doi:10.20944/preprints202111.0083.v2

Article

A Systematic Modelling Procedure to Design Agent-Oriented Control to Coalition of Capabilities - In the Context of I4.0 as Virtual Assets (AAS)

Jackson T. Veiga *, Marcosiris A. O. Pessoa *, Fabrício Junqueira, Paulo E. Miyagi and Diolino J. dos Santos Filho

University of São Paulo (USP); (jackson.veiga, marcosiris, fabri, pemiyagi, diolinos)@usp.br

* Correspondence: jackson.veiga@usp.br, marcosiris@usp.br; Av. Prof. Luciano Gualberto, 380 - Butantã, São Paulo - SP, 05508-010

Abstract: Manufacturing systems need to meet I4.0 guidelines to deal with uncertainty in scenarios of turbulent demand for products. The engineering concepts to define the service's resources to manufacture the products will be more flexible, ensuring the possibility of re-planning in operation. These can follow the engineering paradigm based on capabilities. The virtualization of industry components and assets achieves the RAMI 4.0 guidelines and (I4.0C), which describes the Asset Administration Shell (AAS). However, AAS are passive components that provide information about I4.0 assets. The proposal of specific paradigms is exposed for managing these components, as is the case of multi-agent systems (MAS) that attribute intelligence to objects. The implementation of resource coalitions with evolutionary architectures (EAS) applies cooperation and capabilities' association. Therefore, this work focuses on designing a method for modeling the asset administration shell (AAS) as virtual elements orchestrating intelligent agents (MAS) that attribute cooperation and negotiation through contracts to coalitions based on the engineering capabilities concept. The systematic method suggested in this work is partitioned for the composition of objects, AAS elements, and activities that guarantee the relationship between entities. Finally, Production Flow Schema (PFS) refinements are applied to generate the final Petri net models (PN) and validate them with Snoopy simulations. The results achieved demonstrate the validation of the procedure, eliminating interlocking and enabling liveliness to integrate elements behavior.

Keywords: RAMI4.0; Asset Administration Shell (AAS); Multi-Agent Systems (MAS); Evolutionary Assembly Systems (EAS); Engineering Capabilities Based, Production Flow Scheme (PFS); Petri Net (PN).

1. Introduction

Industry 4.0 (I4.0) drives global strategies in manufacturing systems to deal with turbulent product demands [1]. Traditional industry engineering tasks systematize each application: i) the development tools specify the project functionalities; ii) establish the necessary resources and, iii) formalize the process - determining the capabilities of the real assets (Skills). However, unpredictable behaviors can occur when the production phase starts, such as unavailability of resources or a new product insertion, causing a re-planning in production. [2,3].

In the current production paradigms, a rigid connection exists between the real asset functionality and digital services offered, but this approach does not meet the concepts of I4.0. [3,4].The [1] treats the need to modernize the processes involved in manufacturing with the use of appropriate ontologies that enable digital models. A digital representation of resources was presented in [5] which establishes guidelines for an I4.0 component (I4.0C). The I4.0 standardizes models to the information and communication between assets. They provide a library with different metamodels that offer semantics to enable intelligent manufacturing applications [2,3,6].

The I4.0 guidelines demonstrate a reference architecture (RAMI4.0 - Reference Architecture Model for Industry 4.0) that enables the design of digital asset models following a concept called Asset Administration Shell (AAS). I4.0 presents a capabilities submodel to specificity the capabilities of the resources [2]. The AAS models, suggest products based on manufacturing processes in which it is possible to choose the resources skills that best meet the product need [7]. However, the I4.0 AAS specifications use passive object concepts requiring a plugin for applications that allow the management of virtual resource capabilities [8].

This work intends to minimize the gap between emerging manufacturing solutions and intelligent manufacturing concepts by proposing a method to design intelligent control applications adherent to the I4.0 context. The proposal uses the concepts of multi-agent systems (MAS) following paradigms of Evolutionary Assembly Systems (EAS) [9,10]. This concept is related to the control of intelligent entities that impose self-organization, cooperation, and reasoning to fill the need to add intelligence to I4.0C [8]. These components are active objects that internally have knowledge and capacity to learn with the environment, seek new capabilities and, negotiate with the agent society [11].

The aspects described related to distributed intelligent systems (DAI), a specific area of Artificial Intelligence (AI) [12]. Intelligent agents are fundamental components for control, optimizing, improve and organizing the manufacturing process. The agents are helpful to model interactions between intelligent entities, which are responsible for attributing to the system characteristics such as cooperation, coexistence, or competition [13]. The use cases studied apply emerging methods to add intelligence in AAS; integrate the real and virtual component [14]; data management elements [15]; or enable the control of entities for reconfiguration [16]. This works proposal presents a process for modeling intelligent assets following the context of I4.0. The AAS describes capabilities and allows to share resources skills (Skill) as a society of agents that seek coalition through contracts to meet a specific production plan.

This work is an upgraded version of the work entitled “Control of Manufacturing Systems by HMS / EPS Paradigms Orchestrating I4.0 Components Based on Capabilities”, (Jackson T. Veiga; Marcosiris A. O. Pessoa; Fabrício Junqueira; Paulo E. Miyagi; Diolino J. Dos Santos Filho, 2021) presented at the 12th Advanced Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS 2021) held in Caparica (Lisbon), Portugal, in which the proposal is explained in more detail through a suggested system for modeling the activities that describe the relationship of AAS capabilities (Skill). The proposes use multi-agent systems to enable self-organization and resource skills coalition according to specific architectures, as presented in work [8].

The purpose of this paper is to describe a method for modeling AAS using concepts of multi-agent systems (MAS) for orchestration and cooperation of resource skills. This objective uses the paradigm following the EAS architectures and the concept of capabilities-based engineering to establish a systematic resource skills coalition. The chosen method implies the steps: i) technicals applied in the methodology; ii) formal description of the components with UML - Unified Model Language; iii) procedural description of the activities that relate the AAS and, iv) dynamic modeling of behavior with Petri Net (PN) enabling the validation of the proposed method.

2. Basic Concepts and Literature Review

This section presents the main tools used in the procedure, including I4.0 guidelines through AAS sub-modes and methods for modeling oriented to Petri Net (PN). The class diagrams (UML) specify the elements. The MAS control system is coupled to the RAMI4.0 architecture, delimiting the proposal’s scope. The discussion of related works extracts the work proposal contribution aspects.

2.1. Contextualizing Industry 4.0

The Fourth Industrial Revolution gained strength mainly with the German initiatives to modernize manufacturing systems, seeking to introduce traditional approaches to the context of Industry 4.0 (I4.0) [17]. The globalization demands proposals for integrating the entire manufacturing value chain, communication between machines, hierarchical levels, and applications involved in the production process, distributed production, and sharing of resources. The massive use of information and technologies is present in virtual manufacturing environments; however, significant efforts are committed to standardizing legacy applications to I4.0 [1].

The Manufacturing environments describe activities through discrete processes that cooperate with elements and applications to achieve a particular goal. In the traditional manufacturing systems:

- this cooperation provides a static and pre-defined model; however, the I4.0 characterized by uncertainty scenarios needs to change the engineering paradigms, allowing located and self-organize assets [2];
- the life cycle stages of a plant are well determined and known among Engineers; first are defined the basic processes for the production of a new product; second, engineering and documentation procedures describe how to perform the previous step; third is selected the necessary equipment from catalogs of manufacturers; the fourth step defines the plan of work, production order standards and test applications; fifth, it concerns the implementation stage, commissioning and delivery of the plant to the customer [3];
- in case of revisions, the processes must restart. The hardware (HW) and software are not updated. The equipment changes without any connection with the planning and, the different engineering tools don´t integrate information [3];
- there is a lack of tools to integrate the value chain, allowing detection in changes to the factory configuration.

The literature concerns the concept of Digital Twin (DT) as a possible way to solve this gap. This term was initially used by the National Aeronautics and Space Administration (NASA) in 2010 [3]. However, in the industry, it became known as a simulation model accurate and detailed that tries to reproduce the physical behavior, approaching the physical asset of the virtual asset [3].

In I4.0, the engineering paradigms for manufacturing activities change:

- it defines the contents and objectives (similar to the traditional factory); however, the (DT) performs the other actions;
- a digital model enables analysis and decision making by experts and planning engineers. All the data is stored containing requirements and functions in the "object" model enabling select the features from a library of functions;
- electronics manufacturers' catalogs conduce to choose all aspects and details of the equipment;
- instances of the model; this means select every "object function" and "equipment type" needed;
- occurs validation and simulation with the object models; generates the equipment orders, builds, tests, and put the plant in operation (digital world);
- the I4.0 model has generated automatically in run-time from the instance. Each component receives a default "administration shell". In this sense, the parts will be ready to communicate with the real automation solution and other plant components [3].

It is observed in this description the need to create the development flow of the I4.0 components. The I4.0 establishes this system using service-oriented paradigms or features, requiring libraries and specialist knowledge to build the different relationships between the models. At the current stage of the guidelines, the specific applications that define the framework for orchestration and control of various components allow exploring works in this direction.

2.2. *Virtual Representation in the I4.0 context with AAS*

The I4.0 context requires virtual resources guidelines [6]. A three-dimensional model, divided into layers for the treatment of information (including the life cycle and hierarchical levels of traditional systems), was introduced by RAMI4.0. This reference architecture standardizes and systematizes virtual resources through technical descriptions in I4.0C [7].

The I4.0 Component (I4.0C) is considered a proposal for global standardization, capable of communicating I4.0 assets. The I4.0C is composed of the Asset Administration Shell (AAS), and the Asset [6,7]. The Asset Administration Shell is the digital and active virtual representation of an I4.0C on the I4.0 system (Figure 1).

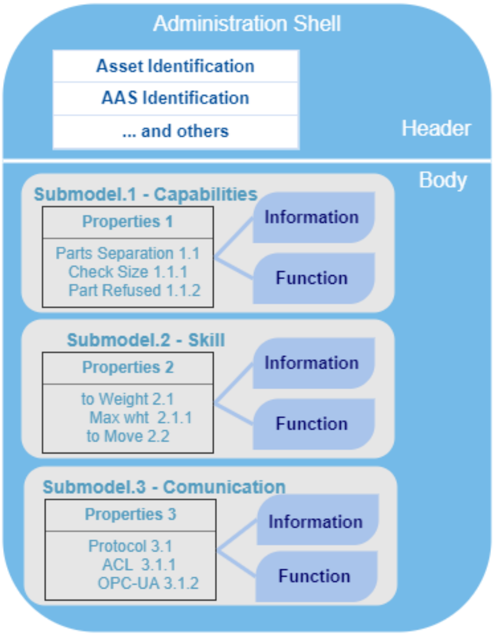

Figure 1. Global structure of the Administration Shell - AAS, adapted from [6].

The I4.0 context implements an AAS consisting of several submodels containing all the necessary information and functionality of a given asset. The AAS enables the asset to realize a dedicated set of use cases, including their features, characteristics, properties, status, parameters, measurement data, and describes capabilities. It allows different communication channels and applications and serves as the link between I4.0 components and the connected, digital, and distributed world [2,6].

The AAS, presented in [2], has the purpose of assisting this information through the design of sub-models specific for data standardization. Machines can interpret that providing a context to represent the features, and AAS allows different applications in particular domains. Since a consistent way of structuring information within the AAS is necessary, consult the guidelines to establish a metamodel to meet this need [6].

An AAS consists of a header and a body, as depicted in Figure 1. The header contains all information relevant to asset identification and AAS and, the body has all properties and operations that describe the asset. These descriptions are into sub-models that cover specific aspects of the active (e.g., functionalities, skills, and abilities)[2].

The conception of an AAS metamodel as a UML class diagram describes all the leading entities adopting the information structure. This work proposal has used these technicians to describe virtual assets using AAS. In this case, was chosen the following submodels:

a) Submodelelement, responsible for generating sufficient information for a specific case of use; b) Capabilities, demonstrate descriptions of the potential of an asset to achieve a particular effect on the physical or virtual world; c) Relationshipelement, used to define the relationship between two or more elements; d) AssetInformation, the identification of asset metadata is defined as represented by AAS.

These works propose the design of metamodels, PAS, and RAS as AAS designed with UML diagram presented in [2]. The extraction of minimum sub-models follows the I4.0 guidelines.

2.3. Review of the main concepts used to model the procedures

The Production Flow Scheme (PFS) technique supports the modeling system of this work derivated from Petri Net (PN) being a representation of discrete event systems (DES) [18]. These systems allow the systematic decomposition of activities and their flows at a more abstract level, unlike UML (Unified Modeling Language), a language for modeling software. The PFS understands a high-level language independent of technologies and manufacturers widely used to design system processes [19,20].

The proposed modeling system considered refinement techniques in PFS and simplifications until reaching the PN models (PFS/PN Technique) [20]. This technique makes it possible to analyze and specify the system and verify the behavior of the AAS components (Figure 2).

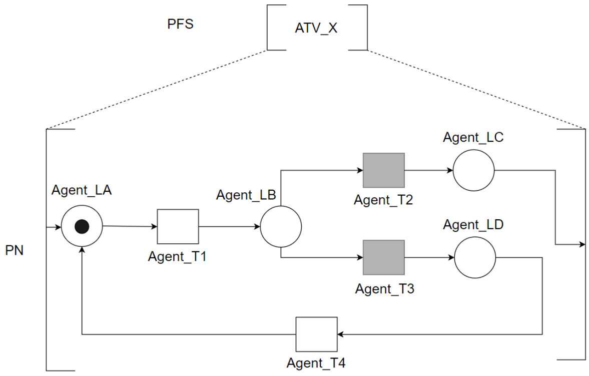

Figure 2. Example of PFS/PN Activity Flow Conversion.

This work considers the iteration between the AAS, compared to Discrete Event Systems (SED) using Petri nets (PN) modeling and simulation techniques to specify the procedure from the decomposition of the elements and the different relationships in the system [18].

The PN execution characterizes marks flowing in the process, which changes according to the transitions triggering. A transition triggers the previous marks, allowing the flow towards the arcs that connect the post-transition places. [19,20].

The PN models generated in this work proposal were validated using an engineering simulation tool that understands the PN syntax. Usually, academics explore applications like Platform Independent Petri net Editor (PIPE) or CPN-Tools. However, in this work, we chose to use the Snoopy tool to present some extensions to PN modeling. For example, the possibility of representing stochastic networks or the use of places and logical transitions. This tool simplifies the techniques for merging components, as is the case of transition merges used to model the message exchange between the components (agents) of the proposal [21].

Figure 3 demonstrates the main components of the Snoopy tool. Circles graph the places of PN, and rectangles represent transitions. In the proposal of this work, places represent passive components, intermediate spaces that allocate the result of message exchange and communication between agents, and transitions represent active components.

doi:10.20944/preprints202111.0083.v2

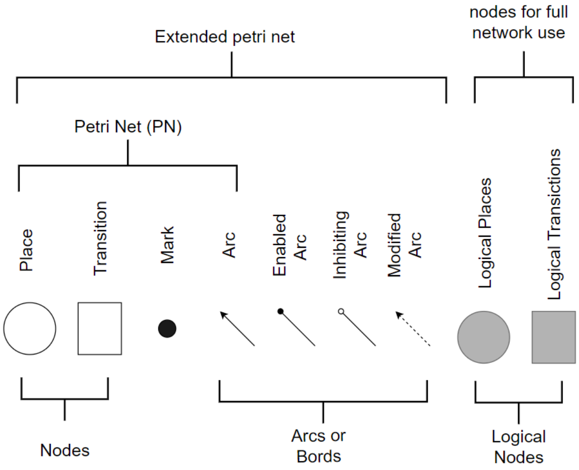

Figure 3. Main components of the Snoopy tool.

The transitions grouping between [ATVS] of the procedure is made possible with the extension of "logical nodes" or simply “merger nodes” for simulations in components (agents). These are similar to simple marks but presented in gray color (Figure 3).

2.4. Representing the proposal elements with UML

The modeling method of this work proposal seeks to describe a procedure for representing an intelligent application based on Multi-Agent Systems (MAS). The class diagrams (UML - Unified Modeling Language) illustrate these elements at a higher level of detail, typical in object-oriented projects (OOP). The context of I4.0 [21] provides some features for (Asset Administration Shell - AAS) modeled with (OO) to build systems that deal with the relationship and control of virtual entities (I4.0C).

The classes diagram was chosen in this works, being functional models for representing the individual components of the MAS control system [22]. Figure 4 exemplifies the simplified class diagram. The lines represent the relationships, treated as a dependency, association, or generalization (inheritance).

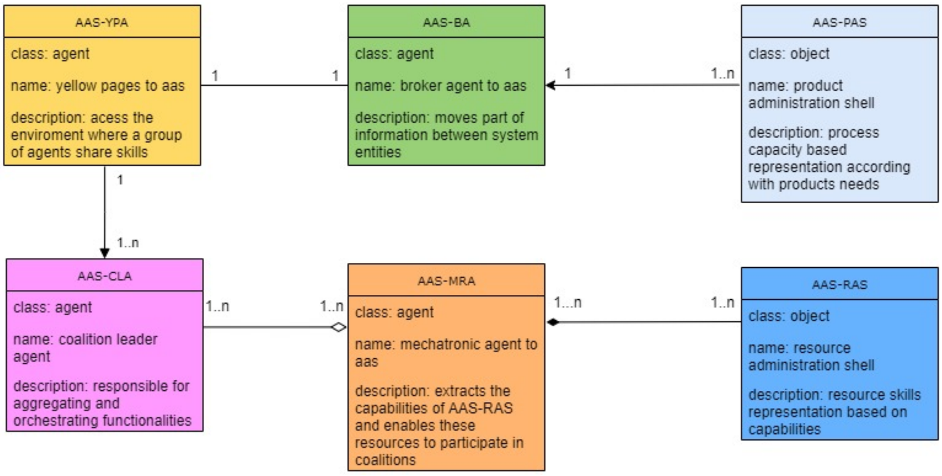
AAS-YPA
class: agent
name: yellow pages to aas
description: acess the enviroment where a group of agents share skills
AAS-BA
class: agent
name: broker agent to aas
description: moves part of information between system entities
AAS-PAS
class: object
name: product administration shell
description: process capacity based representation according with products needs
AAS-CLA
class: agent
name: coalition leader agent
description: responsible for aggregating and orchestrating functionalities
AAS-MRA
class: agent
name: mechatronic agent to aas
description: extracts the capabilities of AAS-RAS and enables these resources to participate in coalitions
AAS-RAS
class: object
name: resource administration shell
description: resource skills representation based on capabilities
1
1
1
1..n
1
1..n
1..n
1..n
1..n
1..n

Figure 4. Simplified Class Diagram for Proposal.

The descriptions presented on the class diagrams do not depict the behavior for individual entities; for this reason, it needs models representing dynamic behavior and relationships between agents. However, this work presents a proposal closer to models used in mechatronics engineering, making efforts to describe the activities of the elements using refinement techniques in PFS/PN [22].

The UML diagram details the internal elements of AAS-PAS following the guidelines of I4.0 [6]. The AAS-Products (WorkPlan) sub-model relates components that deal with procedural attributes to identify "Capabilities" necessary to execute the plan. In this diagram, a SubmodelElement lists the Capabilities described in (Product Administration Shell – PAS). The design of this diagram followed the I4.0 guidelines [2], so the main items considered were: (i) Asset Information, containing the standard identification (ID) and basic resource information; (ii) Sub-models, these make up the standardized objects to describe product work plans, to each product, elements of capabilities were related; (iii) RelationShipElement, describes access to data from the specified sub-models and also provides access standardized to this information through external entities (Figure 5).

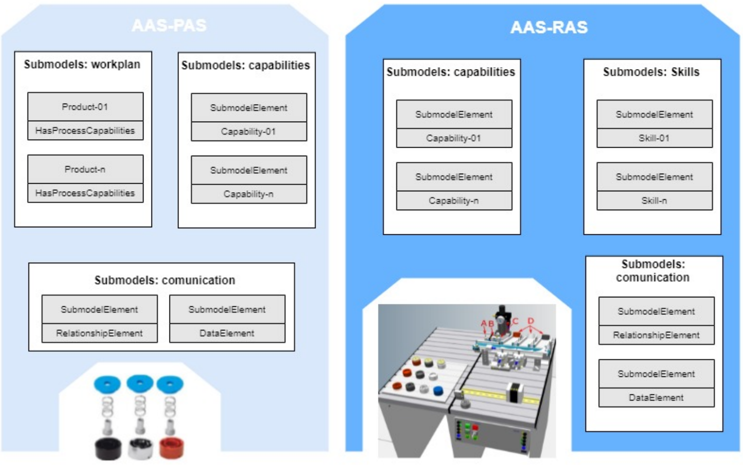

Figure 5. UML Classes Diagram for “Administration Shell – PAS”.

Like the Administration Shell of (PAS), the (RAS) object has its UML representation, which describes the capabilities elements. However, in this AAS, the sub-model Capability-n the (Skill) represents the real skills of the resources. These are elements that have attributes in data format; therefore, this object must be consulted by an agent (MRA) when specifying control of this proposal [8] (Figure 5).

The components presented in Figure 6 represent the control system coupled with the RAMI 4.0 to establish a consolidation concerning the main elements of the I4.0C layers are integrating the MAS self-organizing mechanism [8].

Figure 6 describes the main components of the architecture; (i) Assets Layer, responsible for representing the resources and devices of control; (ii) Integration Layer uses an “Agent Engine” responsible for integrating legacy systems and standardizing the communication of AAS adherent with the MAS context.

The two Layers described represents the real part of I4.0C, but this work applies more significant efforts in the virtual component of I4.0C:

(iii) Communication Layer, integrates traditional systems to components, controls described in the virtual environment, the (Agent Communications Language – ACL), establishes a standardization of communication in the MAS context.

(iv) Information Layer, this layer is composed of three databases to support the control strategies described in the functional layer, one for the Administration Shell (DAs), another to store real resource information (DRr), and finally, the Yellow Page Database (DYp).

(v) Functional Layer, specifies the MAS-based control mechanism to integrate AAS. It has two main components: a) Yellow Pages that follow the description of IADE and also understand a Cluster or location for publishing and feature consortia. b) Society of agents to coordinate and publish the skills (Skills) and c) Mechatronic Resource Agent

doi:10.20944/preprints202111.0083.v2

(MRA) as (Cognitive Agent), having social skills to express skills (Skill) described in the AAS and allow iterations with other agents members of the society [8].

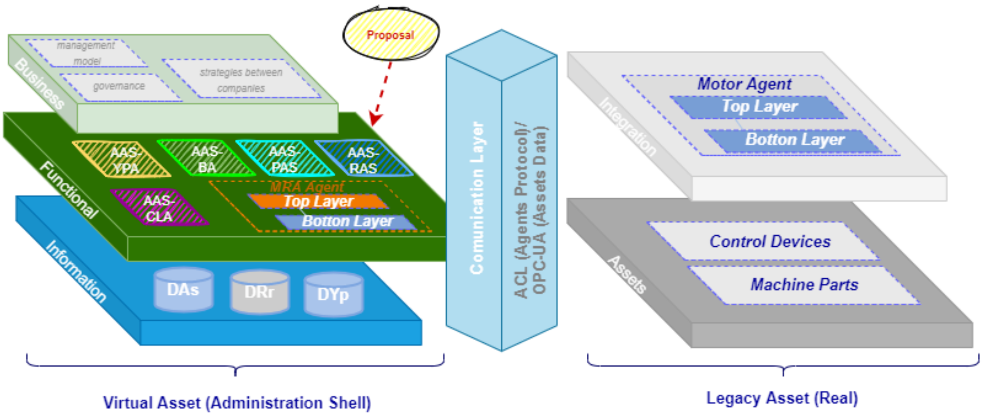

Figure 6. MAS Control System coupled to RAMI 4.0, adapted from [8].

2.5. Related Works about the Research

Different works provide ways of contributing to the implementation of the DT [1–3,7,11,18]; however, the proposals generally implement specific cases, such as searching for equipment, connecting assets to the cloud [1,23], digitizing PLCs [24] and, the tractability of assets [25]. However, it was impossible to notice works that describe a methodology to enable the concept of an active object in AAS, allowing it to understand its attributes and collaborate with the connected components to form societies in the connected world.

The work of [19] presented an architectural proposal to discover and select equipment that can process the operations requested by the products. This process takes place from iterations between assets in the physical world with virtual assets. The conception of this system establishes the coupling of an architecture based on services to the layers of RAMI4.0. This system contributes to developing I4.0 applications, presenting a technical path for integrating traditional hierarchical levels to its virtual component. This project contributes to the development of custom manufacturing that refers to product requirements.

However, this work is more focused on specifying service components for RAMI4.0 layers. It is currently considered a large number of legacy devices that need to adhere to the context of I4.0. Initially, these devices migrate to web services from cloud platforms. In this sense, the work of [23] describes an approach to integrating legacy devices to a system based on Cloud Platform (IoT - Internet of Things). This project applies virtual equipment to enable simulation and validation of control codes, attending step 3 of developing I4.0 systems. After verification, the real equipment is connected to the cloud, enabling the implementation of virtual models. However, the main objective of this proposal was to present protocols and means of communication between dumb devices and virtual tools in the cloud. It was not part of the scope of the work to develop a system for the control specification based on MAS that group Skills described in virtual assets in the context of I4.0 [7,8]. The article [26] proposes a decentralized and federated capacity-based AC (Access Control) mechanism to protect devices, services, and information effectively in large-scale IoT systems.

The [24] presents greater alignment with the stages of development of a production process in the context of representation of the Asset (Programmable Logical Controller - PLC) through the concept of Asset Administration Shell (AAS). The article shows an AAS metamodel to represent a PLC program and its relationships with other entities in the controlled plant. This approach fulfills the requirements of the I4.0, developing process regarding component virtualization and aligned with the concepts of Software Engineering. In the field of mechatronic engineering, the [18] presents a procedure to systematize the specification of virtual components and their functionalities in the production system, following the I4.0 guidelines.

In this context, the work of [14] presents a proposal to develop new control solutions based on I4.0 for legacy manufacturing systems. The work proposes the communication and integration of legacy manufacturing systems using AAS, IoT, and distributed architectures.

The proposal of this work is considered an extension of [7] which described a method to design a structure that allows an approach to control the capabilities of resources (Skill). The concept of capability-based engineering deals with[2]; it will enable resources to self-organize and resource reconfiguration given the (Single Batch Manufacturing). This concept allows identifying whether current resources can meet the demand for a new product or whether they need to determine which skills should add or replace to manufacturing.

Table 1 summarizes the main related works for designing control applications that demonstrate reconfiguration, cooperation, flexibility, or coalition of virtual component attributes.

Table 1. Comparing the reference projects characteristics studied in this work

Reference	Project	Paradigm	Main aspects
[2]	RAMI4.0 AAS	I4.0 SOA	Guidelines for AAS capabilities based in I4.0 context
[3]	DT I4.0C	I4.0 DOA	Plant Life Cycle, Digital Twin data set of AAS, data-objects
[6]	RAMI4.0 AAS	I4.0	Standardized Models, Architecture and references for I4.0 context
[7]	RAMI4.0 AAS	HMS EPS	Framework orchestrating AAS Intelligent Control - MAS, Skill Coalition
[8]	RAMI4.0 IADE CoBASA	I4.0 EAS MAS	Self-organization in complex systems, multi-agent, architecture capability checking and coalition
[13]	PERFoRM	MAS SOA	Generic architecture, reconfiguration legacy systems, distributed services IoT layers based and modularity
[5]	RAMI4.0 AAS	I4.0 SOA	Granularity abstraction for representation of functionalities, I4.0, DIN SPEC 91345
[27]	CoBASA	EPS MAS	Guidelines for MAS Control mechatronic virtual entities Skill ontology coalitions, contracts
[9,28]	FP7 IDEIAS	EPS MAS	Self-organization, self-ability tolerance to disturbances, MAS architecture

The reconfiguration and plug-and-produce follow the descriptions of the I4.0 [2,5]. The DT presented in [3] as the key to shifting the paradigms in manufacturing concepts. However, it is necessary to describe the relationship between assets, allowing them to search for data, using different protocols and languages that represent the I4.0Cs submodels with a closer approximation to the real Asset [5,7,8]. The realization of the DT will be possible with the cooperation of the different initiatives and technologies dealt with various proposals of the I4.0, which represent the cooperation between other areas, technologies, engineering, microelectronics, data science, information technology, and communication [6].

The [5] discusses the relationship between AAS, seeking to demonstrate different granularity levels of the resources' attributes and aggregation of assets functionalities. The sum of capabilities seen in various AAS can result in a new AAS, thus creating a new

I4.0C. These works [2,5] are in early stages, but the details presented on these concepts can be used for proposals that specify a methodology that allows the design of models to orchestrate the different relationships between AAS, such as a coalition of resource capabilities. In this sense, the work [8] performs architectural specifications that make it possible to implement the concepts of self-organization, plug-and-produce in the context of I4.0. The complex evolutionary systems seem to be the state-of-the-art guidelines to grant cooperation and grouping of resource abilities.

The works [2,5,6] contribute to the proposal of this work with ontological descriptions to address the abilities of resources using Engineering concepts based on AAS capabilities and standardized models in the context of I4.0. In addition, MAS enables the society and reasoning of components. In the related works, there was no objective evidence of active objects applying cooperation, relationship and describing methods for the coalition of component attributes in the context I4.0 as presented in [7,8]. Therefore, the proposal of this work considers characteristics such as the coalition, reasoning, and self-organization with MAS implementing intelligence, [9,13,27,28], however, it does not assign these attributes to the context of the I4.0 where [2,5] describe the modeling of I4.0 objects through the concept of AAS capabilities and models.

This work describes a systematic modeling method to design agent-oriented control seeking to organize Skills related to AAS. This method will apply to virtual entities, coalitions, and the self-organization of assets.

3. Proposal of a Conceptual Modeling of Activities for Procedure Creation of Agents in AAS

Modeling the MAS-based control system that seeks to address the gap for organizing resource capabilities without choosing these resources previously in the production order (PO) is described in this section employing a procedure that allows the specification and definition of component activities in Administration Shell. In this sense, the related elements seek to incorporate a self-organization mechanism, carrying out negotiations and forming coalitions, being regulated by virtual model capacity contracts (AAS).

The procedure division for modeling the proposal:
i) Problem definition delimiting the project scope; ii) Detailing of components designed specifying architectural characteristics specific for integration into RAMI4.0; iii) Diagrams detailing the relationship between the components of the proposal through activities encapsulated in the RAMI4.0 information; iv) Successive details for modeling the process components seeking to represent the procedures for a coalition of capabilities using refinement techniques and, v) Analysis and validation of models conceived in the proposed procedure using simplification until achieving the PN.

The Table 2 the processes handled in which each component (agent) receives [Activities]. The components and [Activities] models design the control system procedure for capabilities coalition.

The AAS are proposed for each component of the architecture: (i) AAS-RAS that, "External Skill IDs and Skills"; (ii) AAS-PAS, "Request a New Consortium Application"; (iii) AAS-MRA, "Request Capability IDs", "Record input for Adhesion to (CAC)"; "Performs Adherence to Coalitions" and "Search for Capabilities of Resources"; (iv) AAS-CLA, responsible for “Member Search for (MCC)” and “Coalition Mechanism”; (v) AAS-BA, transports to "Publication of the Request", "Capabilities Identification", "Iterations for Coalitions of (MCC)"; (vi) AAS-YPA, manages cluster activities by performing, "Definition of Terms (CAC)", Search of Candidates’ Credentials", "Intention to Adhesion to the (CAC)", "Contract Requests (MCC)", "Execution of (CAC)" and "Complex Skill Mechanism" Table 2.

Table 2. Descriptions of the activities and assignments of each object or agent

Component	Main Activities Description
AAS-PAS - (Product Administration Shell)	[01 - New Consortium Order]

AAS-BA - (Agent Broker Administration Shell Broker)	[02 - Order Publication];
	[05 - Identification of Capability IDs];
	[11 - Iterations for Coalitions of (MCC)].

AAS-YPA - (Agent Yellow pages Administration Shell)	[03 - Definition of Terms (CAC)];
	[04 - Search for Candidate Credentials];
	[09 - Intention of Adhesion to (CAC)];
	[10 – Contracts (MCC) for Execution of the (CAC)];
	[17 – Complex Skill Mechanism].

AAS-MRA - (Agent Mechatronic Resource Administration Resource Shell)	[06 - Capabilities ID Queries],
	[08 - Register the (CAC) Adhesion];
	[13 - Adhesion to Coalitions];
	[14 – Search for Resource Capabilities];

AAS-RAS - (Resource Administration Shell)	[07 - External Capabilities IDs]; [15 - External Capabilities (Skill)]
AAS-CLA -(Agent Coalition Leader Administration Shell)	[12 – Member Search for (MCC)]; [16 - Mechanism of Coalitions].

The detailing of the [Activities] specified in the procedure to meet the proposal's control architecture is carried out using PFS diagrams and rules of refinement seeking to reach the suggested model for simulation in Petri Net (PN). Figure 7, generalizes the flow of activities represented in each of the respective AAS that make up the society for resource capabilities coalition.

The process begins with the identification of a "New Consortium Request" ([ATV-01]). The AAS-PAS, after receiving a new request from the "traditional system ", sends an event to AAS-BA, who is in charge of preparing the "Publication Order" ([ATV-02]).

The second activity aims to insert the request for preparation and "Definition of Terms (CAC)" ([ATV-03]), this activity being the responsibility of the AAS-YPA. After the trial of the (CAC) is completed, the next activity will be the "Candidate Credential Search" ([ATV-04]). This activity enables the creation of tables containing the capabilities IDs identification of the resources found to participate and execute the "manufactured plan".

The AAS-BA brokers this request by performing the activity "Identify IDs of Capabilities" ([ATV-05]). The execution of this task enables an initial AAS-MRA that will carry out the "Query of Capacity IDs" ([ATV-06]) in AAS-RAS.

These procedures enable to access the capability objects in AAS-RAS through the AAS-MRA agent. This step simplifies the "External Capabilities ID" ([ATV-07]). The respective IDs are then registered in the "Cluster" by the AAS-YPA that releases AAS-MRA so it can "Register the entry for adhesion of the (CAC)" ([ATV-08]).

The activity ([ATV-09]) means the "Intention to adhere to (CAC)", which is registered by AAS-YPA. That last iteration of activities enabled the registration of members who will compete for a coalition.

The agent society can go to the next step that concerns the request for "Contracts orders (MCC) for the execution of the (CAC)" ([ATV-10]). It is an activity that initiates the contract search request (MCC) for AAS-BA. This agent, in turn, starts the request for iterations of the society of agents with "Iterations for MCC Coalitions" ([ATV-11]). The coalition establishes, the contracts (MCC) need to be signed, and an AAS-CLA call performs it. This agent assumes the role of initial leader for coalitions, performing the "Member Search for (MCC)" ([ATV-12]).

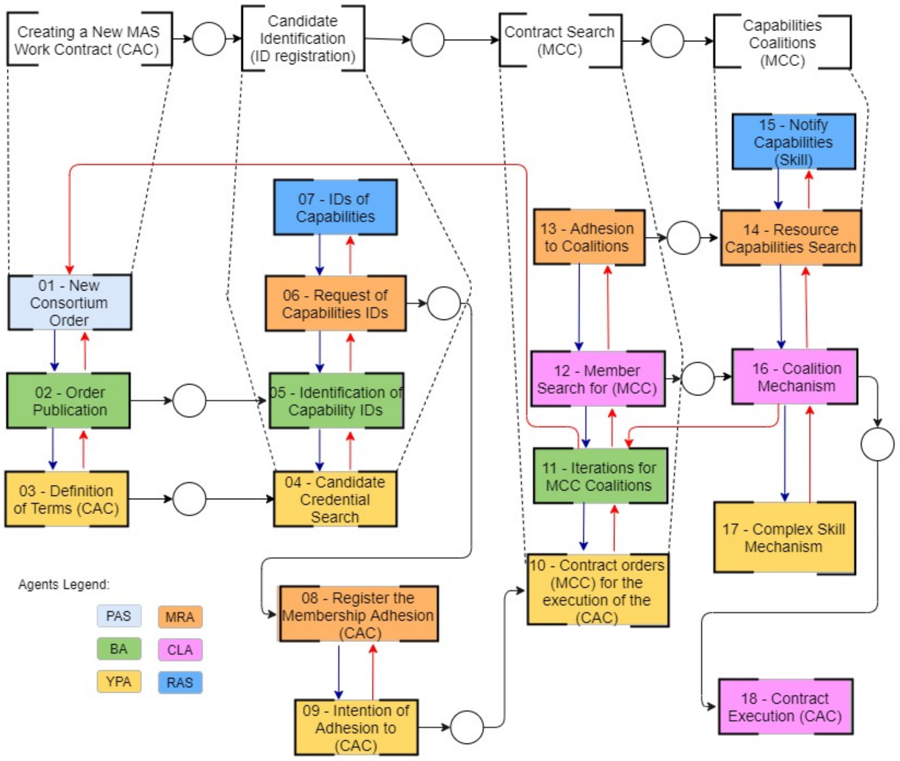
Creating a New MAS Work Contract (CAC)
Candidate Identification (ID registration)
Contract Search (MCC)
Capabilities Coalitions (MCC)
15 - Notify Capabilities (Skill)
07 - IDs of Capabilities
13 - Adhesion to Coalitions
14 - Resource Capabilities Search
01 - New Consortium Order
06 - Request of Capabilities IDs
12 - Member Search for (MCC)
16 - Coalition Mechanism
02 - Order Publication
05 - Identification of Capability IDs
11 - Iterations for MCC Coalitions
03 - Definition of Terms (CAC)
04 - Candidate Credential Search
17 - Complex Skill Mechanism
10 - Contract orders (MCC) for the execution of the (CAC)
Agents Legend:
PAS
MRA
BA
CLA
YPA
RAS
08 - Register the Membership Adhesion (CAC)
09 - Intention of Adhesion to (CAC)
18 - Contract Execution (CAC)

Figure 7. Flow of activities for the proposed models.

This activity has a mechanism capable of delimiting the AAS-MRA that can act together. For this, the AAS-MRA establish a negotiation with AAS-CLA through "Adhesion to Coalitions" ([ATV-13]).

It is considering that the requirements for contract (MCC) were met, in the next block of activities, AAS-MRA does the "Resource Capabilities Search" ([ATV-14]). In this third block of activities, AAS-MRA will search for capabilities that meet the criteria for the (MCC). Therefore, this executes the request for capabilities in AAS-RAS, enabling it to perform the activity "Notify Capabilities (Skill)" ([ATV-15). Each AAS-RAS related runs an internal capabilities routine and makes it available to the AAS-MRA to decide whether it will be possible or not to establish the coalition with (MCC) brought in by its lead agent AAS-CLA.

Finally, if a respective AAS-MRA answers the call, the AAS-CLA agent executes the "Coalition Mechanism" ([ATV-16]). Note that a mechanism to establish partnerships among the AAS-RA, provides the capabilities of resources brought to the coalition as "simple" type; otherwise, it will be necessary to invoke AAS-YPA to execute the activity "Complex Skill Mechanism" ([ATV-17]) looking for new skills (Skill). Whereas the latter activities have good behavior, that is, complete the "Mechanism for the coalition", the society of agents will be ready for "Contract Execution (CAC)" ([ATV-18]), a superior AAS-CLA agent monitors this activity.

4. Results

This section presents the method used to validate the supported procedure by PN diagrams. In this sense, the individual activity models demonstrate the "Capabilities Coalition (MCC)" process.

4.1. Functional Modeling of the Procedure using a Petri Net - PN to create the AAS-Agent based activities

The activity [14- Search for Resource Capabilities] resulted in the PN of Figure 8. This process begins with the request for (CLA) to search for capabilities, and the transition (MRA-T18) represents the activity [Request search for capabilities]. Then, the place (MRA-LA) enables the transition (MRA-T19), which is the representation of the activity [Receive Capabilities in RAS].

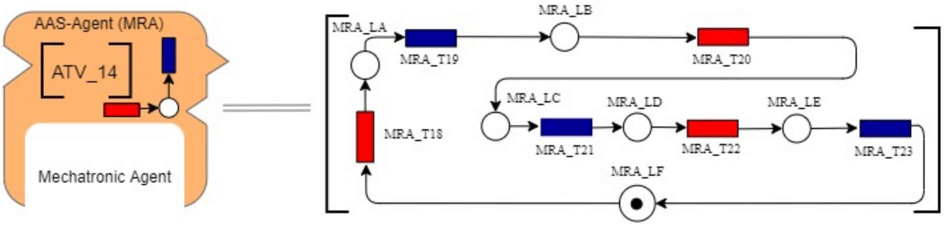

Figure 8. PN for activity [14- Resource Capabilities Search].

The place (MRA-LB) represents the end to process completely in component (RAS) of [ATV15]. In addition, the (MRA-LB) enables the transition (MRA-T20), which simplifies the activity [receives Capabilities]. Therefore (MRA-T20) is the return transition (RAS) with the capabilities requested by (MRA). The next place (MRA-LC) enables the transition (MRA-T21) that represents the activity [Informs Capabilities], that is, the agent (MRA) informs (CLA) the capabilities (Skill). The place (MRA-LD) indicates the availability of (MRA) to join in a coalition of (MCC), this allows the process flow, activating (MRA-T22) being this transition the simplification of the activity [Receive Request signature (MCC)]. The place (MRA-LE) is filled and enables the transition (MRA-T23), which indicates the response of (MRA) accepting participation in the contract (MCC). This procedure simplifies the activity [Informs Contract (MCC) signed] (Figure 8).

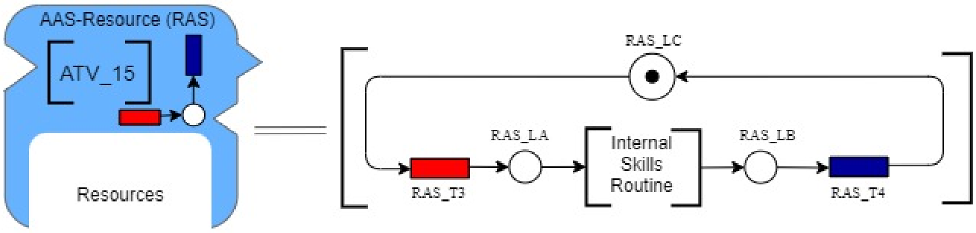

Figure 9. PN for Activity [15 - External Capabilities (Skill)].

The procedural model in (Figure 9) represents the PFS simplification by the activity [15 - External Capabilities (Skill)]. The process starts with the transition (RAS-T3) being responsible for receiving the [Skills Request]. In the next step, the place (RAS-LA), represents the activity [Internal Skills Routine] in (RAS). The intermediary [ATV] intends the AAS internal events externalizing the data contained in sub-models as capacity invoking resources (Skill). The place (RAS-LB) fills after the conclusion of the activity [Internal Skills Routine] that enables the transition (RAS-T4). This transition represents the [External Capabilities]. Finally, the information is available to (MRA), which invokes the [ATV-15] and filling the place (RAS-LC), which enables a new procedural cycle. The PN model simulation in Figure 10 is possible by the activity [16 – Coalition Mechanism]. Initially, the agent (CLA) enables [ATV-16] with the transition (CLA-T10), which represents the [Skills Search] activity. The place (CLA-LA) indicates the condition for the start of [Capacity Search] in which activated the transition (CLA-T11), enabling the agent (MRA) activity [Request Search of Capabilities]. The place (CLA-LB) indicates the waiting processes that evolve after a call from (MRA). The action (CLA-T12) means the return of (MRA) with the capabilities (Skill) for the coalition, represented by [receive Capabilities (Skill)].

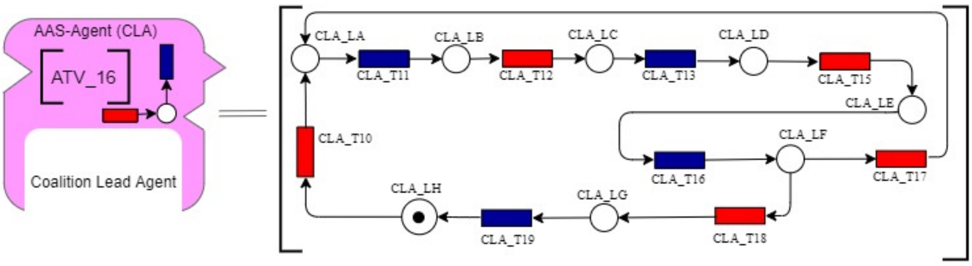

Figure 10. PN for Activity [16 - Coalition Mechanism].

The place (CLA-LC) represents the precondition to enable the activity processes [Contract Signature (MCC)]. The transition (CLA-T13) indicates the (CLA) request to (MRA) sign the contract (MCC) by the activity [requests Signature (MCC)]. The place (CLA-LD) means the (MRA) processes for analysis and signature of the contract and enables the transition (CLA-T15) that indicates the activity [return Signature (MCC)].

The process finishes when the capabilities (Skill) of resources is brought by (MRA), and the (MCC) is signed. The agent (CLA) calls the agent (YPA) through the activity [Complex Skill Search], so the place (CLA-LE) enables the transition (CLA-T16), which represents the [Request complex Skill] activity. The position (CLA-LF) indicates the wait for the action "Complex Skill" that enables two transitions; the (CLA-T17), indicating the inability to find (the complex Skill) [Does not receive complex Skill]. In this case, (CLA-LA) is filled, allowing the agent (CLA) to restart the new searches (MRAs) and, the transition (CLA-T18) is enabled in case of the possibility to find the (complex Skill), representing the activity [get complex Skill].

Finally, this activity fills (CLA-LG) indicates the completion of the capacity searches, with the agent (CLA) ready for [Execution of MCC Contracts]. The transition (CLA-T19) means the [ATV-17] conclusion in (YPA), and it changes the procedure status to the next level, represented by the activity [Execution of the Plan], last enables the (CLA-LH) realizing the [ATV-16].

The activity [17 – Complex Skill Mechanism] showed in Figure 11 is a simplified PN model for simulation. Initially, the transition (YPA-T25) indicates a new CLA request to execute the complex Skill machine [Receives Complex Skill Request].

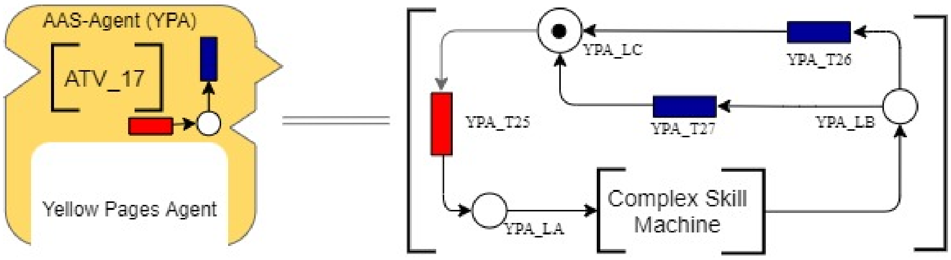

Figure 11. PN for Activity [17 – Complex Skill Mechanism].

The place (YPA-LA) enables the activity ["Complex Skill Machine"]. The place (YPA-LB) results in a decision through two transitions; a) (YPA-T26), returns to (CLA) a message of impossibility to find (Complex skill) [return “Complex skill” no found] and, b) (YPA-T27), returns to (CLA) with the (Complex Skill) founded [return “Complex Skill” found]. The [ATV-17] conclusion indicates filling the place (YPA-LC), making the activity available for a new search.

doi:10.20944/preprints202111.0083.v2

4.2. *Combination of process activity diagrams in PN to simulate the procedure*

Figure 12 presents the combination of activity diagrams treated in the functional models in Figure 8 to Figure 11. The procedure described enables PN simulation of the process [Coalition of Capabilities for (MCC)].

The main process flow is responsible for grouping the (capabilities - Skills) described in (RAS) that meet the requirements of secondary contracts. The agents achieve the activities collaboration: (i) (MRA) contributing to the activity [14- Search for Resource Capabilities]; (ii) (RAS) describes the standardization for virtual representation of resource capabilities [15 – External Capabilities (Skill)]; (iii) (CLA) is the leading agent of coalitions with the activity [16 - Coalition Mechanism] and; (iv) (YPA) assists the agent (CLA) in case of need to run the state machine to search for a (complex skill), executing the activity [17 – Complex Skill Mechanism].

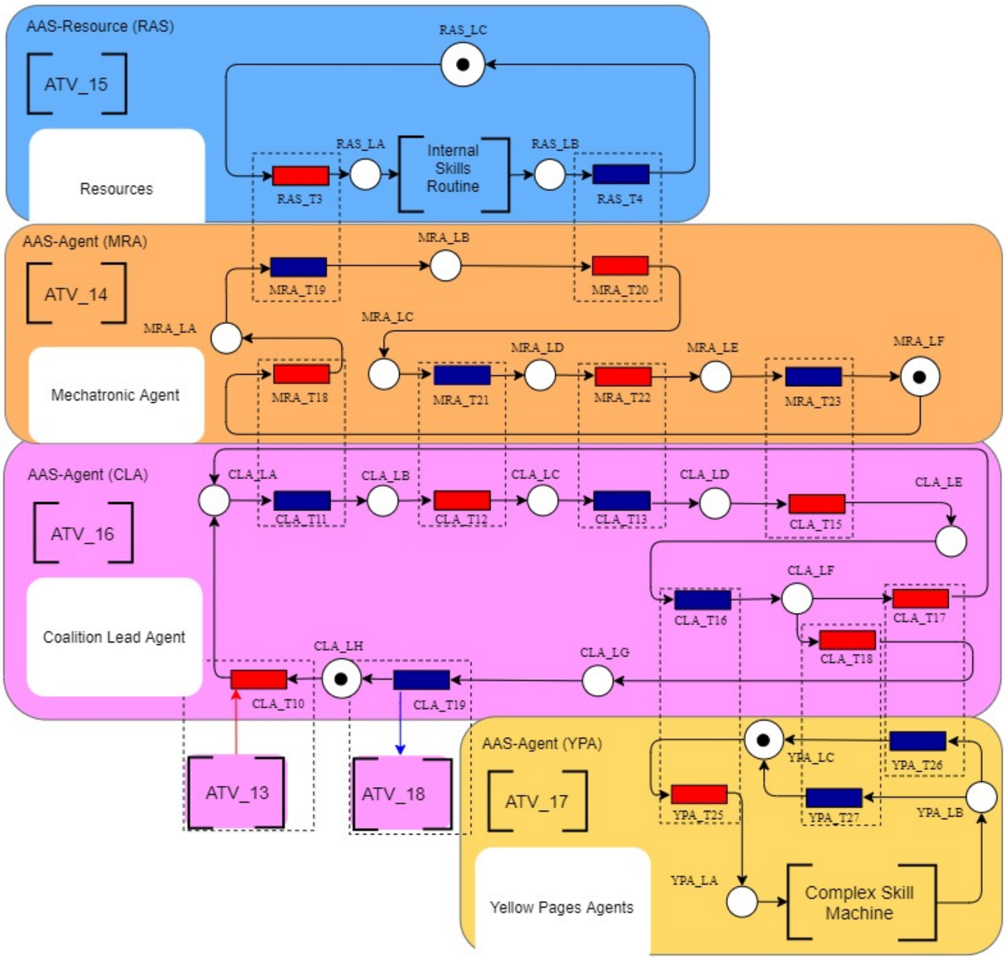

Figure 12. PN of the process [Coalition of Capabilities for (MCC)].

The collaboration process starts with interactions between the components represented by transitions as follows (Figure 12):

- CLA-T10 cooperate with [ATV-13]. This grouping of transitions starts after completing the activity [Preparation of Terms (MCC)], being responsible for exchanging information to "Search Skills";
- CLA-T11 and MRA-T18, this group of activities, enables the exchange of information between the agent (CLA) and (MRA) to search for new capabilities. These transitions represent the activities [Skills Search Request] and [Skills Search Request capabilities];

- MRA-T19 and RAS-T3, these transitions indicate a message of (MRA) with the activity [Request Capabilities in RAS] by the agent (RAS) that receives the [Skills Request];
- RAS-T4 and MRA-T20, this transition group indicates the return of (RAS) with [External Capabilities] activity to (MRA) that [receives Capabilities];
- MRA-T21 and CLA-T12, these grouping of transitions indicates the answer message from (MRA) with capabilities that execute the activity [Informs Capabilities] to the agent (CLA) that receives it through the [receives Capabilities (Skill)];
- CLA-T13 and MRA-T22, the following (CLA) internal routine is the activity [Signature of Contracts (MCC)]. The grouping of these transitions enables (CLA) to send the message [requests Signature (MCC)] to the agent (MRA) that process this information with the activity [Receive Signature Request (MCC)];
- MRA-T23 and CLA-T15, after completing the activities to adhere to the (MCC) contract. The agent (MRA) sends [Inform Contract (MCC) signed] to the agent (CLA) that enables the activity [Subscription return (MCC)];
- CLA-T16 and YPA-T25, the grouping of these activities allows the exchange of messages between agent (CLA) that [Request complex Skill] to an agent (YPA) that [Receive Complex Skill Request];
- YPA-T26 and CLA-T17, after the agent (YPA) execute the [Complex Machine Skill], it enables the activity [return "Complex Skill" not found] for (CLA) that receives the message of [Does not receive complex Skill] in case of impossibility to find the Skill needed for a coalition;
- YPA-T27 and CLA-T18, in case of possibility, to conclude the [Complex Skill Machine], the agent (YPA) returns to (CLA) with the activity [return "Complex Skill" found] enabling (CLA) to receive the missing capabilities for coalition via message [Receive Complex Skill];
- CLA-T19 – This transition is merged with a component in [ATV-18] that flags the [Plan Execution] for monitoring the (CLA) leader on the (MAS) plan to be accomplished;

4.3. Simulation result using Snoopy to validate the effectiveness of the procedure

The Snoopy simulation tools enable the representation and models validation; this technique supported the development of conceptual models. It was possible to use consecutive refinement techniques and simplification, improving both modeler and programmer activities. Therefore, the process described eliminates the Interlocks and guarantees the stability of the process sequence according to the reasoning provided in the previous steps (Figure 13).

The chosen systematic implements modeling techniques in a split proposal in stages prove to be efficient for the early stages of developing MAS-based control software or application. This procedure contributes to the specification of component activities and reduces efforts to implement the models. The MAS designers use OO techniques; however, the system's complexity is unavoidable.

It is understood that this proposal is an academic scenario of low complexity; however, in real use cases, the number of AAS submodels can increase significantly. So the methods assigned with PFS in a top-down approach helps modelers reduce the complexity of systems by generating simplified models in PN for validation of components and relationship between AAS objects and agents.

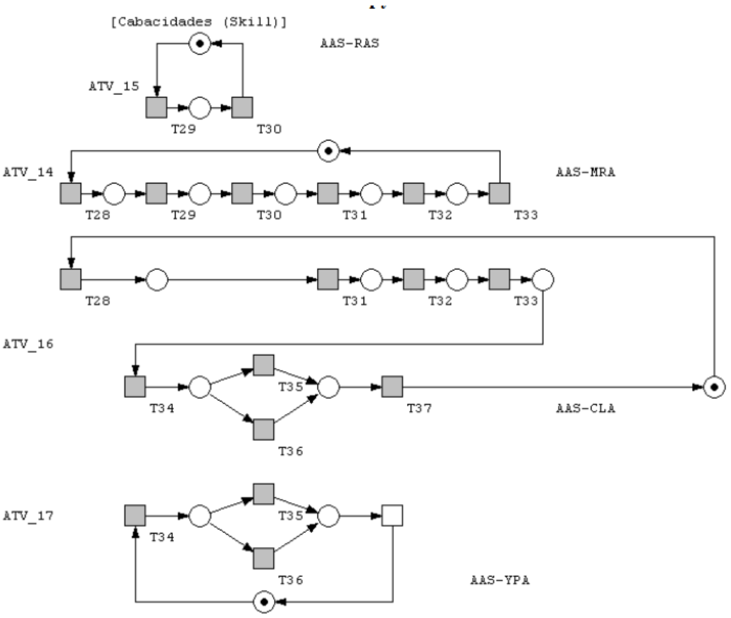

Figure 13. PN simulation for [Coalition of Capabilities for (MCC)] with Snoopy.

5. Conclusions

This work presents a method for building MAS models in the I4.0 context, which describes a system for the coalition of capabilities in AAS. The supported procedure defines the modeling method in which the problem and scope delimitation. The architectural components were coupled in RAMI 4.0 architectures then specified the activities of the elements to which they describe activities in the information layer. The features were represented the features by class diagrams (UML).

The preliminary steps consist of conceptual modeling techniques to simplify the systematized procedure through specific [Activities]. The Petri Nets (PN) support the functional modeling process, enabling the models in generated PFS to be simulated and validated. This system allows to carry out adjustments in the developed models. An applied example demonstrates the step [Coalitions of Capabilities for (MCC)] modeling techniques. In this sense, a model generates the PN diagrams of behavior.

The final step of the procedure describes each [Coalitions of Capabilities for (MCC)] activities. The details of elements and adopted techniques ensure liveliness and avoid interlocking in the behavior of the system. The consolidation of each PN proposes to integrate activities in the final simulations process. This procedure allows the integration of activities to carry out the described "Skill" capabilities coalitions (RAS). The relationship of activities is possible based on the collaboration of the agents. These relationships represent techniques used for the transitions coupling in the model. Finally, the system is simulated and validated with the tool snoopy.

Author Contributions: Conceptualization, J.V., M.P. and D.F.; methodology, J.V.; software, J.V.; validation, J.V., M.P. and D.F.; formal analysis, J.V.; investigation, J.V.; resources, J.V.; data curation, J.V.; writing-original draft preparation, J.V., M.P., F.J., P.M. and D.F; visualization, J.V.; supervision, M.P and D.F.; project administration, M.P.

Funding: This research was supported by the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP, São Paulo Research Foundation) grant numbers 2016/18797-0 and 2020/09850-0; and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq. National Council for Scientific and Technological Development) grant numbers 303210/2017-6 and 431170/2018-5.

Conflicts of Interest: The authors declare no conflict of interest.

Acknowledgments: The authors would like to thank the Brazilian governmental agencies CAPES, CNPq, PET, and FAPESP for their partial support.

References

1. Borangiu, T.; Trentesaux, D.; Thomas, A.; Leitão, P.; Barata, J. Digital transformation of manufacturing through cloud services and resource virtualization, 2019.
2. Bayha, A.; Bock, J.; Boss, B.; Diedrich, C.; Malakuti, S. Describing capabilities of industrie 4.0 components. *German Electrical and Electronics Manufacturers Association, Frankfurt am Main, Germany* **2020**.
3. Wagner, C.; Grothoff, J.; Epple, U.; Drath, R.; Malakuti, S.; Grüner, S.; Hoffmeister, M.; Zimermann, P. The role of the Industry 4.0 asset administration shell and the digital twin during the life cycle of a plant. 2017 22nd IEEE international conference on emerging technologies and factory automation (ETFA). IEEE, 2017, pp. 1–8.
4. Kagermann, H.; Anderl, R.; Gausemeier, J.; Schuh, G.; Wahlster, W. Industrie 4.0 im globalen Kontext. *Acatech Studie, https://www. acatech. de/wpcontent/uploads/2018/03/acatech_de_STUDIE_Industrie40_global_Web. pdf (14.09. 2018)* **2016**.
5. Bedenbender, H.; Bentkus, A.; Epple, U.; Hadlich, T.; Hankel, M.; Heidel, R.; Woolschlaeger, M. Relationships between I4. 0 Components–Composite components and smart production, 2017.
6. Bader, S.; Barnstedt, E.; Bedenbender, H.; Billman, M.; Boss, B.; Braunmandl, A. Details of the Asset Administration Shell Part 1 - The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC01), 2020.
7. Veiga, J.T.; Pessoa, M.A.; Junqueira, F.; Miyagi, P.E.; Santos Filho, D.J.d. Control of Manufacturing Systems by HMS/EPS Paradigms Orchestrating I4. 0 Components Based on Capabilities. Doctoral Conference on Computing, Electrical and Industrial Systems. Springer, 2021, pp. 62–70.
8. Veiga, J.T.; Pessoa, M.A.; Junqueira, F.; Miyagi, P.E.; Dos Santos Filho, D.J. Intelligent Manufacturing Systems: Self-organization in the I4. 0 context. 2021 14th IEEE International Conference on Industry Applications (INDUSCON). IEEE, 2021, pp. 153–160.
9. Onori, M.; Lohse, N.; Barata, J.; Hanisch, C. The IDEAS project: plug & produce at shop-floor level. *Assembly automation* **2012**.
10. Peixoto, J.A.; Oliveira, José Antonio Barata Rocha, A.D.; Pereira, C.E. The migration from conventional manufacturing systems for multi-agent paradigm: The first step. Doctoral Conference on Computing, Electrical and Industrial Systems. Springer, 2015, pp. 111–118.
11. Barata, J.; Camarinha-Matos, L.M. Coalitions of manufacturing components for shop floor agility-the CoBASA architecture. *International journal of networking and virtual organisations* **2003**, *2*, 50–77.
12. Frei, R.; Barata, J.; Serugendo, G. A complexity theory approach to evolvable production systems. Proceedings of 3rd International Workshop on Multi-Agent Robotic Systems in conjunction with ICINCO, 2007, pp. 44–53.
13. Dias, J.; Vallhagen, J.; Barbosa, J.; Leitão, P. Agent-based reconfiguration in a micro-flow production cell. 2017 IEEE 15th International Conference on Industrial Informatics (INDIN). IEEE, 2017, pp. 1123–1128.
14. Veiga, J.T.; Pessoa, M.A.; Junqueira, F.; Miyagi, P.E.; dos Santos Filho, D.J. Controle de Sistemas de Manufatura Distribuídos no contexto da Indústria 4.0. Congresso Brasileiro de Automática-CBA, 2020, Vol. 2.
15. Chakravorti, N.; Rahman, M.M.; Sidoumou, M.R.; Weinert, N.; Gosewehr, F.; Wermann, J. Validation of PERFoRM reference architecture demonstrating an application of data mining for predicting machine failure. *Procedia CIRP* **2018**, *72*, 1339–1344.
16. Trunzer, E.; Calà, A.; Leitão, P.; Gepp, M.; Kinghorst, J.; Lüder, A.; Schauerte, H.; Reifferscheid, M.; Vogel-Heuser, B. System architectures for Industrie 4.0 applications. *Production Engineering* **2019**, *13*, 247–257.
17. Mindas, M.; Bednar, S. Mass customization in the context of industry 4.0: implications of variety-induced complexity. *Advanced Industrial Engineering* **2016**, pp. 21–38.
18. Pisching, M.A.; Pessoa, M.A.; Junqueira, F.; Miyagi, P.E. PFS/PN technique to model Industry 4.0 systems based on RAMI 4.0. 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2018, Vol. 1, pp. 1153–1156.
19. Pisching, M.A.; Pessoa, M.A.; Junqueira, F.; dos Santos Filho, D.J.; Miyagi, P.E. An architecture based on RAMI 4.0 to discover equipment to process operations required by products. *Computers & Industrial Engineering* **2018**, *125*, 574–591.
20. Miyagi, P.E.; Hasegawa, K.; Takahashi, K. A programming language for discrete event production systems based on production flow schema and mark flow graph. *Transactions of the Society of Instrument and Control Engineers* **1988**, *24*, 183–190.
21. Heiner, M.; Herajy, M.; Liu, F.; Rohr, C.; Schwarick, M. Snoopy–a unifying Petri net tool. International Conference on Application and Theory of Petri Nets and Concurrency. Springer, 2012, pp. 398–407.
22. Gaedicke, L.; Guedes, G.; da Silva, J.P. MAS-ML Uma Linguagem para Modelagem de Sistemas Multi-Agentes: Uma Análise do Estado da Arte por Meio de uma Revisão Sistemática. Anais da II Escola Regional de Engenharia de Software. SBC, 2018, pp. 41–48.
23. Pessoa, M.A.; Pisching, M.A.; Yao, L.; Junqueira, F.; Miyagi, P.E.; Benatallah, B. Industry 4.0, how to integrate legacy devices: a cloud IoT approach. IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society. IEEE, 2018, pp. 2902–2907.
24. Cavalieri, S.; Salafia, M.G. Asset Administration Shell for PLC Representation Based on IEC 61131–3. *IEEE Access* **2020**, *8*, 142606–142621.
25. Frankó, A.; Vida, G.; Varga, P. Reliable identification schemes for asset and production tracking in industry 4.0. *Sensors* **2020**, *20*, 3709.
26. Xu, R.; Chen, Y.; Blasch, E.; Chen, G. Blendcac: A smart contract enabled decentralized capability-based access control mechanism for the iot. *Computers* **2018**, *7*, 39.
27. Barata, J.; Camarinha-Matos, L.; Onori, M. A multiagent based control approach for evolvable assembly systems. INDIN'05. 2005 3rd IEEE International Conference on Industrial Informatics, 2005. IEEE, 2005, pp. 478–483.

28. Ribeiro, L.; Barata, J.; Onori, M.; Hanisch, C.; Hoos, J.; Rosa, R. Self-organization in automation-the IDEAS pre-demonstrator. IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society. IEEE, 2011, pp. 2752–2757.