Collaborative Cyber-Physical Systems Design Approach: Smart Home Use Case

Artem A. Nazarenko(✉) and Luis M. Camarinha-Matos

School of Science and Technology and UNINOVA-CTS, Nova University of Lisbon,
2829-516 Monte Caparica, Lisbon, Portugal
a.nazarenko@campus.fct.unl.pt, cam@uninova.pt

Abstract. The growing trend on moving from isolated services to dynamically integrated/composed ones in a context where the cyber and physical worlds are interlinked, led to emergence of the concept of Collaborative CPSs (CCPSs). These systems rely on collaboration among internal and external components. An important aspect, in this regard, is the establishment of a design methodology for those systems. To satisfy agility requirements, the design process should be accomplished in a modular way, so that the system can be updated by adding or replacing modules. In traditional ICT systems the design process can be split into two parts/phases: the computational model design, i.e., functionality modules, and the design of a shell or service layer, providing the auxiliary services to utilize the computational model, e.g., security, human-machine interface, etc. In the case of CCPS design, the process also must consider the collaborative aspects within the design workflow. In the proposed work, we provide a model and design pattern (framework and a set of steps) for building Collaborative CPSs. To illustrate the approach, a smart home use-case is used.

Keywords: Collaborative cyber-physical systems · Smart home · Design patterns · Collaborative services

1 Introduction

The concept of systems composed of integrated physical and digital elements is referred as Cyber Physical Systems (CPS) [1]. Along with Internet-of-Things (IoT), CPSs are considered as one of the pillars of Industry 4.0, being applied in many other areas of human activities, such as: smart home, smart city, healthcare, smart farming, etc. The concepts of IoT and CPS are often interchanged. Both concepts include cyber and physical parts, however, IoT is more focused on connectivity to Internet, while CPS is more concerned with integration [2]. In both cases, the underlying idea of the concept is that a physical layer composed of physical devices, such as sensors, actuators, or more complex machines, is integrated with software or virtual components. This integration adds intelligence to the physical components. The idea of virtual and physical space integration is also the basis of the notion of Digital Twins (DT), used to establish virtual

Published by Springer Nature Switzerland AG 2021
L. M. Camarinha-Matos et al. (Eds.): DoCEIS 2021, IFIP AICT 626, pp. 92–101, 2021.
https://doi.org/10.1007/978-3-030-78288-7_9

replicas of physical entities through the modelling of the behaviour/processes occurring in the physical layer [3] and connecting to the physical counterparts.

An important element of advanced CPS is the collaborative aspect, as added value can be gained through the collaboration of system's components with each other, and even collaboration between those components and humans enabling the necessary abstraction level. This led to emergence of the concept of Collaborative CPS (CCPS), defined as systems "jointly acting and sharing information, resources and responsibilities in order to achieve a common goal" [2]. A common goal is, for instance, a collaborative or complex service that is delivered to the human-users of the system or the other components. The collaborative parties can originate from within the system (intelligent sub-systems) or be external entities. These ideas reflect an ongoing trend of moving from isolated services to rather interlinked ones [4].

CCPSs have to consider the collaborative aspect from the very beginning or the design phase. It is important to provide such a structure that can include technical components, "things", as well as users who can potentially collaborate and establish new services for mutual benefit. Most of the works devoted to design issues of CPS usually address the low-level design challenges. Some examples can be the timing requirements of CPS [5], model-based performance analysis [6] or use-case related design [7]. In this work we are focused on the CCPS design process, contemplating mechanisms to facilitate collaboration, and guided by the following research question:

> *What could be a suitable set of models and organizational structures to support the design of increasingly complex and evolving CCPSs?*

Even though, the adopted design process is briefly addressed, we provide an example of such process applied to a Smart Home domain. Domestic environments have been identified as "typical application domain for CPS" [8]. A smart home contains different sensors, actuators and smart appliances controlling the physical environment, as well as humans interacting/collaborating with them. Thus, a smart home can be considered as suitable case for validating the proposed design process.

2 Contribution to Applied AI Systems

Due to growing intelligence interconnection, autonomy and collaboration readiness of devices and sub-systems, modern CPS can be described as Cyber Physical Systems of Systems with "unprecedented capabilities and opportunities" [9]. Moreover, the constituents of those systems have a heterogeneous nature and vary from cyber and physical artefacts to human members [3], including the integration of different levels of complex sub-systems. Thus, modern CPS, where collaboration is one of the corner stones, can be seen as collaborative ecosystems of smart components/sub-systems. However, even a system containing smart and intelligent components cannot be fully recognised as smart, as only collaborative mechanisms forcing components to interact and establish more complex units makes the systems really smart. Thus, conceiving a proper organisational structure, as well as the methodology to design Collaborative CPS is needed to contribute to further smartification of these complex systems.

One possible way for systems' smartification is the creation of collaborative services, in analogy with Business Services [10], where several parties come together to satisfy user's or customer's needs. In the case of CCPS, these parties can be the smart sub-systems or smart components within the systems that can collaborate to generate added value. The formation of such networks or consortia/coalitions is an interesting topic derived from the Collaborative Networks domain [11]. As defined in previous work [12], a coalition of Smart Components is formed from the pool of virtual agents/digital twins associated to Smart Environments. After the coalition is established, it can provide collaborative services targeting the users' needs, and considering conditions of a particular environment.

3 Research Approach

The research approach is based on the CCPS design framework proposed in [3], inspired in the Design Science Research method (Fig. 1). The core pillars of this framework are: (i) Application Domain, possessing information about the use-case and its requirements, (ii) CCPS Design, used to design and develop solutions, and (iii) Knowledge Base, that corresponds to a repository of models, taxonomies, and design rules. The idea is to focus on the design of the CCPS ecosystem, in which members, as smart entities, are able to build temporary alliances in order to provide collaborative services. One of the key advantages of the adopted framework is that it allows storing the knowledge generated during the iterative design process within the Knowledge Base. Thus, generated knowledge can be re-used during the next design iterations. Moreover, the system being designed is considered as a set of modules – modular approach, which after the design phase can evolve and be further updated during the operational phase.

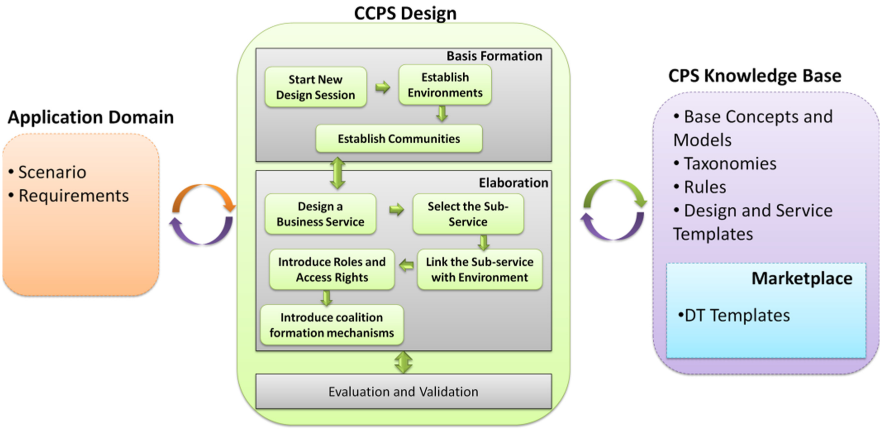

Fig. 1. Adopted CCPS design framework

After identifying/setting the requirements for the planned CCPS and acquiring the information about the domain, the CCPS design process starts. It includes some core steps: (i) new design session or establishment of the ecosystem, (ii) establishment of Smart Environments (SEs), (iii) establishment of Smart Communities (SCs), (iv) business service design, (v) sub-service selection, (vi) linking the sub-services with appropriate SE, (vii) access policy formulation and (viii) coalition formation mechanisms. The overall process is supported by the Knowledge Base. There is also the possibility, besides the models, taxonomies, and rules, to connect to an external Marketplace delivering DT building blocks.

One of the key models, part of the contribution from the Knowledge Base, is the high-level model (meta-model) of the complex CCPS. This model has three layers: (i) Ecosystem Layer, (ii) Organisational Layer, and (iii) Entities layer. The Ecosystem Layer gives the high-level view, while integrating both technical and social aspects of the system. The Organisational Layer relates to the social or human-related and technical entities, which are grouped into communities or Smart Communities and environments or Smart Environments, respectively. Thus, in general, the CCPS ecosystem can be considered as a set of Smart Environments and a set of Smart Communities. Moreover, the Organizational Layer is used to group/form/organize digital entities or digital representations of the physical objects (e.g., devices or human users) as digital twins. The model considers the existence of two types of DTs, the Asset and Human DTs that are used to reflect the real-world entities, such as devices/systems, and humans respectively. The Smart Environments are used to represent the logical partition of the Ecosystem that can match the physical partition, e.g., rooms in a home. In the case of the smart home, each environment stands for a certain room, such as bedroom, kitchen, etc. A typical SE consists of Asset DTs which are deployed in the environment at each moment. Please note that some smart devices/objects are mobile and can move among different SEs.

The Smart Community (SC) is an important element used as organizational entity for the human members of the ecosystem. Along with the SE, it belongs to the Organizational Layer. The human members of a SC are represented through Human DTs used to enable collaboration within the digital layer.

Human DTs are considered along with Asset DTs as a part of the Entities Layer. A typical Asset DT represents either a smart object, i.e., sensor or actuator, or a subsystem that might encapsulate several smart objects. The human DT, on one hand, is part of the system, providing some data about the owner, but on the other hand fulfilling social and administrative tasks [3]. The Entities layer can be considered as a bridge between virtual and physical entities. The DTs in the context of this work are used as data aggregators providing the necessary abstraction from the field level details. For instance, sensor sending the values every second, however, not every value triggers the change in condition/state of the DT. Thus, the Entities Layer containing Asset DTs and Human DTs aggregates data acquired from the field level, only focusing on the state/condition of the physical entity within the virtual layer. The high-level abstraction allows focusing on collaborative aspects, at the same time not ignoring the real-world events/processes. In this regard, behavioural changes are considered at design time through introduction of a set of rules or meta-rules determining the behavioural patterns of DTs. During the operational phase, the case-specific rules can be applied under the assumption that they are not violating the meta-rules, enabling customization and dynamic adjustment of services provided.

Further contribution is detailed elaboration of the design process steps introducing the order and logic of actions required to establish a CCPS. As the design is modular based, missing modules can be added during the next design sessions. The next section presents some domain-specific taxonomies that are utilized during the design process and used to populate the KB.

4 Smart Home Scenario

In this work, as mentioned above, the considered use-case is a Smart Home scenario. The models that are acquired from the KB are loaded into the design space in the form of taxonomies with interrelated concepts, attributes, and corresponding relations. At the end, the designed system can be represented as a complex graph composed of elements extracted from different models stored within the KB. However, the designer can enrich the KB, while updating the currently available taxonomies/models, so that the updates are available for other sessions. After the designed system will be in the operational stage, semantic relations can change due to dynamic nature of real-world systems [13], e.g. device changing the location. In this section we address the steps of the design process identified in the previous section (Fig. 2). The logic of every step, for the sake of explicitness, is represented in the form of simplified pseudo-code that can be converted into the Prolog and Python code.

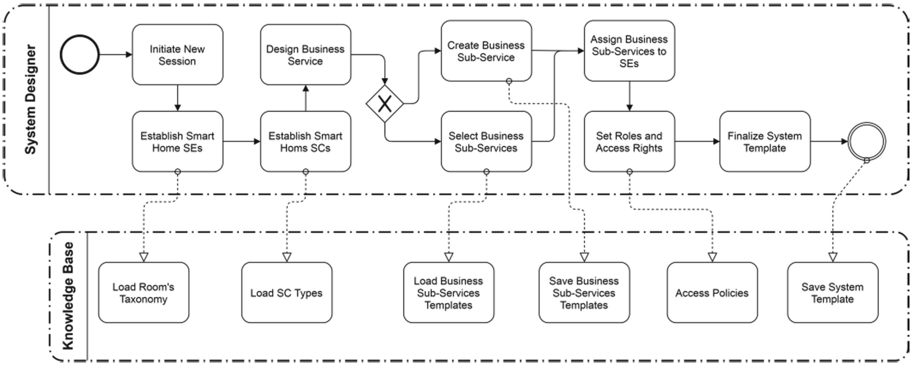

Fig. 2. CCPS design process

During the first step a new ecosystem entity is created, that belongs to a specific domain, in the example case a Smart Home. When the design process is launched, several options for the ecosystem type are offered to the designer acquired from the Knowledge Base. If no suitable options are available, the designer can update the currently available options and thus enrich the KB:

```
INPUT: ecosystem creation request, KB
for each ecosystem request do
    show available ecosystem types in KB
    if type required is available then
          set ecosystem type
    else
          enter new ecosystem type and save to KB
    end if
          return ecosystem instance of type
end for each
```

The second phase involves the creation of SEs that are immediately associated with the corresponding Ecosystem established during the previous step. Each SE is, as mentioned in the section above, an abstraction used for services/devices grouping. In the case of a Smart Home, SEs are associated with specific rooms/location types that can be imported from the KB.

```
INPUT: Ecosystem instance, KB
if ecosystem type is in KB then
    show available SE types from KB
    if SE type is available then
          set SE and assign to Ecosystem instance
    end if
else
    update KB
end if
```

Figure 3 shows a room's/space's taxonomy stored in the KB, where every room type like a "kitchen" is assigned to a more generic type, such as "utility spaces". Moreover, the taxonomy can be updated or extended, while importing new taxonomy and accomplishing appropriate mapping [14] or by integrating other atomic concepts into the common taxonomy.

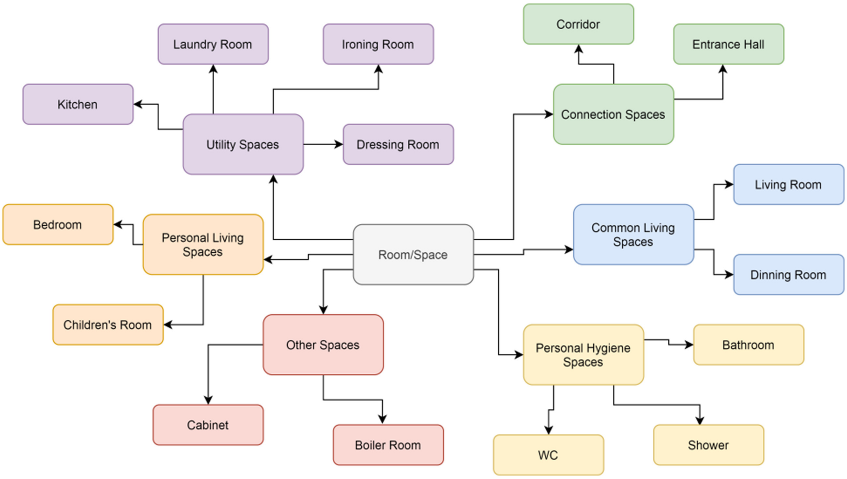

Fig. 3. Space/Room Taxonomy for the Smart Home

The third step involves the creation of SCs that are needed to reflect/represent different user's groups existing within the ecosystem. There can be also different types of communities that can be acquired from the KB. Some examples are: "family members", "visitors" and "service personnel". The type of community directly affects the priority and access rights to the resources. Similar to SEs, the SCs are assigned to a particular instance of the Ecosystem:

```
INPUT: Ecosystem instance, KB
if community type is in KB then
    show available SC types from KB
    if SC type is available then
          set SC and assign to Ecosystem instance
    end if
else
    update KB
end if
```

After SEs and SCs are established, the business services can be formed. Some service templates are also stored within the KB. The business services are of generic types, such as "comfort", "entertainment" or "care" services. Based on these generic types, sub-services can be formed that will directly target certain users' requirements and needs. The services might be complex, i.e., composed of some capillary/smaller ones. Important is to mention, the high-level entities that are designed hide the implementation information, such as protocols used to transmit payload, etc. Thus, we consider that sub-services are offered by the Asset DTs and Human DTs virtual entities:

```
INPUT: environment, Asset DT, Human DT, KB
select business service type
for each business service type
Create sub-services
    for each sub service
    Assign Asset DT or Human DT
    return sub service
return business service
end for each
```

After a service is established, it can be assigned to a specific SE or room, where it is available. Then a specific location, as for instance "kitchen", where the service is deployed will be assigned. If some service is not static, but can move to different locations, like the service offered by a smart vacuum cleaner, the location value will be changed, based on the taxonomy of locations specified for the considered ecosystem.

```
INPUT: set of smart environments, set of sub-service
for each sub-service do
    select smart environment and assign sub-service to smart
    environment
end for each
```

The access policy is intended to regulate the access of users to various resources/services. For the human-users the access policy is specified inside the corresponding communities; if the changes are applied, those will be immediately relevant for all the community members. The SCs have a set of roles, being intrinsic to the SC type that can be assigned to the community members defining the access to the services of the smart home. The access policy depends both on belonging to a certain community, as well as on having a certain role within this SC. All the entities getting access to a specific resource are restricted by the defined policy, unique for every community type. If the access policy residing in a different service is different from the SC one, first the access policy of the system (SC), in terms of priorities for the service access, will be applied and afterwards the access policy of external service. A possible solution for conflict resolution, if the access policy of a service is different from the access policy of a community for the same or compatible role, is to use the approach similar to logical conjunction. In other words, if both policies presume granting different access rights for the same role, e.g., the one defined by SC allows reading and writing, but the service one only allows reading, only reading option should be granted. Moreover, members of different SC can have different access rights, as well as members within the same SC also can have different access rights based on roles assigned.

The final step is the establishment or design of use-case specific reasoning rules. These rules should be generic enough, whereas setting the framework in which the ecosystem components can co-exist. To some extent this aim is similar to the invariant-based approach, relying on identification of basic situations that can appear during the algorithm execution [15] including the pre- and post-conditions. One simple example could be the case, when a fire alarm is raised (pre-condition), then all devices have to be switched off (post-condition). Moreover, additional reasoning rules are needed to back up the coalitions' formation and subsequently collaborative services.

The reasoning rules serve as the basis for coalition's formation (see Fig. 4) that is a complex process of combining various Asset and Human DTs in order to provide integrated collaborative services. The notion of the "coalition of smart components" corresponds to "*a temporary association of a set of cyber entities – digital twins, representing physical components, which can be dynamically configured and adjusted to a changing surroundings/demand in order to provide an integrated solution*" [12].

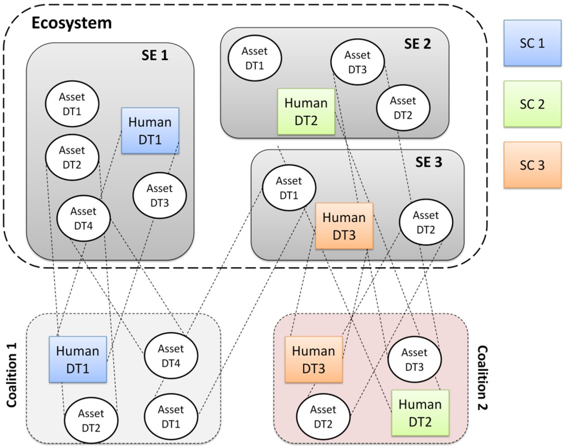

Fig. 4. Ecosystem view and coalition formation

The process of coalition formation consists of a set of stages, as for instance, partners' (i.e., DTs) suggestion [15]. An example of coalition's formation workflow presented in [12] contains the following stages: (i) ecosystem establishment, (ii) discovery/selection of components to be included into the coalition, (iii) negotiation, (iv) coalition launching. The full life cycle of a coalition also includes the operation stage (when the service is delivered) and the dissolution or evolution stages. These stages make a good match with the process of virtual organization creation in the context of a virtual organization breeding environment [10].

Figure 4 shows a simplified view of the coalition's formation process. The ecosystem and related components are already established in pool of virtual agents/digital twins. From this pool a set of ready-to-collaborate Asset and Human DTs are chosen to form the temporary alliances in order to generate collaborative services. After the service is no longer needed, we can consider a final stage of coalition's dissolution or evolution. The work on coalition formation mechanisms is still ongoing.

5 Conclusions

The main goal of this work is to have a design framework and a set of design steps/stages for complex Collaborative CPS. Advanced CCPS are viewed as systems composed of heterogeneous digital/virtual entities that tend to collaborate with each other, and thus able to generate added value. Two types of digital/virtual entities are considered: Asset digital twin and the Human digital twin. Foremost, the CCPSs are considered as intelligent systems taking into account the intelligence and autonomy of their components, as well as the collaborative focus of the systems. In this regard, the concept of coalitions of smart components formation is included in the work. The proposed framework is applied to a Smart Home use-case. Following a Design Science method, a knowledge base supports the design process through the provision of a CCPS meta-model and some domain taxonomies, as for instance the Space/room taxonomy of the smart home. A direction for the ongoing and further work is the development and broadening of the reasoning rules, setting the collaborative mechanisms, required to assist/support coalition's formation.

Acknowledgments. This work was supported in part by the Portuguese FCT foundation through the program UIDB/00066/2020 and European Commission (project DiGiFoF (Project Nr. 601089-EPP-1-2018-1-RO-EPPKA2-KA).

References

1. Xu, H., Yu, W., Griffith, D., Golmie, N.: A survey on industrial Internet of Things: a cyber-physical systems perspective. IEEE Access **6**, 78238–78259 (2018). https://doi.org/10.1109/ACCESS.2018.2884906
2. Nazarenko, A.A., Camarinha-Matos, L.M.: Towards collaborative cyber-physical systems. In: 2017 International Young Engineers Forum (YEF-ECE), Almada, pp. 12–17 (2017). https://doi.org/10.1109/YEF-ECE.2017.7935633

3. Nazarenko, A.A., Camarinha-Matos, L.M.: The role of digital twins in collaborative cyber-physical systems. In: Camarinha-Matos, L.M., Farhadi, N., Lopes, F., Pereira, H. (eds.) DoCEIS 2020. IAICT, vol. 577, pp. 191–205. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45124-0_18
4. Camarinha-Matos, L.M., Rosas, J., Oliveira, A.I., Ferrada, F.: A collaborative services ecosystem for ambient assisted living. In: Camarinha-Matos, L.M., Xu, L., Afsarmanesh, H. (eds.) PRO-VE 2012. IAICT, vol. 380, pp. 117–127. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32775-9_12
5. García-Valls, M., Perez-Palacin, D., Mirandola, R.: Pragmatic cyber physical systems design based on parametric models. J. Syst. Softw. **144**, 559–572 (2018). https://doi.org/10.1016/j.jss.2018.06.044
6. Pagliari, L., Mirandola, R., Trubiani, C.: Engineering cyber-physical systems through performance-based modelling and analysis: a case study experience report. J. Softw. Evol. Process (2019). https://doi.org/10.1002/smr.2179
7. Bhuiyan, M.Z.A., Wu, J., Wang, G., Cao, J., Jiang, W., Atiquzzaman, M.: Towards cyber-physical systems design for structural health monitoring. ACM Trans. Cyber-Phys. Syst. **1**(4), 1–26 (2017). https://doi.org/10.1145/3086508
8. Seiger, R., Huber, S., Schlegel, T.: Toward an execution system for self-healing workflows in cyber-physical systems. Softw. Syst. Model. **17**, 551–572 (2018). https://doi.org/10.1007/s10270-016-0551-z
9. Törngren, M., Grogan, P.T.: How to deal with the complexity of future cyber-physical systems? Designs **2**, 40 (2018). https://doi.org/10.3390/designs2040040
10. Oliveira, A.I., Camarinha-Matos, L.M.: Negotiation environment and protocols for collaborative service design. In: Camarinha-Matos, L.M., Baldissera, T.A., Di Orio, G., Marques, F. (eds.) DoCEIS 2015. IAICT, vol. 450, pp. 31–41. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16766-4_4
11. Camarinha-Matos, L.M., Afsarmanesh, H.: Collaborative networks: a new scientific discipline. J. Intell. Manuf. **16**, 439–452 (2005). https://doi.org/10.1007/s10845-005-1656-3
12. Nazarenko, A.A., Camarinha-Matos, L.M.: Basis for an approach to design collaborative cyber-physical systems. In: Camarinha-Matos, L.M., Almeida, R., Oliveira, J. (eds.) DoCEIS 2019. IAICT, vol. 553, pp. 193–205. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17771-3_16
13. Wang, X., Dong, J.S., Chin, C.Y., Hettiarachchi, S., Zhang, D.: Semantic space: an infrastructure for smart spaces. IEEE Pervasive Comput. **03**(03), 32–39 (2004). https://doi.org/10.1109/mprv.2004.1321026
14. Nazarenko, A.A., Sarraipa, J., Camarinha-Matos, L.M., Garcia, O., Jardim-Goncalves, R.: Semantic data management for a virtual factory collaborative environment. Appl. Sci. **2019**(9), 4936 (2019). https://doi.org/10.3390/app9224936
15. Back, R.-J., Preoteasa, V.: Semantics and proof rules of invariant based programs. In: Proceedings of the 2011 ACM Symposium on Applied Computing - SAC 2011 (2011). https://doi.org/10.1145/1982185.1982532
16. Rosas, J., Camarinha-Matos, L.M.: An approach to assess collaboration readiness. Int. J. Prod. Res. **47**(17), 4711–4735 (2009). https://doi.org/10.1080/00207540902847298