



HAL
open science

Formal and modelling frameworks for social holonic control architectures

Etienne Valette, Guillaume Demesure, Hind Bril El-Haouzi, Rémi Pannequin

► **To cite this version:**

Etienne Valette, Guillaume Demesure, Hind Bril El-Haouzi, Rémi Pannequin. Formal and modelling frameworks for social holonic control architectures. *Computers in Industry*, 2021, 132, pp.103521. 10.1016/j.compind.2021.103521 . hal-03283782

HAL Id: hal-03283782

<https://hal.science/hal-03283782>

Submitted on 12 Jul 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Highlights

A Formal and Modelling Frameworks for Social Holonic Control Architectures

Etienne Valette,Guillaume Demesure,Hind Bril El-Haouzi,Rémi Pannequin

- Human integration is crucial for future manufacturing systems
- Social approaches for human integration are various but mostly techno-centred
- A holon can both represent human and objects or agents
- Any system could be represented as a society of socially related holons
- A Social Holonic Control Architecture framework could ease human integration

A Formal and Modelling Frameworks for Social Holonic Control Architectures

Etienne Valette^{a,b,*}, Guillaume Demesure^{a,b}, Hind Bril El-Haouzi^{a,b} and Rémi Pannequin^b

^aUniversity of Lorraine, CRAN CNRS UMR 7039, CMQ Bois Grand Est, F-88000, Epinal, France

^bUniversity of Lorraine, CRAN, CNRS UMR 7039, Campus Sciences, BP 70239, 54506 Vandœuvre-lès-Nancy cedex, France

ARTICLE INFO

Keywords:

Social Holonic Control Architectures
Industry 4.0
Social Relationships
Multi-Agent System
UML modelling framework

ABSTRACT

For decades now, manufacturing systems have grown in size and complexity. Between new consumption habits and hypercompetitive markets, manufacturing systems have started a race towards the industry of the future. Still, many technological and societal issues are paving their way: connectivity, resilience and human integration being among the most critical ones. The idea of this paper is to bring a new framework to help answering these issues by bringing resilience to systems, improving their interoperability, enhancing Data acquisition, transmission & processing, enabling the establishment hierarchical levels among agents, or by facilitating the system acceptance by human agents and the human integration within the system. To this end, this article brings a new formal framework for Social Holonic Control Architectures, based on an analysis of the existing literature. An UML-based modelling framework will equally be proposed to ease understanding and implementation of this new Social Holonic Control Architecture, illustrated by a concrete application on a small-scale Multi-Agent System.

1. Introduction

From PROSA in 1998 [1] to ARTI [2, 3] in 2019, the holonic manufacturing paradigm has continuously evolved for the last decades in order to answer industrial systems' adaptability and flexibility issues, as part of the Intelligent Manufacturing Systems (IMS) programme [4]. Its developments have brought concrete and encouraging results, notably thanks to successful implementations of Holonic Control Architectures (HCA) in industrial environments [5, 6, 7, 8, 9, 10, 11]. However, despite these results and manufacturers' strong needs, notably motivated by the march toward Industry 4.0, HCAs' deployment within manufacturing systems stays limited [12]. These applications therefore do not fully meet the needs of the latter yet.

This study and its conclusions highlight the importance taken by technological component into HCA-related research, and how this importance will keep growing with the rise of supportive technologies of Industry 4.0. Besides, despite multiple assertions of its importance [2, 12, 13], we can equally observe that human integration into HCA remains marginal in today's HCA field. As a matter of facts, this marginality is not only the prerogative of HCA, but is pretty observable in the IMS community's works [12] even though cooperation between human and technological entities has been widely studied, especially in air/rail traffic control [14, 15], applied robotics [16], or carpooling [17]. Concerning human-machine cooperation, many efforts have been provided to balance human-machine interactions, improve human-machine communication, or understand human behavior [18, 19]. In previous analysis [20, 21], these elements have led to the two following questions:

- [QR1] How to take advantage from the new concepts introduced by paradigms such as Industry 4.0 for future HCA developments?
- [QR2] How to get the human better integrated in future manufacturing systems, as socio eco-technic ones?

* This paper is an extended version of the paper *Toward a Social Holonic Manufacturing Systems Architecture Based on Industry 4.0 Assets*, presented at the 10th International Sohoma workshop on Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future, 2020

** The authors gratefully acknowledge the financial support of the CPER 2015-2020 Projet Cyber-Entreprises of Programme Sciences du numérique, through regional (Région Lorraine, Grand EST), national (DRRT, CNRS, INRIA) and European (FEDER) funds used to extend the TRACILOGIS Platform

✉ etienne.valette@univ-lorraine.fr (E. Valette); guillaume.demesure@univ-lorraine.fr (G. Demesure); hind.el-haouzi@univ-lorraine.fr (H. Bril El-Haouzi); remi.pannequin@univ-lorraine.fr (R. Pannequin)
ORCID(s): 0000-0002-0153-9118 (E. Valette)

By definition, a HCA is a recursive structure composed of collaborating holons, themselves being recursive, communicating, and decisional entities generally consisting of both an information and a physical-processing parts [22, 23]. From our observations, reference control architectures, such as PROSA and its different developments, are conceived on a modular bottom-up aggregation of categorized holons (Product, Resource, Order, Staff, or e-Person for example). The coordination of these holons only being achieved by exchanges of data sets from different nature (*Process knowledges*, *Execution knowledges*, and *Production knowledges* for example). Our idea is to propose an evolution of these holon's representation, by adding human-inspired social character to their classic coordination mechanisms. More precisely, we propose to use a human-inspired social relationship typology instead of considering classical hierarchical, heterarchical, or isoarchic relationships only. Our hypothesis is that this social approach might ease the design and implementation of future human-adapted HCA into actual manufacturing systems, as well as their understanding and consequent acceptability by human agents.

The main contribution of this article is the definition of these social relationships and the proposition of their formal framework. To go beyond this formal theoretical framework, a modeling and visualization software tool have been developed to instantiate the proposed framework to a small-scale Multi-Agent Systems (MAS), as a proof of concept. The remainder of the article is organized as follow. First, in Section 2, regarding the current HCA-related literature, we will study the place today given to "human" holons among artefact ones. In this part, we will identify the current dominant scientific approaches in holonic research regarding architecture implementation, the nature of relationships among their components, and the consideration brought to human dimension. Then, Section 3 will present the principles for a new metamodel supporting our vision of a social system, that aims to bring an answer to the two previous questions. We will provide a first glimpse of its structuration. Section 4 will expose the metamodel's formalization. Section 5 will bring an instantiation of the previous propositions. This instantiation will begin by the UML mapping of modelling elements, that will enable the development of a small-scale MAS simulation and monitoring software. In Section 6, we will detail more precisely the scenario runned in the simulation to finally bring in Section 7 some discussions elements.

2. HMS & HCA: What place for Humans?

The purpose of this section is not to conduct an exhaustive state of the art upon the place of the human dimension into current HMS and HCA-related research literature, but to provide an overview of the situation to identify the various existing orientations and gaps in the literature. We focused on 3 aspects of HCAs: 1) implementation aspect, 2) architecture components relationships' nature, and 3) human consideration. To get the most relevant literature, we were first interested in the two literature reviews led by Cardin, Derigent and Trentesaux between 2018 and 2020 [4, 22]. Regarding their contribution to future industrial challenges, they have identified, studied, and classified reference HCA developed from 1998 to 2020 [1, 2, 5, 6, 7, 8, 9, 10, 11, 24, 25, 26, 27]. To these, we added some more recent works, posteriors to the reviews: the REDCA and EMH² architectures [28, 29] and the holonic reengineering for CPS [30] propositions. From this first sample, 3 clear observations could be made:

- First: for each of these propositions, implementation is achieved by transposing the holonic system into a multi-agent one, which seems common sense in computer sciences. We note in particular the wide use of object-oriented modelling and of Java technology.
- Second: these approaches are seeking to define control architectures. Once holons have been defined, we are calling "relationships" the structuring elements relating them to each other, enabling the definition of an architecture. Concretely, these are commonly not developed beyond the notions of aggregation, hierarchy, or data-exchanges. Formalism is rather rudimentary, based on a direct abstraction of the studied system's components.
- Third: Human or social dimension are absent from these approaches.

To go further, we have led our own research. Once again, our scope was not to be exhaustive nor systematic, but to get a relevant glimpse of the literature. For this reason, we have limited our research to the *Web of Science* multidisciplinary bibliographic database. With the string "(TS=(holonic AND (control OR architecture* OR manufacturing OR system*) AND (social OR human OR anthropocentric))) AND LANGUAGE: (English)", we have searched the articles having either for subject HMS or HCA, considered with social, human, or anthropocentric point of view. Beyond that, we have restricted the research to English-written journal papers, conference papers, books, and book chapters. This research

has returned 110 results, spread from 1996 to 2020. Out of these 110 results, we have identified the 18 articles that seem the most pertinent after applying the methodology exposed in the Figure 1.

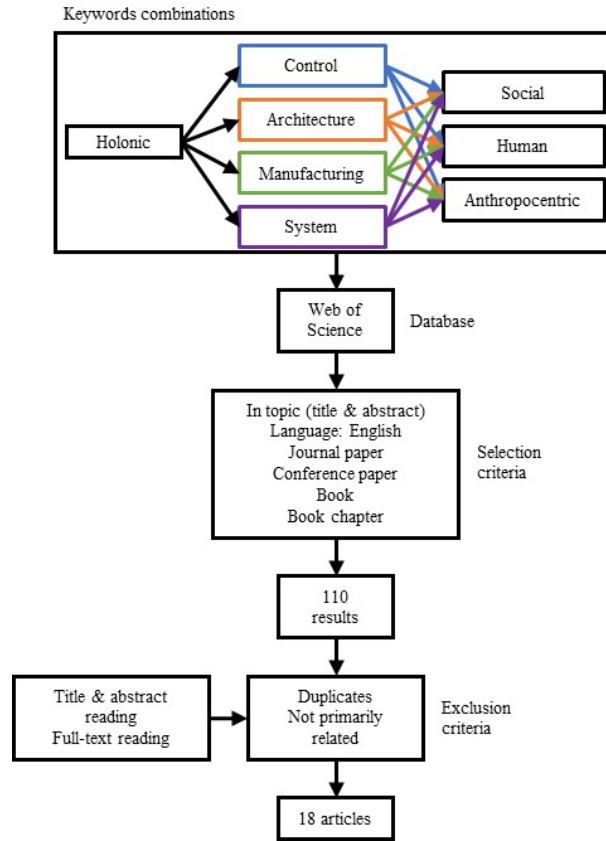


Figure 1: Research methodology

The analysis of this second sample, more heterogeneous for it is not focused on control architecture design, allows more nuanced observations:

- First: this sample is way more conceptual than the precedent. However, for works proposing an implementation method, once again we observe the use of object-oriented modelling and Java technologies, commonly used in MAS research field. Yet, it is interesting to note that MAS are not only used for implementation: several of the identified works are explicitly using them to define all or part of their model.
- Second: these works being not seeking to define control architectures, the notion of “relationship” is there more diversified than in the previous sample. Notably, if notions of aggregation, hierarchy, negotiation, and data exchanges are still found, notions of cooperation, collaboration, and symbiosis are equally used.
- Third: since the keywords “Anthropocentric”, “Human” and “Social” have been used in the search string, a greater consideration for human factor in the sample is necessarily observed. In our case, it must be noticed that all papers are not dealing with its specific integration into manufacturing systems. The next paragraph is developing this aspect more precisely.

In the results of the previous search, 5 distinct approaches can be identified. The first and most represented one mostly relies on techno-centered approaches for holonic systems destined to ease Human-Robot/Computer/System interfacing and cooperation [31, 32, 33, 34, 35, 36]. Between 1997 and 1999 has appeared a more inclusive vision, where human agents were as much as possible considered as full-part holons within the holonic models for system or architecture design [37, 38, 39, 40]. At this point, we wish to raise attention on an approach which is neither explicitly

human nor social-related: the transposition of biological structures & mechanisms to holonic manufacturing systems [41, 42]. Still, from the implications of these is emerging the addition of social relationships to enhance previous holonic developments and structure new HCAs, these relationships being either derived from works in sociology or based on social behaviors observed into, for example, ant-colonies [43, 44, 45]. Finally, a more recent philosophy can be identified, focused on Human-System Integration thanks to enhanced inter-agent interfacing and connectedness [46, 47].

From a global perspective, HMS and HCA-related works are part of the IMS initiative, that has been created as a supportive framework for the development of new manufacturing technologies enabled by industrial revolutions [48]. In this context, research related to the Cyber-Physical Systems (CPS) and Internet of Things (IoT) paradigms, commonly recognized as Industry 4.0's main pillars, naturally falls under the IMS scope. Hence, enlarging our vision to IoT and CPSs' last decades' developments and taking inspiration from them seemed to make perfect sense. Beyond that, thanks to the use of these technological paradigms, the large-scale implementation of holonic architectures is today made credible. In a recent work studying the social and human-centered approaches regarding IoT and CPS [49], it is stated that these concepts have evolved over the past few decades, aiming to bring a greater grasp of human factors, moving from the definition of very conceptual paradigms such as Cyber Physical Social System [50] to the implementation of increasingly complex architectures such as the Social Human-In-The-Loop Cyber-Physical Production System [51]. Authors have established that this evolution was mainly based on the addition of social characters in the development of systems and their architectures. A good illustration would be the Social Internet of Things (SIoT), fostering the idea of socializing objects of an IoT system [52, 53]. From this work, it seems that the expression of social character in the focused literature is either achieved technologically (man-machine interfaces, enhanced human, etc.) or in a techno-centered way (social paradigms transposed to communicating objects' networks, excluding humans). Ultimately, the authors noticed that, for works developing or involving social relationships, focus was set on the application aspect of the proposal, but no formalization was proposed for relationships.

From these literature analyses, the authors came to the following conclusions. First, Human-System Integration (HIS) has been for the last decades and is still today a hot topic in research, and holonic research field is no exception. Second, the structuration of any system and the understanding of its functioning is depending on the definition of its components and of the links between them i.e.: holons and their relationships. In literature, those two elements are generally defined, when they are, by abstracting components of a pre-existing system. Consequently, current literature is lacking a generic formal framework for architecture or systems representation and design. Third, beyond being a consequent inspiration source for holonic systems conception, the MAS paradigm is today the most suited and then commonly used implementation support for holonic research. Hence, what we would like to propose now and to develop in the rest of this article is a social formal framework resulting from our readings and thinking that would help answering the two questions enunciated in the introduction. We envision an HMS including the human operator as one of its holons. Instead of relying on Human-System Interfaces to enable this integration, we will rather exploit the idea of using social relationships to structure the system as a human-like society. This approach echoes the SIoT [54], but aims to extend it to human agents. What is proposed here is a new framework for social HCA. We will start by presenting a holonic view of the social relationships' principles they have established to link both human and artefact assets in our model.

3. Proposal for Social Holonic Control Architecture: principles

A production system is a particular socio-technical environment, where artefact and human assets are interacting in various ways. Our idea is that a good formalization of social relationships could ease the definition of holarchy between different assets/agents/holons, help defining trust and data-sharing levels, enable localization, coordination or even control between communicating entities. These might be means, for example, to solve systems' interoperability and reconfiguration issues in factory based CPS context thanks to machines self-recognition & self-reassignment (see "plugin & produce", from the PERFoRM project [55]). Aside from plugin & produce facilitation, a system structured by human-like social relationships might ease the system's acceptability by the operators and the integration of this last one within the system as one of its agents, and would help to bridge the gap between actual HCA researches and the paradigms pushed by the industry 4.0 [56, 57, 58, 59].

From an holonic point of view, as part of a social HCA, a holon is an entity being either a human or a thing, acting along with other holons constitutive of a system to the realization of its objectives. To reach these objectives, holons must be able to perceive a part of their environment and to control it by executing specific actions. In manufacturing control,

this could either be autonomously achieved by one single holon providing orders to low-level holons (hierarchical structuration), might require cooperation between several holons, planning their activities together to complete their common tasks (heterarchical structuration), or a combination of both (hybrid structuration) [60].

3.1. Holon model definition

The term "holon" has been proposed by Koestler in 1967 as recursive components of self-organizing social and biological systems [23]. In HCA research field, a holon is commonly considered as composed of a combination of both informational and physical processing parts [22]. Consequently, a system's set of holons can both be considered as a sub-system and as a full-fledged holon. For instance, a resource holon could be implemented/developed as a cyber physical system composed by sensors & actuators, and a software holon as a component of the decentralised MES that control them. Hence, the characteristics of holons will equally apply to the ecosystem, and reciprocally. To define those characteristics, we had to keep in mind the implementation aspect of our model. Hence, we strongly relied on the MAS literature, and the agent model that are commonly used for applications in holonic research.

In their literature review, Chin et al [61] have admitted an agent to be an autonomous software entity, situated into an environment, monitoring and responding to changes by itself or through communication with other agents to achieve goals. Agents being at the same time autonomous, social, reactive and proactive [62]. In our work, we used a logic-based-like architecture for the great flexibility and liberty it gives to develop the different components of an agent model. Indeed, those are based upon the symbolic representation and modelling of the agent's behaviour and environment. Concretely, possible internal and external agent/environment states can be represented by sets, while cognitive, measure and applicative functions are describing the agent functioning itself. Furthermore, while characterizing our holon model, we took a great inspiration from Ferber's conception, where a MAS is represented as a set of 6 components [63]: an environment "E" where objects "Ob" are located, whose active ones are appointed agents "A". Objects (and thus agents) are related by relationships "R". Agents are able to perform operations "Op" upon the objects, whose applications and consequences on the environment "E" are represented by operators. The holonic focus of our study naturally implies certain divergences from this model. Notably, we consider that the notion of "holon" includes those of "object" and of "agent". No further consideration is hence brought to the "active" or "inactive" aspect of those.

To get a better grasp of the impact of relationships' nature upon the holonic control architecture system, we turned to the control theory field. In closed-loop controlled systems, outputs are controlled to measure the effects of internal or external disturbances, and feedbacks are reinjected to correct inputs at each time. This enables to visualize the evolutions of the system concerning social relationships establishment and impact. Hence, the autonomous aspect of the holon was considered as being motivated by its ability to observe its environment and own state to constantly adapt itself in order to reach its objectives, and where outputs are controlling a part of the holon's global environment.

3.2. Inter-holons social relationships

What particularly caught our attention in the work from Atzori et al. [54] is the idea of structuring the IoT as "a social network of intelligent objects, bounded by social relationships" [64], inspired from Fiske's anthropological works [65]. A.P. Fiske is today commonly recognized as a reference anthropologist specialized in the study of human social relationships. He has notably established that any human society was organized according to 4 elementary forms of sociability, upon which is build the social fabric. Consequently, as a background for our proposal, we first analysed Atzori's inter-objects relationships' typology: *Parental Object Relationship (POR)*, *Ownership Object Relationship (OOR)*, *Co-Location Object Relationship (CLOR)*, *Co-Work Object Relationship (CWOR)*, and *Social Object Relationship (SOR)* [54]. While paying a close attention to these, we noticed that the 5 relationships established were not based on a same model to link two objects. POR and OOR are established directly among 2 objects, while SOR is conditioned by the pre-existence of an OOR between 2 objects, and then only occurs when the two owners come in touch. Hence, relationships might be *Direct*, established with no intermediaries between two objects, or *Emergent*, established among two objects through the existence of either a third or more objects (Fig. 2).

The case of CLOR and CWOR is a bit particular to us. In their construction, these two relationships are referring to emergent relationships, indirectly established between two objects. Yet, those are not emerging from relationships that objects maintain with a common third one. Their relationship comes from something they share: a common "location", or a common "work". In our model, those will either be parts of a relationship established among agents that share a part their data, behaviors, or capabilities, or, accordingly to the nature of the shared "work", to a relationship of hierarchy or dependence. Still, the need to distinguish these two relationships is making sense in the context they were

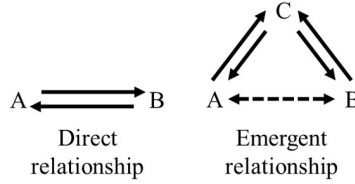


Figure 2: *Direct* and *Emergent* relationships between A and B

		$\mathcal{R}_{C \rightarrow A}^{Direct}$			
		PR	OR	DR	HR
$\mathcal{R}_{C \rightarrow B}^{Direct}$	PR	C-PR	POR	PDR	PHR
	OR	OPR	C-OR	ODR	OHR
	DR	DPR	DOR	C-DR	DHR
	HR	HPR	HOR	HDR	C-HR

Table 1
Typology of *Emergent* relationships

developed within.

Then, from the previous statements, we came with the idea that a social relationships typology should at first be built upon 4 fundamental *direct* relationships, besides echoing Fiske’s 4 elementary forms of sociability. Those are the *Parental*, *Ownership*, *Dependency*, and *Hierarchical* relationships (respectively PR, OR, DR, and HR). Since the existence of a “parent”, “owner”, “dependent”, or “superior” holon suggests the existence of a related “child”, “owned”, “required”, or “subordinate” one, we can consider that these relationships are not reciprocal. At this stage, we will note *Direct* relationships between 2 holons H_A and H_B : $\mathcal{R}_{A \rightarrow B}^{Direct}$.

Based on the combination of these 4 elementary relationships, we can conceive 16 emergent relationships, implying 3 holons each. Considering 3 holons H_A , H_B and H_C , we will note $\mathcal{R}_{A \leftrightarrow B}^{Emergent}$ the resulting *Emergent* relationship from the combination of $\mathcal{R}_{C \rightarrow A}^{Direct}$ and $\mathcal{R}_{C \rightarrow B}^{Direct}$ (Table 1). Hence, our typology would consist in a total of 4 *Direct* and 16 *Emergent* relationships, to which any relationship established among more than 3 holons could be reduced, considering the representation given by the Figure 2. For the sake of brevity, all the 16 emergent relationships are not exposed in detail in this study: reader will quickly notice that all of these are satisfying to the same model. Yet, 4 of them are remarkable, for they are based on the combination of similar direct relationships: *Co-Parental*, *Co-Ownership*, *Co-Dependency*, and *Co-Hierarchical* relationships (respectively C-PR, C-OR, C-DR and C-HR). Hence, these 4 relationships will receive a little more attention than the 12 others, which will simply be called in the rest of this paper *Social Contact Relationships* (SCR).

In the 4th section of this paper, we will begin with the characterization of the relationships and holons’ model. These will support the formal definition of the human-inspired social relationships exposed here above.

4. Proposal for Social Holonic Architecture formal framework

4.1. The holon formal model

We will consider the Social HCA ecosystem “E” as follow: a set of both human and artefact social holons “H”, which are socially related by a set of social relationships “R”, evolving through a set of states “X” (including desired objectives), due to a set of physical, informational, or energetic transformations “T” (Eq. 1).

$$E = \{H, \mathcal{R}, \mathcal{X}, \mathcal{T}\} \quad (1)$$

The control aspect of the holons constitutive of this ecosystem is represented on the Figure 3. A holon “ H_i ” is seen as an entity that evolves in and interacts with its environment “E”.

By being active, each holon H_i is acting on a part “ X_i ” of the ecosystem’s instant state. Hence, by defining “ \mathcal{X}_i ” as the set of all possible values of “ X_i ” that could be reached due to H_i (Eq. 2) and define the set of states of the ecosystem

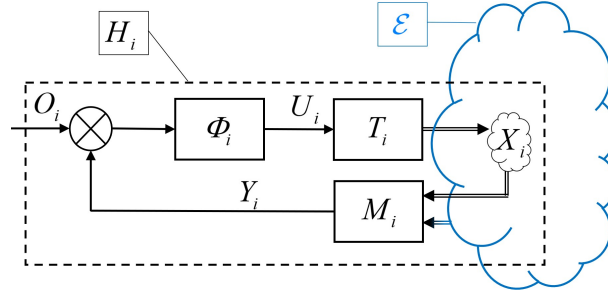


Figure 3: Control representation of holon H_i

as the set of all states that could be perceptible by its constitutive holons (Eq. 3).

$$\mathcal{X}_i = \{X_i\} \quad (2)$$

$$\mathcal{X} = \bigcup_{i \in \mathcal{H}} \mathcal{X}_i \quad (3)$$

These evolutions are motivated by a set of objectives " O_i " and enabled by a set of capabilities " C_i ". These capabilities include a cognitive/decisional function " Φ_i ", a transformation function " T_i " and a measurement function " M_i " (Eq. 4). The output " Y_i " represents the perception that holon H_i gets of the state of X_i and of the state of its environment $\{x\} \in \mathcal{E}$ thanks to the measurement function M_i (Eq. 5). The transformation function T_i represents H_i 's ability to change the state X_i accordingly to the input set " U_i ", representing the "*Intentions*" of the holon (Eq. 6). This input set is obtained by the processing of holon H_i 's objectives " O_i " by its cognitive function " Φ_i " considering its perceptions " Y_i " (Eq. 7).

$$C_i = \{\Phi_i, T_i, M_i\} \quad (4)$$

$$Y_i = M_i(X_i, \{x\}) \quad (5)$$

$$X_i = T_i(U_i) \quad (6)$$

$$U_i = \Phi_i(O_i, Y_i) \quad (7)$$

When facing an objective in a given situation, a holon might have different choices, and two holons might take different decisions. This will depend on the cognitive function " Φ_i ", that will generate to the holon its own tendency to provide an input U_i when facing an objective O_i considering its perception of its situation Y_i and its related Data set " D_i ". The Data set D_i related to H_i and involved in its cognitive function Φ_i is defined as the sets of some of its properties such as: its identification (ID), its type (e.g. product, resource, human ...), its different roles (e.g. planner, transporter, operator...), or its history (previous states, completed objectives...). Hence, we can finally represent a holon H_i by the Equation 8.

$$H_i = \{X_i, C_i, O_i, U_i, Y_i, D_i\} \quad (8)$$

For an holon H_i to effectively reach an objective $o_i \in O_i$, a process ρ_i might be required. A process is defined as a set of actions $\{a_k\}$ to be performed in order to make the system evolve from a current state X_i , towards a desired state X_d [62]. This process is modelled as follows (Eq. 9):

$$\rho_i = \{a_k\}_{X_i \rightarrow X_d} \quad (9)$$

The desired state X_d stands for the representation of the objective in the environment. The process ρ_i might be computed by the holon itself in order to reach its own objective or might be provided to this holon with the objective, depending on the holon capabilities to plan processes. Furthermore, the holon H_i might not be able to achieve some actions a_k planned in the process, which will require tasks delegation to other holons. Hence, we define \mathcal{U}_i as the set of actions that H_i can perform. These actions are selected by the cognitive/decisional function " Φ_i " and will lead to the transformation function " T_i ".

In the rest of this section, we will consider either 2 or 3 holons H_i, H_j & H_k , satisfying to the social holon model detailed previously (Eq. 8), with at any time $H_i \neq H_j \neq H_k$. In the proposed work, the recursive definition of a holon involves that as soon as one agent is existing, then a system is existing. Hence, any existing holon will necessarily be and be part of a system and will be assumed to be linked to any other holon by at least one social relationship. The next subsection will focus on the definition of the 4 elementary *Direct* social relationships and of a generic model for *Emergent* social relationships. Here we will bring highlight on the 4 emergent relationships that we have considered as remarkable in section 3. For brevity and convenience, will might replace “ H_i ” by “ i ” for the notation of holons in the next sections. Notably, the couple (H_i, H_j) will be written (i, j) .

4.2. Parental Relationship – PR

Two holons are in Parental Relationship (*PR*) when an holon i is parent of another holon j and where the child holon j inherits Data and/or capabilities from its parent i . We establish:

$$\begin{aligned} (i, j) \in PR &\iff i \text{ is parent of } j \iff j \text{ is child of } i \\ (i, j) \in PR &\iff (j, i) \notin PR \\ (i, j) \in PR &\iff \exists T_i : \mathcal{U}_i \longrightarrow H_j, (\{d_j\} \subseteq D_j \vee \{c_j\} \subseteq C_j) = T_i(a_i) \end{aligned} \quad (10)$$

where $PR \subseteq \mathcal{R}$ is the set of all Parental Relationships. The *PR* where holon i is parent of holon j being either noted:

$$\mathcal{R}_i \xrightarrow[\text{PARENT}]{PR} j \text{ or } \mathcal{R}_j \xrightarrow[\text{CHILD}]{PR} i$$

In a production system, PR relationships would appear when a product receive a distinctive transformation from a specific resource. For example, a drilling machine will be parent of a drilled product.

4.3. Ownership Relationship – OR

Two holons are in Ownership Relationship (*OR*) when an holon i is controlling and transforming a part of the state that holon j is acting on $\{x_j\} \subseteq X_j$. With the transformation T_i , holon i influences holon j 's cognitive function Φ_j , while the measurement function M_i enables holon i to perceive the state X_j . The Figure 4 proposes a control representation for Ownership Relationship.

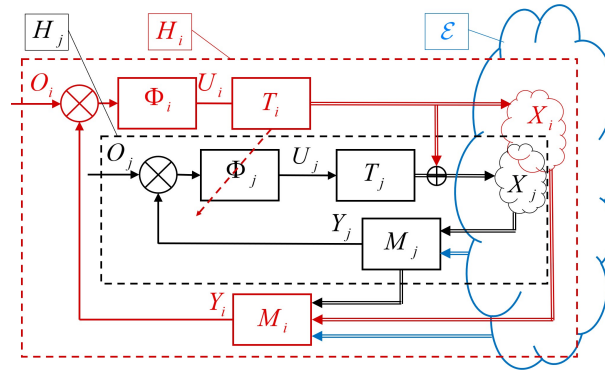


Figure 4: A control representation of $(i, j) \in OR$

We establish:

$$\begin{aligned} (i, j) \in OR &\iff i \text{ is owner of } j \iff j \text{ is owned by } i \\ (i, j) \in OR &\iff (j, i) \notin OR \\ (i, j) \in OR &\iff \exists T_i : \mathcal{U}_i \longrightarrow \mathcal{X}_j, X_j = T_i(a_i) \end{aligned} \quad (11)$$

And $\exists M_i : \mathcal{X}_j \times \mathcal{X}_i \times \mathcal{E} \longrightarrow \mathcal{X}, Y_i = M_i(X_j, X_i, \{x\})$

where $OR \subseteq \mathcal{R}$ is the set of all Ownership Relationships. The OR where holon i is parent of holon j being either noted:

$$\mathcal{R}_i^{OR} \xrightarrow{OWNER} j \text{ or } \mathcal{R}_j^{OR} \xrightarrow{OWNED} i$$

In a production system, OR relationships would appear when a product enters a resource to be processed. Once the product “loaded” into the resource, resource is physically controlling the product: the resource is then owner of the product.

4.4. Dependency Relationship – DR

Two holons are in Dependency Relationship (DR) when an holon j requires an action or a sequence of actions from an holon i to perform a task it has in its processes but is unable to perform. We establish:

$$\begin{aligned} (i, j) \in DR &\iff i \text{ is required by } j \iff j \text{ is dependent on } i \\ (i, j) \in DR &\iff (j, i) \notin DR \\ (i, j) \in DR &\iff \exists a \in \rho_j, (a \notin \mathcal{U}_j) \wedge (a \in \mathcal{U}_i) \end{aligned} \quad (12)$$

where $DR \subseteq \mathcal{R}$ is the set of all Dependency Relationships. The DR where holon i is required by holon j being either noted:

$$\mathcal{R}_i^{DR} \xrightarrow{REQUIRED} j \text{ or } \mathcal{R}_j^{DR} \xrightarrow{DEPENDENT} i$$

In a production system, taking the example of a Product-Driven System (PDS) [66], DR relationships would appear when a product needs to be processed by a resource. “Being transformed” is part of the process of the product, but it has no mean to achieve this by itself: the product is then dependent from the resource.

4.5. Hierarchical Relationship – HR

Two holons are in Hierarchical Relationship (HR) when an holon i provides an holon j order(s) to transform its environment and reach a desired state. To achieve this, holon j has the possibility to give order(s) to its own subordinates, to act on its environment, or to plan a process. The Figure 5 proposes a control representation for Hierarchical Relationship, where an order $o_j \in O_j$ is provided to holon j by an action from holon i , based on holon i 's objectives. The completion of this objective is notified to holon i as a feedback, according to holon j 's perception M_j . This way, holon j informs its superior of its perception of the objective's competition, which might be inaccurate.

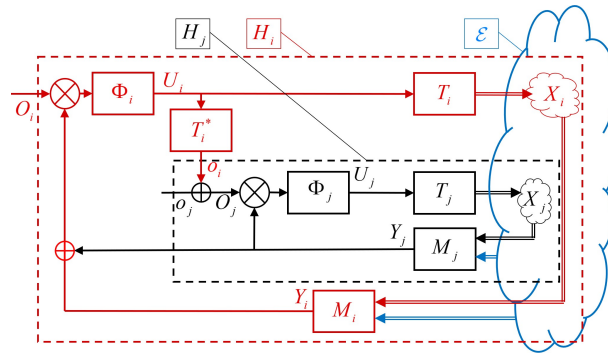


Figure 5: A control representation of $(i, j) \in HR$

We establish:

$$\begin{aligned} (i, j) \in HR &\iff i \text{ is superior to } j \iff j \text{ is subordinated to } i \\ (i, j) \in HR &\iff (j, i) \notin HR \\ (i, j) \in HR &\iff \exists T_i^* : \mathcal{U}_i \times O_i \longrightarrow O_j, o_j = T_i^*(a_i, o_i) \end{aligned} \quad (13)$$

where $HR \subseteq \mathcal{R}$ is the set of all Hierarchical Relationships. The HR where holon i is superior holon j being either noted:

$$\mathcal{R}_i^{HR} \xrightarrow{SUPERIOR} j \text{ or } \mathcal{R}_j^{HR} \xrightarrow{SUBORDINATED} i$$

In a production system, taking the example of a PDS, HR relationships would appear when a product gives the order to the resource to process it. Notwithstanding some constraints (queuing & processing rules, capabilities of the resource, etc.), the product is then the hierarchical superior of the resource in this situation.

4.6. Emergent Relationship

Two holons are in Social Contact Relationship (SCR), if they are each linked to a common third one by one of the 4 direct relationships. We establish:

$$\begin{aligned} (i, j) \in SCR &\leftrightarrow (j, i) \in SCR \\ (j, k) \in SCR &\iff \exists i \in \mathcal{H}, \langle (i, j), (i, k) \rangle \in DR \end{aligned} \quad (14)$$

With $DR \subseteq \mathcal{R}$ is the set of direct social relationships, and where $SCR \subseteq \mathcal{R}$ is the set of all Social Contact Relationships. The SCR between holon i and holon j being noted:

$$\mathcal{R}_{i \leftrightarrow j}^{SCR} \quad (15)$$

In a production system, SCR relationships are multiple. They appear when 2 agents of the system have a direct relationship with a same 3rd one, and the implications are varying accordingly to the nature of the direct relationships involved.

4.6.1. Co-Parental Relationship – C – PR

Two holons are in Co-Parental Agent Relationship ($C - PR$) when they have Data or capabilities in common with a third common holon with which they are in Parental Relationship. We establish:

$$\begin{aligned} (i, j) \in C - PR &\iff (j, i) \in C - PR \\ (j, k) \in C - PR &\iff \exists i \in \mathcal{H}, \langle (i, j), (i, k) \rangle \in PR \wedge \exists \{dc_{jk}\}, \{dc_{jk}\} \neq \emptyset \end{aligned} \quad (16)$$

With $\{dc_{jk}\} = (\{d_j\} \cap \{d_k\}) \cup (\{c_j\} \cap \{c_k\})$, where $(\{d_j\} \subseteq D_j, \{d_k\} \subseteq D_k, \{c_j\} \subseteq C_j, \{c_k\} \subseteq C_k) = T_i(a_i)$ and where $C - PR \subseteq \mathcal{R}$ is the set of all Co-Parental Relationships. The $C - PR$ between holon i and holon j being noted:

$$\mathcal{R}_{i \leftrightarrow j}^{C-PR}$$

In a production system, C-PR relationships would appear between parallel twin-resources (similar capabilities) or between products from the same production batch (similar data) for example.

4.6.2. Co-Ownership Relationship – C – OR

Two holons are in Co-Ownership Relationship ($C - OR$) when they belong to a same owner, with which they are in Ownership Relationship, but none of them owns of the other. We establish:

$$\begin{aligned} (i, j) \in C - OR &\iff (j, i) \in C - OR \\ (j, k) \in C - OR &\iff \exists i \in \mathcal{H}, \langle (i, j), (i, k) \rangle \in OR \wedge \langle (j, k), (k, j) \rangle \notin OR \end{aligned} \quad (17)$$

Where $C - OR \subseteq \mathcal{R}$ is the set of all Co-Ownership Relationships. The $C - OR$ between holon i and holon j being noted:

$$\mathcal{R}_{i \leftrightarrow j}^{C-OR}$$

In a production system, C-OR relationships would appear between two products on the same conveyor for example.

4.6.3. Co-Dependency Relationship – $C - DR$

Two holons are in Co-Dependency Agent Relationship ($C - PR$) when a third one is able to perform actions both of them have in their processes (common or not) that they are not able to perform. We establish:

$$\begin{aligned} (i, j) \in C - DR &\iff (j, i) \in C - DR \\ (j, k) \in C - DR &\iff \exists i \in \mathcal{H}, \langle (i, j), (i, k) \rangle \in DR \end{aligned} \quad (18)$$

Where $C - DR \subseteq \mathcal{R}$ is the set of all Co-Dependency Relationships. The $C - DR$ between holon i and holon j being noted:

$$\mathcal{R}_{i \leftrightarrow j}^{C-DR}$$

In a production system, C-DR relationships would appear between two products needing to be processed on a same resource for example.

4.6.4. Co-Hierarchical Relationship – $C - HR$

Two holons are in a Co-Hierarchical Relationship ($C - HR$) when they both receive at least one order (common or not) from a same third one. We establish:

$$\begin{aligned} (i, j) \in C - HR &\iff (j, i) \in C - HR \\ (j, k) \in C - HR &\iff \exists i \in \mathcal{H}, \langle (i, j), (i, k) \rangle \in HR \end{aligned} \quad (19)$$

Where $C - HR \subseteq \mathcal{R}$ is the set of all Co-Hierarchical Relationships. The $C - HR$ between holon i and holon j being noted:

$$\mathcal{R}_{i \leftrightarrow j}^{C-HR}$$

In a production system, C-HR relationships would appear between two resources receiving orders from a same product. For instance, in the case of a PDS: a robotic arm processing the product and the conveyor displacing it. Another example would be the case of an ERP providing instructions to two different products.

5. Modelling framework proposal

This section describes the modelling framework and its software application conceived to ease the modelling of an industrial system's social holonic architecture, using a small-scale MAS environment as a case study. Our goal here is to create a model-based software able to monitor a live MAS and to show in real time the social relationships that run through it. This proof-of-concept software will consist in 2 parts: the first one aims at developing modelling tool elements for MAS while the second one aims at developing a MAS-analysis software running upon the models produced by the modelling tool. Here, the instantiation of the notion of "holon" in a MAS pushes us to turn to the notion of agent, which will be used in the following sections.

The following subsections will first detail the mapping of the social holonic architecture formal framework concepts to an UML profile. To do so, the structure of the agents and of the relationships used in the model are detailed. The purpose is to provide enough knowledge to the proof-of-concept software (tool) concerning the multi-agent application to extract relationships between agents. This will be the subject of the last subsection, where the methods for visualizing and evaluating the social relationships' evolution within the MAS are exposed.

5.1. System and Agent 's structuration

The structure of the MAS can be represented by a class diagram: agent types are modelled by classes upon which "UML stereotype" "agent" is applied. Stereotypes are used to extend UML vocabulary in order to create new model elements with specific properties. Here, the creation and use of this stereotype enables the tool to associate an "agent instance" in the running MAS to an "agent type" in the model (Fig. 6). Agent classes are modelled by 2 types of properties: "Data" and "State" (in the model: D_i). We segment the "Data" property into:

- Identification data: any data that enable the agent to identify its informational or physical elements and bind them to him

- Objective data: the objectives set to the agent (e.g. for an agent controlling a product, the product’s bill of material)
- Process data: the agent’s knowledge concerning environment’s physical transformation abilities, their structuration and their request methods (e.g. in the example of a manufacturing application, the machines in the shopfloor associated with the agents that controls them)
- Record data: any kind of data generated during the agent life cycle, where a record of past events is stored

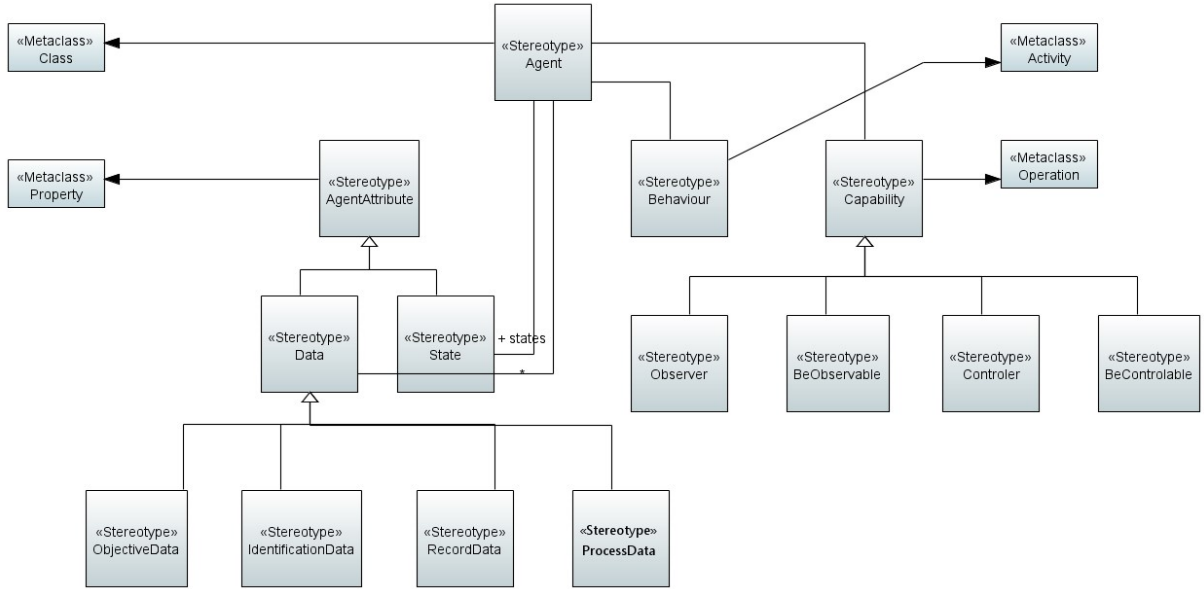


Figure 6: Agent’s profile diagram

The capabilities of agents are represented by methods, and its behaviors by activities (in the model: C_i). We modelled by activity diagrams the use of the agent’s capabilities by a behavior. Capabilities are classified using stereotypes into :

- Decide / Be Decidable
- Transform / Be Transformable
- Measure / Be Measurable

5.2. Social relationships modeling

Before establishing relationships among individual agents, we have first defined relationships between agent types. The relationships between agent types can be represented in composite structure diagram: relationships are modelled by “UML collaboration” upon which a stereotype corresponding to the actual relationship’s nature is applied (*PR/HR/DR/OR*, etc.) (Fig.7). In this work the elementary direct and emergent relationship previously detailed are modelled.

These generic relationship between agent types are used to set up the *actua* relationships between agent instances. Still, attention have to be paid when instantiating these. For instance, even if the agent types "Product" and "Resource" are related by a generic relationship, a specific product will not automatically be in relationship with every resource agent at all time. Relationships will only occur sometimes, in specific conditions and among specific product and resource agents. A relationship is therefore evolving through “states”. Hence, since relationships might become *active* or *inactive*, we developed the “RelationshipActive” and “RelationshipInactive” states, stereotypes. This switching process is described by a statechart where the “RelationshipActive” and “RelationshipInactive” stereotypes can be applied to states, to show that when entering such states, the relationship status change (Fig.7 & 8).

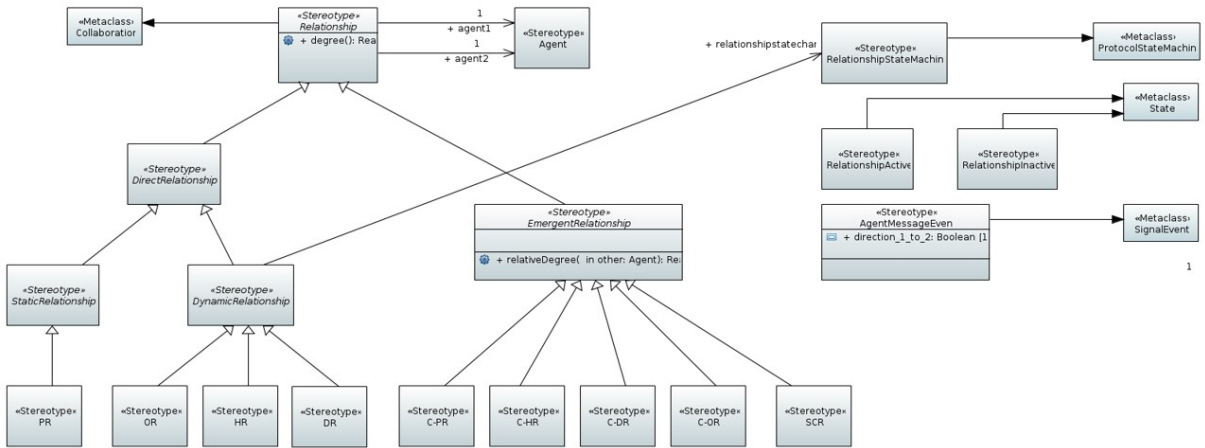


Figure 7: Relationship's profile diagram

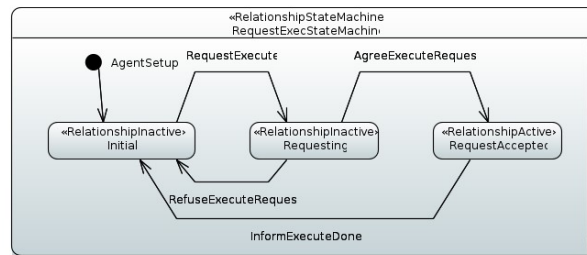


Figure 8: Example of relationship activity-triggering State Machine

The transitions between the different states can be associated to events happening in the system. These events' implementation depends on the actually used MAS. In our case, we used FIPA specifications for implementing the modelling process: events can be matched against the attributes of FIPA's ACL messages (e.g. performative, protocol, language, etc.), and also against the type of content of the message.

5.3. MAS-analysis software at runtime

Loading the UML model results in a set of objects representing agent types, and a set of "relationship factories"¹ that are used to instantiate the actual relationships between agents. A relationship factory contains the state machine defined in the model, that it instantiates when a new relation is created.

In this proof-of-concept application, the FIPA standard, implemented by the Jade library is used. FIPA, the Foundation for Physical Agent, is an international organization that is dedicated to developing specifications supporting interoperability among agents and agent-based applications. JADE (Java Agent DEvelopment Framework) is a software framework fully implemented in the Java language that complies with the FIPA specification and simplifies the implementation of multi-agent systems through a set of graphical tools that support the debugging and deployment phases. Using the same approach as for the "sniffer" tool of the Jade library, we are able to get notification of two kind of events:

- the creation and destruction of agents in the MAS
- the messages exchanged in the MAS between the agents

When a new agent instance appears in the system, the corresponding agent type is searched in the model. If this agent type is related to a relationship (as a source or a target), all corresponding agent instances are retrieved, and

¹In an oriented object language, factory is an alternative to constructor, that allows to create objects by instantiating various classes.

for each pair a new relationship is created. When a message is exchanged between agents, all relationships between the agent instances are selected. For each of these, the message is matched against transition filters, making the state machine evolve. When a direct relationship is created, all relationships of the same type with the same target are retrieved to create a co-relationship between the new relationship's source agent and each of pre-existing relationships' source agents.

The content of the system is represented by a multigraph where each agent instance corresponds to a node, and each active relationship to an edge. The style of the nodes can be set according to the type of agent; likewise, the style of edge reflects the type of relation. The graph is animated according to the stream of messages received. This animated graph is constituting one of the end results of the modelling tool for it enables the real-time visualization and tracking of relationships evolutions among agent within an actual MAS. The next section will present a scenario where these multigraphs and their evolutions will come as both examples and illustrations (Fig. 9 & 10).

6. Application scenario on a small-scale MAS

6.1. MAS components definition and initialization

This first test case consists in a simple MAS, inspired from the Tracilogis testbed from the CRAN's laboratory. It is composed of six manufacturing cells: a loading cell, an unloading cell, and four assembly cells, linked with a conveyor belt. Products are assembled with parts found in the assembly cells. There are five type of agents:

- The Directory Facilitator (DF) of the platform, to declare services and then find agents implementing these services (not represented in the example)
- The real or emulated agent that enables communication with the physical system ("emu" in the example)
- Resource Agents, controlling a manufacturing cell ("assy1" to "assy4" plus "exit" in the example, loading cell is not represented)
- Product Agents, representing a product ("Prod_1" and "Prod_2" in the example)
- Product launcher, that enable the user to easily creates product agents ("launcher" in the example)

In this system, we modelled four generic direct relationships:

- *HR* from the product to the resource, when the product requests the cell to execute some operation
- *HR* from the resource to the physical system, when the resource agent requests the physical system to do some actions
- *DR* from the product to the physical system, so that the product agent is notified of location events
- *OR* from the resource to the product when a product is inside a cell

No Parental Agent Relationship have been established in this first implementation, for no inheritance links is directly appearing to us (even if similar characteristics could be identified among the different agents). Hence, 3 types of direct relationships are involved in this application, along with their depending emergent relationships i.e. $C - HR$, $C - DR$ and $C - OR$. For convenience and to facilitate the lecture of the multigraphs in the example, *SCR* will equally not be tracked. Hence 6 types of social relationships will be observable.

The basic steps followed by an agent during its lifespan within the MAS are:

- Step 1: At creation, the product agent searches the DF to find resources and event sources
- Step 2: When notified that it is in front of a manufacturing cell, the product agent requires the corresponding resource agent to execute its next operation
- Step 3: If the resource agent is able to perform the operation, it agrees and sends a sequence of action requests to the physical system
- Step 4: If not, it refuses the request

- Step 5: When the product agent is finished, or if no operation can be executed in the cell, it requests the cell to release it

In our scenario, the initial system consists in the previously described agents. At this stage, the generic HR established between resources agents and the physical system are already created. Yet, although it is defined between these types of agents, its value (degree) is 0 because it does not actually exist at this moment. Consequently, no relationship is observable on the associated multigraph (Fig.9).

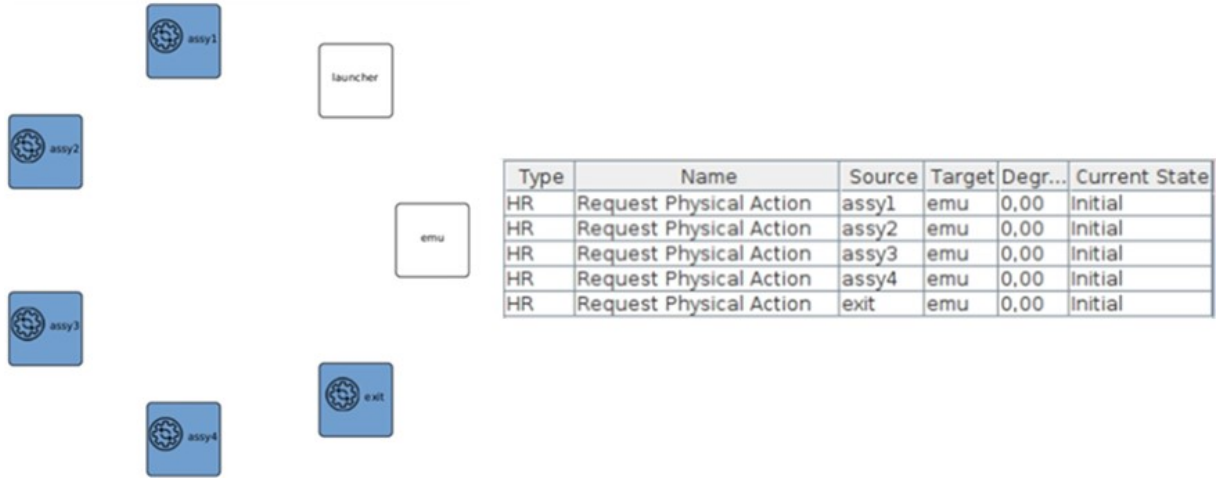


Figure 9: visualization of the initial state of the MAS

6.2. Scenario and visualization

First, a product agent “prod_1” is created through the Product Launcher. From its creation, prod_1’s generic relationships with its environment are generated: HR from the product to the resource, OR from the resource to the product, and DR from the product to the physical system. At the time of its creation, prod_1 has been given product type and the related information set and is in no relationship with any of the resource agents of the system, for it still has not been physically created (Degree = 0). Hence, it is only dependent from the physical system’s virtual representation “emu” that provides the agent the data corresponding to the physical agent’s perception of the physical system ($\mathcal{R}_{prod_1 \xrightarrow{DEPENDENT} emu}^{DR} = 1$). This relationship is represented by the arrow linking prod_1 and emu on the multigraph (Fig. 10).

Following the same method, a second production agent “prod_2” is created. At this stage, no relationship but $\mathcal{R}_{prod_1 \xrightarrow{DEPENDENT} emu}^{DR}$ and $\mathcal{R}_{prod_2 \xrightarrow{DEPENDENT} emu}^{DR}$ is existing into the system. Still, from the very existence of these two relationships is emerging a third one: $\mathcal{R}_{prod_1 \leftrightarrow prod_2}^{C-DR}$. This relationship is modelled by the dotted line on the multigraph. Hence, Just by their existence, two product agents are generating 3 active social relationships among the MAS (Fig. 11).

Once the agent has been created in the “cyber” part of the system, the “physical” parts of the product agents are introduced (event triggered by a “New Physical Product” button in the emulator). In this simple system, the pairing is done by associating the first agent with the first product, and so on (in an industrial system, an identification of the physical product would be required to create this association). Figure 12 represents the state of the system at this moment.

Once physically created, product agents are conveyed across the physical system. When arriving in front of a manufacturing cell, the product agent is notified by the physical system through their existing DR relationship (prod_1 depends on emu to know its location).

At this time, the product agent requests the resource agent to execute its operation. This request is an order provided by the product agent to the resource agent and is expressed by the creation of a Hierarchical Relationship between the two of them. In our scenario prod_1 becomes the superior of assy1, noted: $\mathcal{R}_{prod_1 \xrightarrow{SUPERIOR} assy1}^{HR} = 1$.

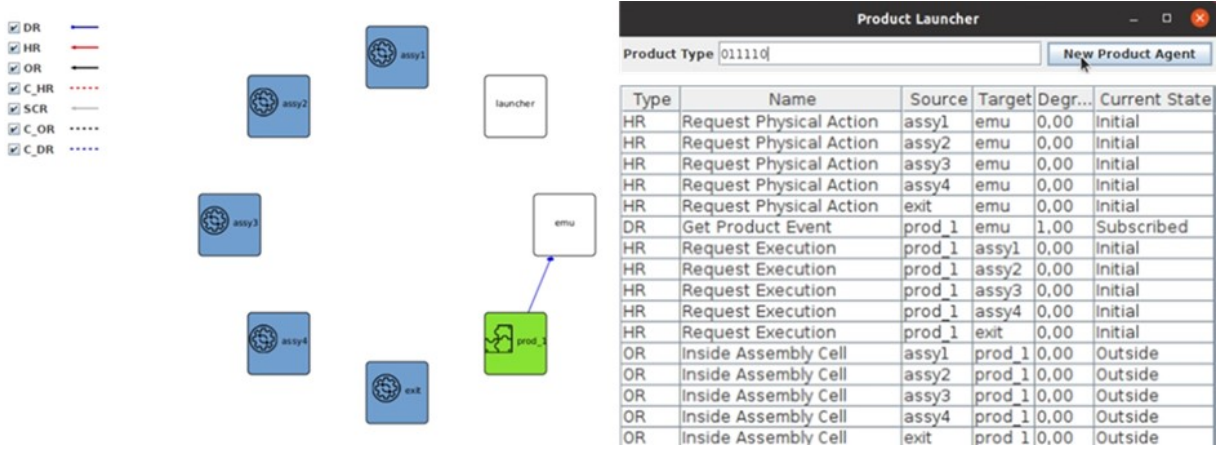


Figure 10: Visualization of the creation of a product agent

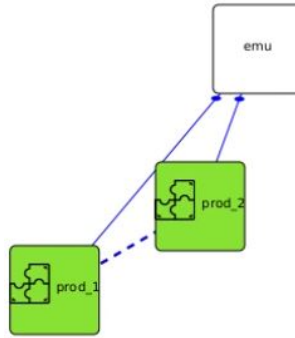


Figure 11: Relationships representation between 2 product agents and the physical system emulation agent

If the resource agent is able to perform the requested task, it sends a sequence of action requests to the physical system that will effectively perform the operation. The resource agent commands to the physical system and hence establish a Hierarchical Relationship with it. In our scenario assy1 becomes the superior of emu (as the representant of physical system), noted: $\mathcal{R}_{assy1}^{HR} \xrightarrow{SUPERIOR} emu = 1$.

Once this relationship has been established, a load event is triggered by the resource agent commanding to the physical system, so that the resource agent enters the manufacturing cell to be processed (here, operations st1 and st2 are realized). In our scenario, this last step creates an Ownership Relationship where assy1 becomes owner prod_1: $\mathcal{R}_{assy1}^{OR} \xrightarrow{OWNER} prod_1 = 1$.

At the same time, the product agent prod_2 has equally reached the assy1's manufacturing cell. Like prod_1, it is alerted of its position by emu, reaches assy1 to request execution (HR). In our study case, a resource agent is unable to physically "owns" two product agents simultaneously. Yet, it still is able to perform some operations simultaneously. For instance, assy1 is able to perform an assembly on prod_1 while providing a "go" for prod_2 to cross the zone without entering the production cell. Consequently, another action sequence request is sent from assy1 to emu, but no loading event is triggered. Then, only the two following social relationships are established: $\mathcal{R}_{assy1}^{HR} \xrightarrow{SUPERIOR} emu$ and $\mathcal{R}_{prod_2}^{HR} \xrightarrow{SUPERIOR} assy1$. At this precise moment, a new relationship is emerging. Since both prod_1 and prod_2 became superior to assy1, a Co-Hierarchical Relationship is established between the two of them i.e., $\mathcal{R}_{prod_1 \leftrightarrow prod_2}^{C-HR} = 1$.

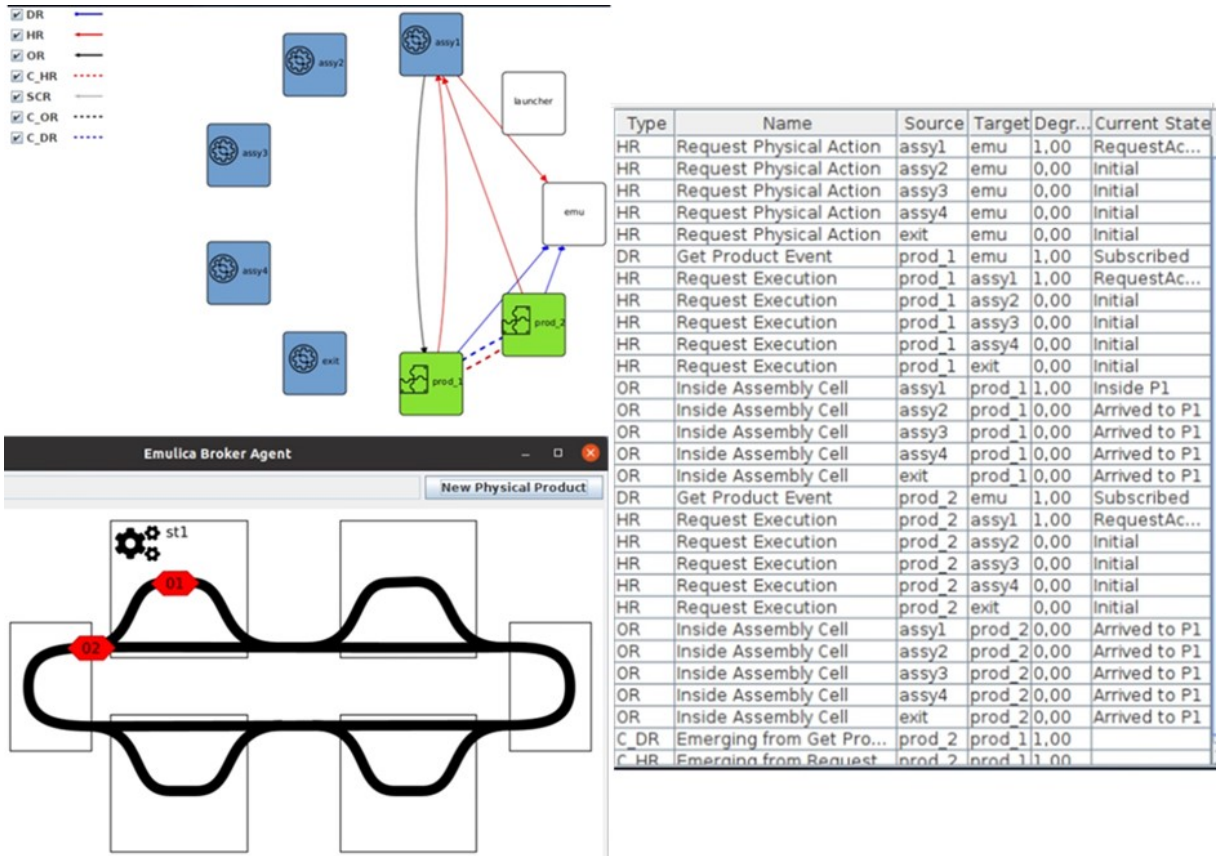


Figure 12: Visualization of the creation of social relationships during the simulation of a production process

7. Discussion and future work

We are defending the idea that any manufacturing system could be represented as a society of socially related holons. We therefore have developed a typology of social relationships inspired from works from various research fields. After defining a relevant representation for our holonic system and its constitutive holons, we have established a formal representation for each, and for the relations binding them. An UML implementation has then been forwarded, supporting a small-scale MAS-based production scenario simulation. This scenario successfully provided us clear representation of social-relationships establishment and evolutions among the constitutive agents of the system.

Future works should now aim at answering several points, starting by enhancing the model robustness. Although implementation on our small-scale MAS has shown encouraging results, no proof of the exhaustivity of our typology can be made. Plus, the double “translation” work made, from semantics to formal language, and then to UML modelling, came with its lot of difficulties and necessary bias whose impacts will have to be tried. Hence, we will have to confront our architecture to more and more complex and developed systems in order to validate or adjust it accordingly to its reactions to the situations it will face. For instance, we are planning to begin with a physical instantiation onto the TRACIOLOGIS testbed platform. New issues such as combinatorial explosion may then arise and will need to be answered.

Concerning the results obtained, since we were able in section 6 to show social relationships evolutions during our simulation of a production process, we were able to show the possibility for a holon to be related to one or more other ones by one or more social relationships of different nature. Notably, a holon can simultaneously be socially related to the other holons constitutive of its environment and to its environment itself (i.e., the upper recursive level of its system). This fact brings the possibility to establish, visualize and even characterize the existence of a plurality of holarchies i.e., of a multi-holarchy withing holonic systems. For these purposes, the authors already are working on the introduction of “relationships degree” that would reflect the strength of a relationship between two holons. Doing so,

the strength of a relationship among two agents or their confidence in each other could be measurable and modelled to enhance the framework thanks to technologies such as reinforced learning. This works will, on the long-term, help answering some of the issues raised by the emergence of the Industry 4.0 by bringing resilience to systems, improving their interoperability, enhancing data acquisition, transmission & processing, help defining trust and data sharing levels, enabling the establishment of confidence or hierarchical levels among agents, or by facilitating the system acceptance by human agents and the human integration within the system.

References

- [1] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, P. Peeters, Reference architecture for holonic manufacturing systems: Prosa, *Computers in industry* 37 (1998) 255–274.
- [2] P. Valckenaers, Arti reference architecture—prosa revisited, in: *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, Springer, pp. 1–19.
- [3] P. Valckenaers, Perspective on holonic manufacturing systems: Prosa becomes arti, *Computers in Industry* 120 (2020) 103226.
- [4] O. Cardin, W. Derigent, D. Trentesaux, Evolution of holonic control architectures towards industry 4.0: A short overview, *IFAC-PapersOnLine* 51 (2018) 1243–1248.
- [5] J.-L. Chirn, D. C. McFarlane, A holonic component-based approach to reconfigurable manufacturing control architecture, in: *Proceedings 11th International Workshop on Database and Expert Systems Applications*, IEEE, pp. 219–223.
- [6] P. Leitão, A. W. Colombo, F. J. Restivo, Adacor: A collaborative production automation and control architecture, *IEEE Intelligent Systems* 20 (2005) 58–66.
- [7] P. Pujo, N. Broissin, F. Ounnar, Prosis: An isoarchic structure for hms control, *Engineering Applications of Artificial Intelligence* 22 (2009) 1034–1045.
- [8] A. Le Mortellec, J. Clarhaut, Y. Sallez, T. Berger, D. Trentesaux, Embedded holonic fault diagnosis of complex transportation systems, *Engineering Applications of Artificial Intelligence* 26 (2013) 227–240.
- [9] C. Pach, T. Berger, T. Bonte, D. Trentesaux, Orca-fms: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling, *Computers in Industry* 65 (2014) 706–720.
- [10] F. G. Quintanilla, O. Cardin, A. L'anton, P. Castagna, A modeling framework for manufacturing services in service-oriented holonic manufacturing systems, *Engineering Applications of Artificial Intelligence* 55 (2016) 26–36.
- [11] J.-F. Jimenez, A. Bekrar, G. Zambrano-Rey, D. Trentesaux, P. Leitão, Pollux: a dynamic hybrid control architecture for flexible job shop systems, *International Journal of Production Research* 55 (2017) 4229–4247.
- [12] H. B. El-Haouzi, E. Valette, Human system integration as a key approach to design manufacturing control system for industry 4.0: Challenges, barriers, and opportunities, in: *17th IFAC Symposium on Information Control Problems in Manufacturing*, INCOM 2021.
- [13] P. Valckenaers, H. Van Brussel, *Design for the unexpected: From holonic manufacturing systems towards a humane mechatronics society*, Butterworth-Heinemann, 2015.
- [14] J.-M. Hoc, M.-P. Lemoine, Cognitive evaluation of human-human and human-machine cooperation modes in air traffic control, *The International Journal of Aviation Psychology* 8 (1998) 1–32.
- [15] F. Vanderhaegen, Pedagogical learning supports based on human–systems inclusion applied to rail flow control, *Cognition, Technology & Work* (2019) 1–10.
- [16] T. K. Haavik, Keep your coats on: augmented reality and sensework in surgery and surgical telemedicine, *Cognition, Technology & Work* 18 (2016) 175–191.
- [17] R. Pokam, S. Debernard, C. Chauvin, S. Langlois, Principles of transparency for autonomous vehicles: first results of an experiment with an augmented reality human–machine interface, *Cognition, Technology & Work* 21 (2019) 643–656.
- [18] F. Flemisch, D. Abbink, M. Itoh, M.-P. Pacaux-Lemoine, G. Weßel, Shared control is the sharp end of cooperation: Towards a common framework of joint action, shared control and human machine cooperation, *IFAC-PapersOnLine* 49 (2016) 72–77.
- [19] F. Flemisch, D. A. Abbink, M. Itoh, M.-P. Pacaux-Lemoine, G. Weßel, Joining the blunt and the pointy end of the spear: towards a common framework of joint action, human–machine cooperation, cooperative guidance and control, shared, traded and supervisory control, *Cognition, Technology & Work* 21 (2019) 555–568.
- [20] E. Valette, H. Bril El-Haouzi, G. Demesure, V. Boucinha, Toward an anthropocentric approach for hybrid control architectures: Case of a furniture factory, in: *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, Springer, pp. 145–155.
- [21] E. Valette, H. Bril El-Haouzi, G. Demesure, Toward social holonic manufacturing systems architectures based on industry 4.0 assets, *10th Workshop on Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future, SOHOMA 2020* <https://hal.archives-ouvertes.fr/hal-02959159>, 2020.
- [22] W. Derigent, O. Cardin, D. Trentesaux, Industry 4.0: contributions of holonic manufacturing control architectures and future challenges, *Journal of Intelligent Manufacturing* (2020) 1–22.
- [23] A. Koestler, *The ghost in the machine.*, Arkana Books, 1967.
- [24] P. Verstraete, B. S. Germain, P. Valckenaers, H. Brussel, J. V. Belle, K. Hadeli, Engineering manufacturing control systems using prosa and delegate mas, *Int. J. Agent Oriented Softw. Eng.* 2 (2008) 62–89.
- [25] J. Barbosa, P. Leitão, E. Adam, D. Trentesaux, Dynamic self-organization in holonic multi-agent manufacturing systems: The adacor evolution, *Computers in industry* 66 (2015) 99–111.
- [26] C. Indriago, O. Cardin, N. Rakoto, P. Castagna, E. Chacòn, H2cm: A holonic architecture for flexible hybrid control systems, *Computers in industry* 77 (2016) 15–28.

- [27] H. Cao, X. Yang, R. Deng, Ontology-based holonic event-driven architecture for autonomous networked manufacturing systems, *IEEE Transactions on Automation Science and Engineering* 18 (2020) 205–215.
- [28] A. Fadil, D. Trentesaux, G. Branger, Event management architecture for the monitoring and diagnosis of a fleet of trains: a case study, *Journal of Modern Transportation* 27 (2019) 169–187.
- [29] T. Mezgebe, H. B. El-Haouzi, Recursive hybrid control architecture to deal with reactivity in the context of industry 4.0, in: 13^{ème} Conférence Internationale de Modélisation, Optimisation et Simulation, MOSIM'20.
- [30] A. Martín-Gómez, M. J. Ávila-Gutiérrez, F. Aguayo-González, Holonic reengineering to foster sustainable cyber-physical systems design in cognitive manufacturing, *Applied Sciences* 11 (2021) 2941.
- [31] W. A. Alford, K. Kawamura, D. M. Wilkes, Human-directed local autonomy for motion guidance and coordination in an intelligent manufacturing system, in: *Architectures, Networks, and Intelligent Systems for Manufacturing Integration*, volume 3203, International Society for Optics and Photonics, pp. 81–86.
- [32] R. Kremer, D. Norrie, Architecture and design of a holonic visual interface, in: *Smc 2000 conference proceedings. 2000 IEEE international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions'* (cat. no. 0, volume 3, IEEE, pp. 1715–1720.
- [33] D. Kotak, S. Wu, M. Fleetwood, H. Tamoto, Agent-based holonic design and operations environment for distributed manufacturing, *Computers in Industry* 52 (2003) 95–108.
- [34] F.-S. Hsieh, Collaborative workflow management in holonic multi-agent systems, in: *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, Springer, pp. 383–393.
- [35] T. Żabiński, T. Mączka, Implementation of human-system interface for manufacturing organizations, in: *Human-Computer Systems Interaction: Backgrounds and Applications* 2, Springer, 2012, pp. 13–31.
- [36] A. R. Sadik, B. Urban, A holonic control system design for a human & industrial robot cooperative workcell, in: *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, IEEE, pp. 118–123.
- [37] E. Adam, E. Vergison, C. Kolski, R. Mandiau, Holonic user driven methodologies and tools for simulating human organizations, in: *Proceedings of the European Simulation Symposium, Passau, Germany*, pp. 57–61.
- [38] K. Toh, S. Newman, R. Bell, An information systems architecture for small metal-working companies, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 212 (1998) 87–103.
- [39] H. Sun, P. K. Venuvinod, The human side of holonic manufacturing systems, *Technovation* 21 (2001) 353–360.
- [40] J. Leuvenink, K. Kruger, A. Basson, Architectures for human worker integration in holonic manufacturing systems, in: *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, Springer, pp. 133–144.
- [41] B. Simsek, S. Albayrak, Living factory: back to koestler in holonic manufacturing, in: *IEEE International Conference on Industrial Informatics, 2003. INDIN 2003. Proceedings.*, IEEE, pp. 255–262.
- [42] P. Leitão, Holonic rationale and self-organization on design of complex evolvable systems, in: *International Conference on Industrial Applications of Holonic and Multi-Agent Systems*, Springer, pp. 1–12.
- [43] P. Valckenaers, H. Van Brussel, Holonic manufacturing execution systems, *CIRP annals* 54 (2005) 427–432.
- [44] E. Adam, R. Mandiau, Roles and hierarchy in multi-agent organizations, in: *International Central and Eastern European Conference on Multi-Agent Systems*, Springer, pp. 539–542.
- [45] A. Esmaili, N. Mozayani, M. R. J. Motlagh, E. T. Matson, A socially-based distributed self-organizing algorithm for holonic multi-agent systems: Case study in a task environment, *Cognitive Systems Research* 43 (2017) 21–44.
- [46] A. Bonci, M. Pirani, A. Carbonari, B. Naticchia, A. Cucchiarelli, S. Longhi, Holonic overlays in cyber-physical system of systems, in: *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, IEEE, pp. 1240–1243.
- [47] K. S. Gill, Holons on the horizon: Re-understanding automation and control, *IFAC-PapersOnLine* 52 (2019) 556–561.
- [48] H. Yoshikawa, Manufacturing and the 21st century—intelligent manufacturing systems and the renaissance of the manufacturing industry, *Technological Forecasting and social change* 49 (1995) 195–213.
- [49] E. Valette, H. Bril El-Haouzi, G. Demesure, L'humain dans les systèmes de production basés sur les paradigmes iot et cps: état des lieux et perspectives, 13^{ème} Conférence Internationale de Modélisation, Optimisation et Simulation, MOSIM'20. <https://hal.archives-ouvertes.fr/hal-03025467/document>, 2020.
- [50] F.-Y. Wang, The emergence of intelligent enterprises: From cps to cps, *IEEE Intelligent Systems* 25 (2010) 85–88.
- [51] C. Cimini, F. Pirola, R. Pinto, S. Cavalieri, A human-in-the-loop manufacturing control architecture for the next generation of production systems, *Journal of manufacturing systems* 54 (2020) 258–271.
- [52] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, H.-W. Gellersen, Smart-its friends: A technique for users to easily establish connections between smart artefacts, in: *international conference on Ubiquitous Computing*, Springer, pp. 116–122.
- [53] M. Roopa, S. Pattar, R. Buyya, K. R. Venugopal, S. Iyengar, L. Patnaik, Social internet of things (siot): Foundations, thrust areas, systematic review and future directions, *Computer Communications* 139 (2019) 32–57.
- [54] L. Atzori, A. Iera, G. Morabito, Siot: Giving a social structure to the internet of things, *IEEE communications letters* 15 (2011) 1193–1195.
- [55] PERFoRM, ????
- [56] R. C. Mayer, J. H. Davis, F. D. Schoorman, An integrative model of organizational trust, *Academy of management review* 20 (1995) 709–734.
- [57] M. Bobillier-Chaumon, M. Dubois, Technology acceptance and acceptability in organizations, *Le travail humain* 72 (2009) 355–382.
- [58] G. A. Boy, Design for safety: a cognitive engineering approach, *The Safety of Intelligent Driver Support Systems. Design, Evaluation and Social Perspectives*. Eds. Y. Barnard, R. Risser & J. Krems. Ashgate, UK ISBN (2011) 978–0.
- [59] T. Moulrières-Seban, D. Bitonneau, J.-M. Salotti, J.-F. Thibault, B. Claverie, Human factors issues for the design of a cobotic system, in: *Advances in human factors in robots and unmanned systems*, Springer, 2017, pp. 375–385.
- [60] O. Cardin, D. Trentesaux, A. Thomas, P. Castagna, T. Berger, H. B. El-Haouzi, Coupling predictive scheduling and reactive control in manufacturing hybrid control architectures: state of the art and future challenges, *Journal of Intelligent Manufacturing* 28 (2017) 1503–1517.

- [61] K. O. Chin, K. S. Gan, R. Alfred, P. Anthony, D. Lukose, Agent architecture: An overview, *Transactions on science and technology* 1 (2014) 18–35.
- [62] M. J. Wooldridge, N. R. Jennings, Intelligent agents: Theory and practice, *The knowledge engineering review* 10 (1995) 115–152.
- [63] J. Ferber, G. Weiss, *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 1, Addison-Wesley Reading, 1999.
- [64] D. Mala, *Integrating the Internet of Things Into Software Engineering Practices*, IGI Global, Online, 2019.
- [65] A. P. Fiske, The four elementary forms of sociality: framework for a unified theory of social relations., *Psychological review* 99 (1992) 689.
- [66] R. Pannequin, G. Morel, A. Thomas, The performance of product-driven manufacturing control: An emulation-based benchmarking study, *Computers in Industry* 60 (2009) 195–203.