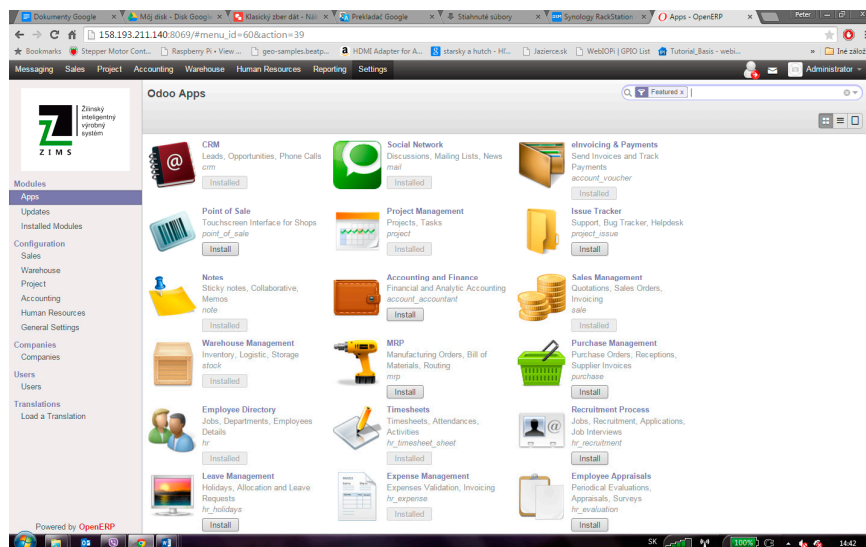


Figure 18. Overview of installed modules in the OPEN ERP 7 system.

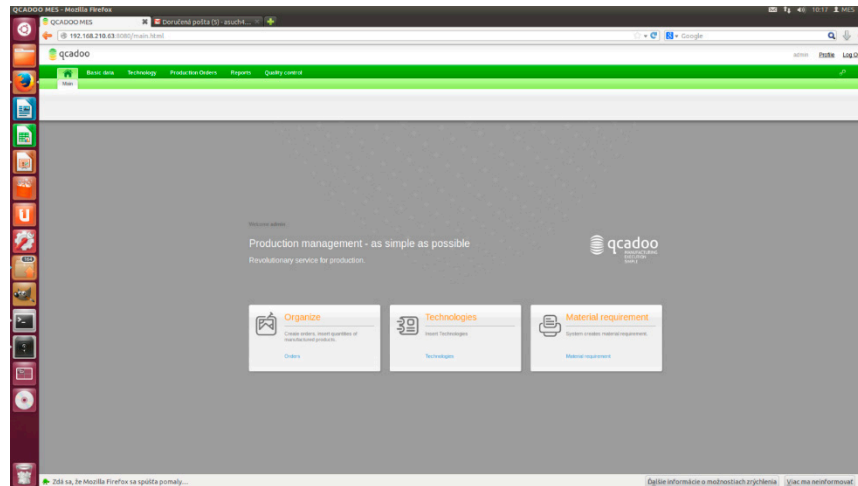


Figure 19. Qcadoo MES system home screen.

The information systems described at the beginning of the chapter are installed on separate devices. In the case of the MES Qcadoo system, it is a PC equipped with the Ubuntu operating system, the kernel of which is Linux. The ERP system is installed on a Synology server, which also uses an operating system, the kernel of which is Linux. Linux-based operating systems have Python language support in the basic installation. The SCADA/HMI system, which is installed on a PC with the Windows operating system, does not contain Python language support in the basic installation. However, this support can be installed additionally. It is thus possible to apply the interaction holon in the software version to information systems with Python language support without the need for additional technical equipment. To verify the operation of the interaction holon, the task of which is the ability to interact with another holon in the network. Example in the figure In the Figure 20 is shown the response to a request that is routed from holon 1 to holon 2, where the response is static text. In a similar way, the code is supplemented with a database connection, or work with the GPIO port. These records can be deactivated, but during the experimental phase they provided me with valuable information about the status of the holon web server. In the Figure 21 is shown a part of the code that illustrates the PYTHON language environment and the syntax of the source code and the return function of the presented function.

```

*Python Shell*
File Edit Shell Debug Options Windows Help
Python 2.7.3 (default, Mar 18 2014, 05:13:23)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Bottle server starting up (using WSGIRefServer())...
Listening on http://10.10.10.127:8080/
Hit Ctrl-C to quit.

10.10.10.127 - - [18/May/2015 06:08:14] "GET / HTTP/1.1" 404 723
10.10.10.127 - - [18/May/2015 06:08:29] "GET /hello HTTP/1.1" 200 13
10.10.10.127 - - [18/May/2015 06:10:13] "GET /hello HTTP/1.1" 200 13
10.10.10.127 - - [18/May/2015 06:31:12] "GET / HTTP/1.1" 404 723
10.10.10.127 - - [18/May/2015 06:31:25] "GET /hello HTTP/1.1" 200 13

Ln: 6 Col: 0

10.10.10.127
http://10.10.10.127:8080/hello

Im RIGHT side
pi@raspberrypi: ~
File Edit Tabs Help
t) in auto mode
update-alternatives: using /usr/bin/identify.im6 to provide /usr/bin/i
identify) in auto mode
update-alternatives: using /usr/bin/stream.im6 to provide /usr/bin/st
m) in auto mode

```

Figure 20. List of holon request records from the network.

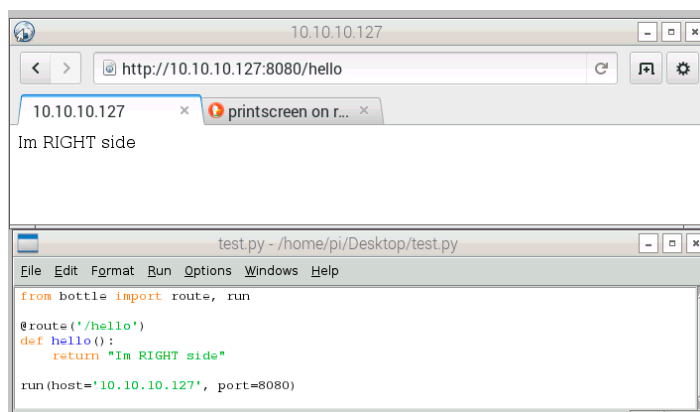


Figure 21. Entry of a function using the REST API.

Using the software and technical means described in this paper, it is possible to achieve effective collection and management of a holonic production system. The proposed methodology has been verified and its functionality confirmed. However, the activities listed in the methodology algorithm outside the marked area of future research are equally important and time-consuming. It is their correct analysis and specification that determines the final length of the integration of holonic control into the production system. The proposed and verified methodology can ensure the sustainability of the solution in terms of further support for communication interfaces, which are currently a trend. However, the question remains of defining responsibility and guarantees for possible system failures in the event of a malfunction in the case of a holon interaction failure, whose creator may not be the manufacturer of the production equipment. However, these and other questions are more of a legislative nature.

Technically, it is possible to modify the proposed system or implement it in different programming languages or computing devices. The condition is to maintain the structure of requests and responses in REST communications. The HTTP protocol thus enables easier interconnection of systems in terms of complexity and security of communication, which companies require to the maximum extent possible. The proposed communication eliminates the isolation of systems in terms of the use of separate databases, but rather a common database, thereby eliminating the duplication of production data, which will lead to increased efficiency in data collection.

Today's automatic control systems are generally centralized and strictly hierarchical. This is how entire large-scale industrial networks are built today, connected to programmable controllers that also operate in a centralized mode. Unlike centralized systems, multi-agent and holonic systems represent a dynamic, easily expandable alternative. The system can respond very effectively to changes caused by the arrival of a priority order or the failure of a production unit, with its response proportional to the severity of the cause. One specific agent will attempt to solve the problem, and if it fails, it will request the cooperation of surrounding agents. The structure of production lines and the production process are not fixed in the control system structure but are created dynamically when a new order is created. They are automatically adjusted with each change. Since the control and planning process is distributed across a larger number of computing units, the risk of instability caused by the failure of a single agent is minimized. Many applications require highly distributed control.

These include applications in the chemical industry or in the distribution of electricity, gas, or water, where it is necessary to have autonomous units that perform many interventions in the controlled technology independently, without communication with the centre. In flexible production sections, it is sometimes necessary to replace, add, or remove certain equipment during operation, not only due to failure or maintenance, but also when changing the production plan. It is important to find a new production path as quickly as possible for each such change. Holonic and multi-agent systems are a suitable solution for all these purposes.

An example is a conveyor system where one conveyor malfunctions. In a classic setup, it is necessary to stop production, reprogram the line, and restart it. With agent control, it works differently: as soon as the device detects a malfunction, it notifies all other devices that are interested. They immediately begin to discuss a replacement solution and send the products via a different route. When the conveyor is repaired, it sends a message that it is ready for operation and agrees with the

surrounding devices what it can do for them. Nothing must be stopped or reprogrammed; everything works completely automatically. Holonic and multi-agent systems are therefore much more flexible and robust than traditional centralized control, allowing the configuration of the production system and production plans to be changed automatically.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used “Conceptualization, B.M. and V.B.; methodology, B.M.; software, A.K.; validation, V.B.; formal analysis, V.B.; investigation, V.B.; resources, B.M.; data curation, A.K.; writing—original draft preparation, V.B.; writing—review and editing, B.M.; visualization, A.K.; supervision, V.B.; project administration, V.B.; funding acquisition, V.B.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Funding: This work was supported by the APVV 19-0305, KEGA 011ŽU-4/2025 and VEGA 1/0633/24.

Conflicts of Interest: “The authors declare no conflicts of interest.”

References

1. Bubeníková, E. and Bubeník, P. (2017). Internet of things (IoT). *Technológ*, Roč. 9, č. 1 (2017), s. 131-134, ISSN 1337-8996
2. P. Peniak - E. Bubeníková. (2019). Validation of IoT secure communication gateway for constrained devices, International Conference on Applied Electronics (AE), Pilsen, Czech Republic, 2019, pp. 1-5, doi: 10.23919/AE.2019.8866990.
3. Campbell, J. D., Reyes-Picknell, J. V., & Kim, H. S. (2015). *Uptime: Strategies for excellence in maintenance management*. CRC Press.
4. Duffuaa, S. O., & El-Ga'aly, A. (2015). Impact of inspection errors on the formulation of a multi-objective optimization process targeting model under inspection sampling plan. *Computers & Industrial Engineering*, 80, 254-260.
5. Bubeník, P. – Horák, F. (2014). Knowledge-based systems to support production planning. *Tehnički vjesnik*, ISSN 1330-3651, ISSN 1848-6339.
6. Rakyta, M. – Binasova, V. (2016). *Totálne produktívna údržba – TPM*, Žilina: EDIS, ISBN 978-80-554-1210-8.
7. Preez, J. (2011). Thesis presented in partial fulfillment of the requirements for the degree of Master of Science in the faculty of industrial engineering at Stellenbosch university. Stellenbosch University, South Africa, 123s.
8. Supadulchai, P. (2008). Reasoning-based capability configuration management in adaptable service systems. NTNU, Nórsko, Trondheim, 2008, 238 s.
9. Vyatkin, V. (2011). *Function blocks for embedded and distributed control systems design*. ISA, New Zealand, 2011, 260 s., (ISBN 978-1-936007-93-6)
10. Peniak, P. Rástočný, K.; Kanáliková, A.; Bubeníková, E. (2022). Simulation of Virtual Redundant Sensor Models for Safety-Related Applications. *Sensors* 2022, 22, 778. <https://doi.org/10.3390/s22030778>. (IoT)
11. Peniak, P. - Bubenikova, E. - Kanalikova, A. (2021). Extended gateway model for OPC UA/IoT device integration, In: 2021 IEEE 19th World Symposium on Applied Machine Intelligence And Informatics (SAMI 2021), Page155-159, DOI10.1109/SAMI50585.2021.9378643
12. Durica, L., Micieta, B., Bubenik, P., & Binasova, V. (2015). Manufacturing Multi-Agent System With Bio-Inspired Techniques: Codesa-Prime. *MM science journal*.
13. Micieta, B., et al. The approaches of advanced industrial engineering in next generation manufacturing systems. *Communications-Scientific Letters of the University of Zilina*, 2014, 16.3A: 101-105.
14. Micieta, B., et al. Modular Intelligent Control System in the Pre-assembly Stage. *Electronics*, 2024, 13.9: 1609.
15. Micieta, B., et al. Interfacing the Control Systems of Enterprise-Level Process Equipment with a Robot Operating System. *Electronics*, 2023, 12.18: 3871.

16. Micieta, B., et al. Product segmentation and sustainability in customized assembly with respect to the basic elements of industry 4.0. *Sustainability*, 2019, 11.21: 6057.
17. Micieta, B., et al. Indication, Solution, Prevention: a Holistic Approach to Financial, Industrial Engineering, and Business Problem Analysis. *Scientific Papers of Silesian University of Technology. Organization & Management/Zeszyty Naukowe Politechniki Slaskiej. Seria Organizacji i Zarzadzanie*, 2023, 188.
18. Dulina, L., et al. Improving Material Flows in an Industrial Enterprise: A Comprehensive Case Study Analysis. *Machines*, 2024, 12.5: 308.
19. Dulina, L., Krajcovic, M., Binasova, V., Zuzik, J., & Chladekova, D. 2024. Enhancing transportation through route strategy proposal using localization measurement technology: a case study. *MM Science Journal*.
20. Hsieh, F.-S. Dynamic composition of holonic processes to satisfy timing constraints with minimal costs. *Eng.Appl. Artif. Intell.* 2009, 22, 1117–1126
21. Kuric, I., et al. Analysis of the possibilities of tire-defect inspection based on unsupervised learning and deep learning. *Sensors*, 2021, 21.21: 7073.
22. Kračovič, M., et al. Utilization of Immersive Virtual Reality as an Interactive Method of Assignment Presentation. *Electronics*, 2024, 13.8: 1430.
23. Kuric, I., et al. Analysis of diagnostic methods and energy of production systems drives. *Processes*, 2021, 9.5: 843.
24. Hlavenka, B.: *Projektování výrobních systémů*. Akademické nakladatelství CERM, Brno 2005
25. Allen, J.: *Inovačné podnikanie*. Vydavateľstvo Pitman Publishing, London 2005. ISBN 80-85323-70-2.
26. Kletti, J. 2007. *Manufacturing Execution Systems - MES*. Mosbach : Springer, 2007. s. 272. ISBN 978-3-540-49743-1.
27. Meyer, H. - Fuchs, F. - Thiel, K. 2009. *Manufacturing Execution Systems (MES): Optimal Design, Planning, and Deployment*. Gainesville : McGraw-Hill Professional, 2009. s. 274. ISBN: 978-0-07-162602-6.
28. Mařík, V., Štěpánková, O., Lažanský, J., a kolektiv, *Umělá inteligence (4)*, Praha, 2003
29. Dvořák, J., *Expertní systémy*, VUT v Brně, Fakulta strojího inženýrství, 2004
30. Kvasnička, V., Pospíchal, J., Tiňo, P., *Evolučné algoritmy*, STU Bratislava, 2000.
31. M. Hadi Valipour, Bavar AmirZafari, Kh. Niki Mki, Negin Daneshpour, A Brief Survey of Software Architecture Concepts and Service Oriented Architecture, in *Proceedings of 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT'09*, pp 34-38, Aug 2009, China.
32. Balogh, R., Bélai, I., Dorner, J., Drahoš, P.: *Priemyselné komunikácie*. Vydavateľstvo STU, Bratislava, 2001. ISBN 80-227-1600-6.
33. Homola, J. 2013. Jaké jsou možnosti pro publikování 3D CAD modelů na webu?. [online]. 17. 5. 2013 [cit. 2013-12-15]. Dostupné na internete:<http://www.caxmix.cz/2013/05/17/jake-jsou-moznosti-pro-publikovani-3d-cad-modelu-na-webu/>
34. Harďoň D. 2012. *Dáta cez internet zadarmo: Súboj cloudových služieb*. [online]. 02.03.2012 [cit. 2014-02-14]. Dostupné na internete:<http://www.zive.sk/data-cez-internet-zadarmo-suboj-cloudovych-sluzieb/sc-3-a-299549/default.aspx>
35. Pinkham. A. *Advanced 3D HMI/SCADA Visualization 2012*[online].2012 [cit.2014-03-21] Dostupné na internete: <http://partners.iconics.com/%2FTechSupport%2FGENESIS64%2FGenesis64%2FWhitepapers%2FAdvanced-3D-Visualization-for-Manufacturing-and-Fa.aspx>
36. Likindoy. *Professional Free Open Source GPL SCADA System* [online].[cit. 2014-03-25] Dostupné na internete: < http://www.likindoy.org/en/About-Likindoy/what_is_it>
37. Openscada. *Documentation*. on-line text. [online].[cit. 2014-03-28]Dostupné na internete:<<http://openscada.org/documentation/>>
38. Gyurák Babel'ová, Z., Stareček, A., & Vraňaková, N. (2025). Selected Attributes of Human Resources Diversity Predicting Locus of Control from a Management Perspective. *Administrative Sciences*, 15(9), 333.
39. Gyurák Babel'ová, Z., Vraňaková, N., & Stareček, A. (2022). Moderating Effect of Industry 4.0 on the Performance of Enterprises in the Constrains Related to COVID-19 in the Perception of Employees in Slovakia. *Administrative Sciences*, 12(4), 183.
40. Bohušik, M., Stenclák, V., Císar, M., Bulej, V., Kuric, I., Dodok, T., & Bencel, A. (2023). Mechatronic Device Control by Artificial Intelligence. *Sensors*, 23(13), 5872.

41. Bubeník, P., Rakyta, M., Buzalka, M., Biňasová, V., & Kovaríková, Z. (2025). Optimization of Business Processes Using Artificial Intelligence. *Electronics*, 14(11), 2105.
42. Rakyta, M., Bubeník, P., Binasova, V., Gabajova, G., & Staffenova, K. (2024). The change in maintenance strategy on the efficiency and quality of the production system. *Electronics*, 13(17), 3449.
43. Grznár, P., Papánek, L., Marčan, M., Krajčovič, M., Antoniuk, I., Mozol, Š., & Mozolová, L. (2025). Enhancing Production Efficiency Through Digital Twin Simulation Scheduling. *Applied Sciences*, 15(7), 3637.
44. Hrubaník, P. 2015. Zber výrobných údajov pre holonický výrobný systém. Dissertation thesis. UNIZA. Zilina.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.