DOI: 10.48011/asba.v2i1.1241

Towards a Formal Design to Service-Oriented Cloud Manufacturing

Jose Reinaldo Silva * **Elinilson Vital** **

* *Design Lab - Universidade de São Paulo, São Paulo, SP 05508-030*
Brazil (e-mail: reinaldo@usp.br).
** *Design Lab - Universidade de São Paulo, São Paulo, SP 05508-030*
Brazil (e-mail: vital@usp.br).

Abstract: Since the beginning of this century, there is a paradigm shift concerning production, pushed by digital convergence, and the need to change the world's energy balance drastically. New production systems tend to become distributed, cloud-based, service-oriented, and anthropocentric, alternating full automation and collaborative human participation. This scenario demands a revision of design methods applied to manufacture systems and integration between new and classic methods. This article brings a proposal to integrate cloud-based and service-oriented systems by a framework capable of supporting design methods suitable to compose a distributed set of manufacturing services (CMfgS).

Keywords: service design; cloud manufacturing; model based systems engineering; product-service systems.

---

1. INTRODUCTION

The concept of Manufacturing-as-a-Service (MAAS) was first introduced thirty years ago by Goldhar and Jelinek (1990). However, its full realization in cloud manufacturing (CMfg) appeared only twenty years later, when cloud service provision became available. Nevertheless, CMfg presupposes a manufacturing process being transformed into a network of collaborative services available in the cloud and associated with a product and respective supply chain. In this context, MAAS (manufacturing-as-a-service) proposal follow current demands pushed by digital convergence and by the need to converge towards a better equilibrium in the use of energy by manufacturing facilities (Nof and Silva, 2018), while still attached to the conventional concept of a product, and consequently, to its detached relationship with consumers. The impacts caused by the introduction of cloud manufacturing have been studied since the beginning of this decade. That includes a direct impact on the supply chain (Jessbi et al., 2014) - a key issue to the success and adoption of CMfg.

Service Engineering also experienced fast development, especially in the last decade. The sharp dichotomy between product and service has been replaced by the new concept of product-service (Morelli, 2006; Tan et al., 2007; Cavalieri and Pezzotta, 2012), which takes conventional products as a resource to provide services and couple it with the user, resulting in substantial value co-creation(Galvagno and Dalli, 2014). Therefore, it is impossible to separate cloud manufacturing from a full service-oriented process, whose primary goal is to provide product-services.

Consequently, new design models for service-oriented cloud manufacturing have attracted more attention from academy and practitioners (Lu and Xu, 2019; Li et al., 2018; Yu et al., 2015; He and Xu, 2015). In this paper, we briefly revisit classic design proposals for cloud manufacturing (Li et al., 2009) and propose a new integrated approach to service-oriented cloud manufacturing systems, directed to produce product-service artifacts. The proposal is based on goal-oriented requirements engineering modeling, which could be dynamically shared and used as a base to collaboration and reuse of designs (Li et al., 2018; Moghaddam and Nof, 2018).

2. CLOUD MANUFACTURING ARCHITECTURE

There are several papers in the literature describing cloud manufacturing (CMfg) issues. Several define CMfg as a manufacturing network of services in the cloud without coordination that configures them as a productive arrangement. Some works have attached hardware based on cyber-physical cells (Yu et al., 2015), inheriting all advantages of cloud systems: scalability, use-on-demand, pay-per-use, and others, besides the potential for reducing the size of manufacturing plants drastically, with the consequent energy saving.

Manufacturing service agents in those architectures fall into three categories:

- *consumers*, who receive the manufacturing results (product-services);
- *providers*, who make manufacturing service facilities available;
- *orchestrators* or *operators* (Adamson et al., 2017), who are responsible for the matching between service demand and service providing [1].

---

[1] Although different authors use different terms, the meaning is the same, that is, a sub-system that finds a service that fits customer expectations, usually pointed by a stakeholder.

There are small differences in the role of the *operator*, but its goal is the matching between service demand and the manufacturing services (MSs) available in the cloud. Therefore, there is a lack in the definition of the role of *operators*. Actually, *Operators* can go beyond demand/service matching - even when formal approach (Moghaddam and Nof, 2017) is applied. That becomes a challenge in the design of collaborative cloud manufacturing services. Another challenge is fitting this design into a supply chain network to provide horizontal integration.

Operators should be partially automated, which demands design methods that fit a process-oriented approach into a suitable architecture. A first attempt to produce a suitable architecture for the design of distributed manufacturing services (not in the cloud) was made by Dutra and Silva (2016), depicted in Figure 1.

Product-service architecture (PSA) in Figure 1 is based on a network of services, like many other works in the literature. However, it includes a design framework where *consumers* can enter with generic intentions that are converted to requirements, automatically or assisted by system *operators*. Another supporting service of this framework introduces modeling for the coupling or matching between service *consumers* and service *providers*, which were predicted but not developed in the referred article.

In the current proposal, manufacturing processes are synthesized from requirements using planning techniques based on AI and Petri Nets (Silva and Silva, 2019). We also propose revising the original PSA architecture, including cloud services and its orchestration, in the design process. It helps to model and design an arrangement of manufacturing services to fit a production goal.

2.1 Goal-oriented requirements for the orchestrator

The design process for cloud manufacturing services (CMfgS) proposed here starts with goal-oriented requirements (Horkoff et al., 2019). That fits better with a process-based approach that intends to model the coupling with a consumer. Instead of looking for specific functionalities of each manufacturing service (MS), a goal-oriented approach is associated with systemic goals.

Goal-oriented requirements engineering was developed by John Milopoulos and recently revised by him and several other authors (Horkoff et al., 2019). Its main advantage is replacing conventional requirements modeling, based on functionality, to a process based on objectives (goals). The GORE approach eliminates the dichotomy between functional and non-functional requirements and enhances traceability - a critical issue to the design of service arrangements - while keeping a formal approach to requirements model in LTL (Linear Temporal Logic).

Besides, it would be better to introduce a formalism based on state-transitions, such as Petri Nets (or Time Petri Nets, if real-time is necessary) to represent dynamic systems, particularly dynamic manufacturing processes. In Silva and Silva (2019) a software tool was presented to help the modeling of manufacturing services called RekPlan, which transfers a set of GORE diagrams into a classic or extended Petri Net. Figure 2 shows an example where RekPlan models a flexible integrated manufacturing site for an automobile company, synthesizing the Petri Net shown in Figure 3. Formal models in Petri Nets can be analyzed, simulated, and formally verified. That is an essential issue in the design of cloud manufacturing services.

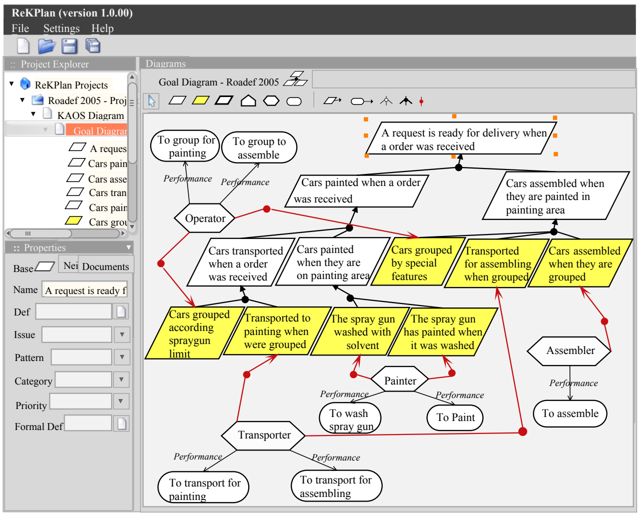
Figure 1. Product-service architecture. Source: Dutra and Silva (2016)

Thus, the design process for manufacturing services (MfgS) proposed by Silva and Silva (2019) can be extended to Cloud Manufacturing Service (CMfgS) and also enhanced by the introduction of specific ontologies directed to GORE (Debbech et al., 2019). Artificial Intelligence planning techniques can be used to automate (partially or totally) the CMfgS orchestrator.

3. THE CLOUD-BASED CHALLENGE

A great debate about the concept of cloud-based design and manufacturing (CBDM) has started at the beginning of this decade. CBDM was initially concerned with designing a network of manufacturing services dedicated to products (Adamson et al., 2017; Wu et al., 2015). A great effort was dedicated in this direction, producing methods, processes, and software environments (Xu, 2012). Some innovations appeared with emerging software environments to support product and manufacturing design (Quickparts, Modelica, and others). That was followed by other tools to support the design of information systems in the cloud (like Enterprise Architect), which evolved into the collaborative design, cloud-based collaborative design, and finally, cloud-based service design.

Design methods also evolved to model-based (systems) engineering, improving formalism (Wymore, 1993) and



Name A request is ready f

Figure 2. RekPlan System with a GORE model for an automobile site. Source: Silva and Silva (2019)

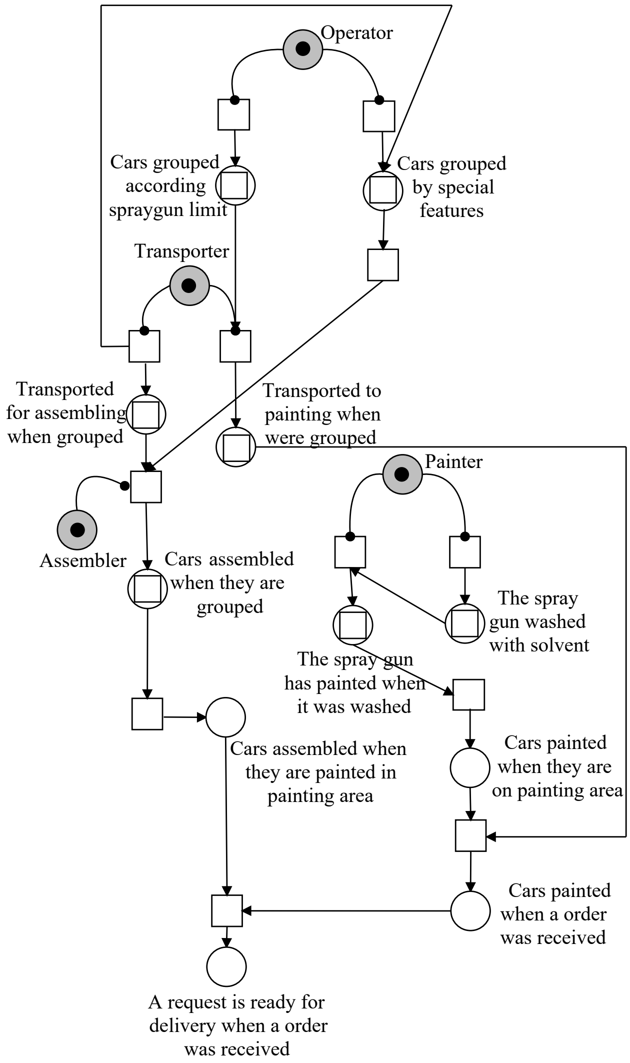


Figure 3. Petri Net model synthesized by RekPlan. Source: Silva and Silva (2019)

holism. Other targets included collaboration (Li et al., 2018; Moghaddam and Nof, 2018); attaching hardware and CPS (Cyber-physical systems)(Liu and Xu, 2017; Yu et al., 2015; Zhang et al., 2017), and finally, the revision of the whole process and methods to fit CMfgS, which motivated the current work.

Requirement methods for CMfgS should include special features, such as:

- requirements should be modeled, formalized and analyzed, preferentially in a goal-oriented approach;
- design methods should also be formalized and verified to include big data and data analysis (Lu and Xu, 2019);
- design methods should have traceable and dynamic requirements;
- AI planning methods should assist the designer to compose a process that models the production arrangement;
- production arrangements should be reusable;
- requirement methods should model and verify the coupling with customers (value co-creation).

Therefore, CMfgS design presents a significant challenge in what concerns design methods, especially in what concerns the requirements phase.

In the current proposal, requirements are the primary concern. They should be modeled and verified using goal-oriented methods, based on KAOS diagrams Horkoff et al. (2019), which can later be transferred to LTL (Linear Temporal Logic) or Petri Nets (Silva and Silva, 2019). Verification is performed using Petri Net property analysis or validated by simulation. The design of new CMfgSs consists of looking for arrangements of MfgSs (available in the cloud) to compose a production process, which can be formalized in Petri Nets isomorph to those in the requirements model. KAOS also includes responsibility diagrams, which help to make the whole process traceable. Specific AI planning and design methods can be inserted, as shown in the previous work of Silva and Silva (2019), where such methods fit the goal-oriented approach. Reusability is crucial and is introduced by searching high-level descriptions of models represented by ontologies, first-order logic, or Petri Nets. There is also an object-oriented proposal based on the concept of abstract views/abstract objects that can be revisited (Silva et al., 1995). Finally, a value co-creation model can be introduced using ontologies (OCL) and goal-oriented requirements (Debbech et al., 2019).

The above features compose a conceptual and practical framework implemented by integrating different tools (ReKPlan, to transfer KAOS diagrams to Petri Nets(Silva and Silva, 2019), itSIMPLE to deal with the knowledge engineering to AI Planning(Silva et al., 2020)(Vaquero et al., 2013), GHENeSys, to support Petri Nets property analysis(Silva and del Foyo, 2012)) [2] . The next section will show the basics of how to put together all these tools in our proposal.

---

[2] All those tools were previously developed in D-Lab

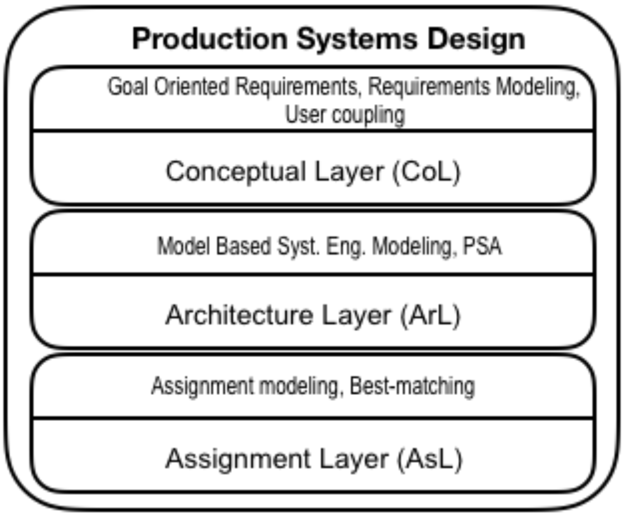


Figure 4. Layered schema for PSA design. Source: Nof and Silva (2018)

4. PROPOSITION OF A CLOUD-BASED FRAMEWORK TO THE DESIGN OF CMFGS

Nof and Silva (2018) proposed a design method for product-services with layers that generically fit the concepts detailed in the previous section. Figure 4 shows the general model.

The design process is synthesized in three layers: the conceptual layer, where intentions about the production arrangement (CMfgS) are transferred to formal requirements, and analyzed, followed by the architecture layer, where a general (planning) design for the arrangement that generates the product-service is designed. The third layer should find suitable arrangements for CMfgSs that fit the requirements. In a first approach, the assignment layer could make the matching - or an optimized matching (best matching) (Moghaddam and Nof, 2017), associating demand, and available services. This process can be enhanced to include intelligent support to manage reusable designs of similar production systems.

A general schema for the framework can be seen in Figure 5. The cloud arrangement at the top of Figure 5 depicts the basic framework. The CMfgS's customer, the agent, looking for a distributed production arrangement, can consult a set of repositories (in the cloud) looking for suitable components. An automated orchestrator (system designer) mediates this search. This orchestrator will be specified in the next subsection. Human-machine relations are explored to make the association between human and machine and provide an integrated collaborative process to support the system designer.

The design framework produces a CMfgS - where some components could have attached cyber-physical systems as resources. The generated production system is capable of manufacturing a product-service (PS), which is supposed to couple with final customers(as shown in the bottom of Figure 5). Therefore, there are two different couplings: between the design system and the CMfgS customer (which is the focus of the present work) and the coupling between the product-service and the final customer. This last coupling is included in the design process of the product-service and will not be treated here.

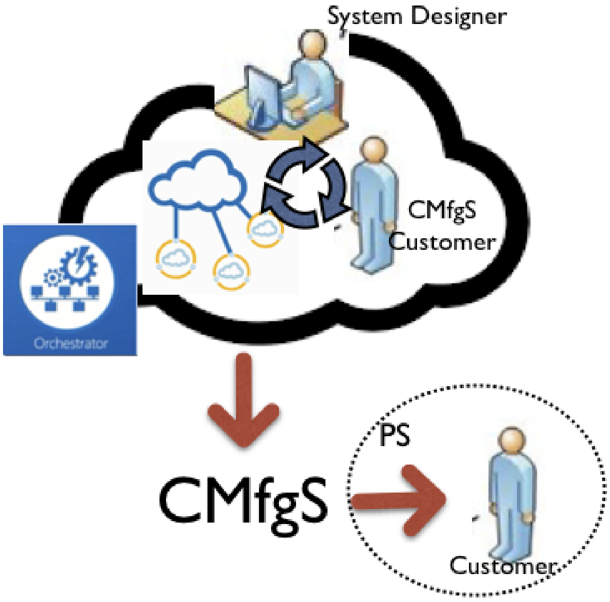


Figure 5. General schema for the proposed cloud design framework.

4.1 Multi-cloud repositories

The proposed framework's efficiency depends on the availability of a CMfgS repository, distributed in different cloud providers. Therefore, it is imperative to orchestrate access to multiple repositories, supported by different cloud providers without affecting the design process.

We define as a repository of CMfgS, a set of components, each composed by a description called *Abstract Model*, complemented by a goal-oriented ontology; a *Cloud service* application that makes it possible to operate CMfgS in collaboration using the internet, eventually with an attached CPS.

Abstract Models resume the main goals and operational facilities of CMfgSs and are the basis for reusability. That means that a search for a reusable component can be performed by AI components of the orchestrator using these general descriptions. Therefore, the condition to get the proper performance is that all general descriptions be a closed set of specifications (Silva et al., 1995). CMfgS model (Abstract Objects) concentrates operations and interfaces to access the cloud manufacturing service, personalized to suit the CMfgS customer.

Different cloud providers could store different repositories. Therefore proper management would require a multi-cloud implementation based on Kubernets (Luksa, 2017). Some cloud providers claim to have new cloud services to manage Kubernetes from different cloud providers. By the time this paper was written, some announcements were made about launching orchestration services capable of running over generic multi-cloud facilities. This paper is based on direct algorithms to manage Kubernets orchestrators using

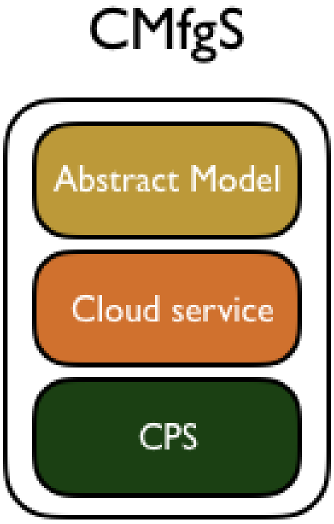


Figure 6. General schema for a CMfgS.

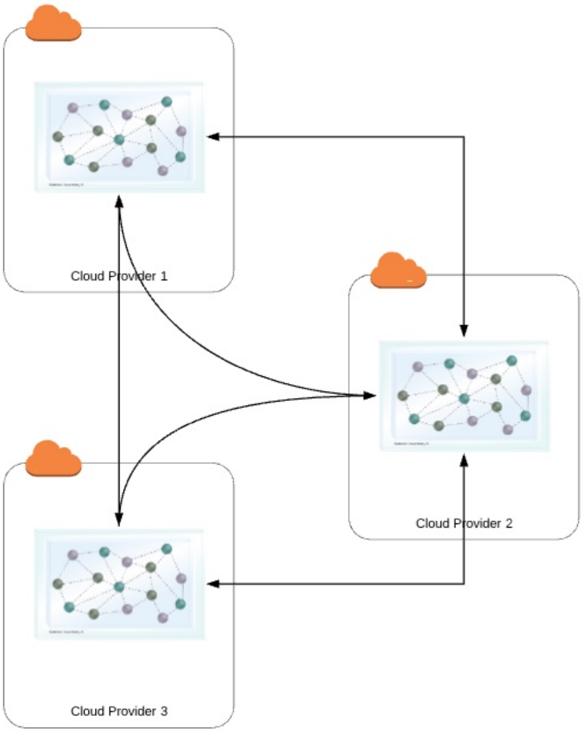


Figure 7. Cluster of repositories to support CMfgS design and reusability.

a unique provider and to simulate different CMfgS that could be in different containers, in different clouds.

A cluster of repositories was implemented in Kubernets, each capable of storing a set of different CMfgS components with the structure described in Figure 6. The orchestrator could access all repositories, as illustrated by 7. The search for a component is performed by webots looking for specific abstract CMfgS models that could stand for potentially reusable components. The confirmation of this component as a potential candidate to compose the target CMfgS should result from the match made by the orchestrator (automatically or in a mixed-mode, with collaborative participation of a human system designer).

Revisiting Figure 1, all service components *Service X* stand for reusable candidates that can be distributed in different cloud providers. The managing protocol is now replaced by a goal-oriented ontology called GOORE (Goal-oriented Ontology for Requirments Engineering) proposed by Shibaoka et al. (2007). We followed this approach, but some enhancements would have to be included to fit the service concept.

The workflow management system is represented by a (classic) Petri Net model. The service planner can be instantiated by a sequence of actions to call specific components and services, resulting in a process plan. The automatic orchestrator synthesizes both plan and workflow models in using PSA (product-service architecture). This duplicated structure stands by ”plant” and ”control” using the terminology associated with classic manufacturing design.

Intention production model is replaced by *Abstract Models*. GOORE requirements express value co-creation using additional representation in predicates (not implemented yet) and user profiles representing the CMfgS customer. CMfgS intentions and generic business processes inspire this profile.

Finally, the orchestrator can generate specific CMfgS customer user interfaces (abstract views), based on the coupling. We propose to generate concrete instances for these elements by interfaces derived from generic models and implemented in Ruby on Rail.

A version of the PSA design process specific to cloud manufacturing is being implemented using a unique cloud provider. However, cloud providers' current tendency to launch large storage systems - eventually spread in different clouds - makes this reduction irrelevant.

The whole developed process can be formalized and verified by theorem proving algorithms. Unfortunately, a practical validation depends on the construction of large repositories - which demands a significant effort - to support a consistent and convincing search for reusable components. Such a repository should be large enough to support data analysis and other bigdata methods (Lu and Xu, 2019).

A small version of a case study was adapted from a manufacturing application suggested by the automotive industry and can be found in Silva and Silva (2019). Different sectors of an automotive company (painting, body shopping, and assembler) were used as a ”repository” into a general operation supported by a PSA architecture. We will not reproduce this same example here because of space limitations. Besides, it is quite impossible to reproduce a cloud search in a paper, and that is why we decide not to include a case study in this paper.

5. THE CLOUD STRUCTURE FOR THE REPOSITORIES

A cloud-based architecture was designed to support repositories, directly programmed over a cloud Infrastructure-as-a-Service (IaaS).

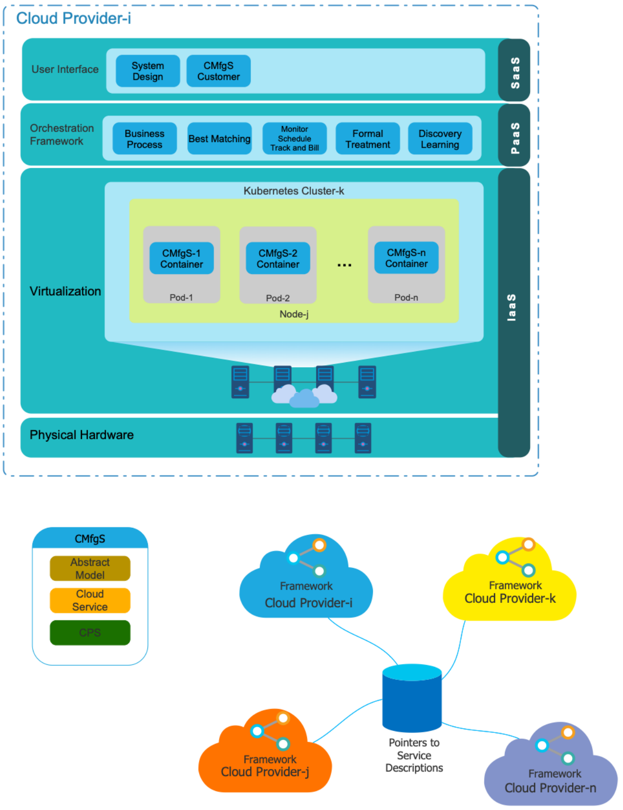


Figure 8. Cluster of repositories to support CMfgS design and reusability.

CMfgSs with the basic structure shown in Figure 6 are implemented in containers and arranged in Pods named CMfg-x, composing a single repository. Several repositories compose a Node, which could be characterized, for instance, by a category of service manufacturing systems. Different nodes could be available in different cloud providers.

The set of repository categories compose a Kubernet cluster and could be stored in just one or distinct cloud providers. The architecture implemented so far is based on a unique cloud provider that agreed to support the current research with its IaaS and some cloud services.

Cloud repositories are potentially supported by physical hardware, eventually CPSs. On the top of the Kubernet structure, a PaaS (Platform-as-a-Service) is coordinated by the orchestrator, which assists the CMfgS customer as an apprentice or act automatically to suggest reusable CMfgS components. The Abstract Model is used to direct and support searches while the orchestrator helps with the process planning and the operation of the target CMfgS that results from this process. Potential candidates to reuse are stored in a data structure controlled by the orchestrator and eventually discarded if not used in the new CMfgS design.

CMfgS customers intend to build a coupling to collaborate with the final user, even if they are not supposed to contact them. The unique way to achieve that is to anticipate the final user model and include that in the requirements of the CMfgS. That should also be done with any component that integrates to main CMfgS, making the process recursive.

The structure in Figure 8 correspond the top level cluster in Figure 5.

6. COMMENTS AND ANALYSIS

Cloud Manufacturing, and especially CMfgS, are still being adopted by the industry, and some clear definitions and contributions from the academy are still expected. Besides, the architecture and design models, control, and operational problems are also challenging (Morel et al., 2019). This paper did not address such problems but is concentrated on two design challenges: orchestrating a collaborative design process and managing the coupling with the CMfgS designer (or customer). Even this design process needs to receive further contributions.

In the current article, the authors explore a combination of formal treatment and practical experience, acquired in convincing design processes. The architecture presented here will be applied to a case study proposed by industry in Roadef (see `http://www.roadef.org`) [3]. The original problem requires the design and sequencing of different manufacturing services to an interleaving car manufacturing (Renault presented problem). Our intention is now to face an international demand to sequence the same target manufacturing in different (manufacturing) services available in the world. The challenge to achieve that is the organization of the repositories, as pointed before.

However, before investing in case studies, some basic processes should be formalized. Scaling applications in the cloud is also a challenging phase that should follow the theoretical proposition of the framework presented here. That affects the efficiency and performance of search algorithms and reusability processes of CMfgS components. Therefore, the size and complexity of Abstract Modes would have to be considered to fit performance in the search for reusable components.

The use of GORE increased a lot the design process and contributed to the process's performance. However, to be fully adopted, some other points should be added, such as the real connection between goal-oriented requirements and production processes documentation. Formal approaches could not hide this practical problem. On the contrary, they should bring some light to it.

Another problem not treated in this article is security: this practical problem is another critical issue to adopting the new paradigm. One of the motivations (besides performance and agility) to use Abstract Models is the need to hide detailed information about the production of unwanted users. That could compromise the possibility of having repositories of CMfgS available in the could, a pillar of the proposed architecture.

However, we do not even touch the point on how specific information could be omitted in Abstract Modes or CMfgS description. It is unclear whether the industry would deal with metamodel descriptions of CMfgS, even if there are applications connected with world projects where this approach is used in practice. Such consideration would

---

[3] ROADEF is an effort of the Association Française de Recherche et d'Aide à la Decision, which works in the connection between the industry and academy

raise the importance of interface relations (abstract views), as proposed in the original work of Silva et al. (1995).

Finally, a critical point that could make the new paradigm unfeasible in practice is the coupling with the supply chain. This coupling could be considered in two ways: in value co-creation, where the presence of products as resources are essential to delivering product-service; and providing and transferring resources to feed CPSs components. The first is more sensitive to the formal model and should be the object of further work.

On the other hand, consolidated works point to methods that support product-service resources supply Jessbi et al. (2014). Modern transportation systems already deal with the combination of resources and services (eventually available in the cloud) offering bicycles, cars, and other artifacts, in multi-modal transportation services. Rehabilitation services could be provided with instructions on the cloud (telemedicine) or the delivery of specialized equipment associated with home care applications. Thus, we do not expect the connection with the supply chain could be a real problem to consolidate the proposed CMfgS design approach.

There is an expectation that the experience accumulated by the practice in service applications could inspire similar manufacturing services solutions. The unfortunate situation presented by the pandemic suggests, for instance, that product (or product-service) artifacts to assist or protect patients in-home would be a manufacturing challenge for the near future.

7. CONCLUSION

A significant change toward a new production paradigm is still a perspective for Industry 4.0. (Liu and Xu, 2017). A comprehensive study about methods and tools associated with the migration from I3.0 to I4.0 and directly associated with Industry 4.0 shows that cloud manufacturing is a demanding issue in the next decade Nakayama et al. (2020). Adopting this new production paradigm is controversial and could be challenging, especially to the industry's conservative sectors - as the automotive. On the other hand, moving to new production arrangements, especially if decentralized, presupposes the improvement of design methods, to reduce risks and enhance CMfgS coupling with the designer (and also with the final user).

Academic perspectives for a new production paradigm that involves CMfgS (Vernadat et al., 2018) should also fit socio-economic impacts as presented by the European Union Committee(Sargsyan, 2011). Such an analysis should also benefit from a formal model and a cloud-based approach to the production arrangement.

The present work presented a cloud-based architecture and pointed to design perspectives beyond I 4.0, focusing on human-centered approaches identified with Society 5.0 and industry 5.0 (I 5.0) Aslam et al. (2020)Salgues (2018). The features described in this paper are based on PSA architecture's proposition, created by one of the authors (Silva and Nof, 2015)(Dutra and Silva, 2016), which evolved to receive - in this work - an Engineering Design treatment and inserted in a cloud framework described in this article. The primary purpose is to provide a complete framework that can formalize the design process of CMfgS, that is, distributed manufacturing systems spread all over the world. To achieve that, we used a requirements cycle based on a goal-oriented approach and treated manufacturing as a collection of services (not just processes), which could be instantiated by CPSs (cyber-physical systems). The framework is composed by tools and methods developed by the authors in D-Lab. We used previous developments of D-Lab, as RekPlan (to transfer KAOS diagrams to Petri nets and allow formal verification)(Silva and Silva, 2019); itSIMPLE (to plan sequences of actions in manufacturing using AI methods)(Vaquero et al., 2013); and GHENeSys (a Petri Net environment to do formal analysis)(Silva and del Foyo, 2012).

The architecture presented so far is based on Kubernets and coordinated by an orchestrator working in a multi-cloud environment. There are results concerning implementation still being explored, but the work is under development and evolving towards a validation with a convincing industrial case study.

Further work points to the inclusion of new Artificial Intelligence methods, primarily based on AI Planning, and the inclusion of ontologies to analyze CMfgS processes' requirements.

8. ACKNOWLEDGEMENTS

The authors thank Universidade de São Paulo and Amazon Web Systems (AWS) for the support of this research in Cloud Manufacturing Services, in the scope of USP-AWS collaboration.

REFERENCES

Adamson, G., Wang, L., Holm, M., and Moore, P. (2017). Cloud manufacturing–a critical review of recent development and future trends. *International Journal of Computer Integrated Manufacturing*, 30(4-5), 347–380. doi:10.1080/0951192X.2015.1031704.

Aslam, F., W., A., Li, M., and Ur Rehman, K. (2020). Product-Service Architecture (PSA): toward a Service Engineering perspective in Industry 4.0. *Information*, 11(2), 91–96.

Cavalieri, S. and Pezzotta, G. (2012). Product–Service Systems Engineering: State of the art and research challenges. *Computers in Industry*, 63, 278–288.

Debbech, S., Bon, P., and Collart-Detilleul, S. (2019). Towards Semantic Interpretation of Goal-Oriented Safety Decisions Based on Foundational Ontology. *Journal of Computers*, 14(4), 257–267.

Dutra, D. and Silva, J. (2016). Product-Service Architecture (PSA): toward a Service Engineering perspective in Industry 4.0. *IFAC-PapersOnLine*, 49, 91–96.

Galvagno, M. and Dalli, D. (2014). Theory of Value Co-creation. A Systematic Literature Review. *Journal of Service Theory and Practice*, 24(6), 643–683.

Goldhar, J.D. and Jelinek, M. (1990). Manufacturing as a Service Business: CIM in the 21st Century. *Computers in Industry*, 14, 225–245.

He, W. and Xu, L. (2015). A state-of-the-art survey of cloud manufacturing. *International Journal of Computer Integrated Manufacturing*, 28(3), 239–250. doi: 10.1080/0951192X.2013.874595.

Horkoff, J., Aydemir, F., Cardoso, E., Li, T., Maté, A., Paja, E., Slnitri, M., Piras, L., Mylopoulos, J., and Giorgini, P. (2019). Goal-oriented requirements engineering: an extended systematic mapping study. *Requirements Engineering*, 24(2), 133–160.

Jessbi, J., Orio, G., and Barata, J. (2014). The Impact of Cloud Manufacturing on Supply Chain Agility. In *Proc. of 12th. IEEE Int. Conf. on Industial Informatics.*

Li, B., Zhang, L., Wang, S.L., Tao, F., Cao, J., Jiang, X.D., Chang, X., and Chai, X.D. (2009). Cloud manufacturing: A new service-oriented networked manufacturing model. *Computer Integrated Manufacturing Sytems*, 16(1).

Li, P., Cheng, Y., and Tao, F. (2018). A manufacturing services collaboration framework toward industrial internet platforms. *2018 International Conference on Intelligence and Safety for Robotics, ISR 2018*, 166–171. doi:10.1109/IISR.2018.8535991.

Liu, Y. and Xu, X. (2017). Industry 4.0 and cloud manufacturing: A comparative analysis. *Journal of Manufacturing Science and Engineering*, 139(3), 1–8. doi:10.1115/1.4034667.

Lu, Y. and Xu, X. (2019). Cloud-based manufacturing equipment and big data analytics to enable on-demand manufacturing services. *Robotics and Computer-Integrated Manufacturing*, 57(November 2018), 92–102. doi:10.1016/j.rcim.2018.11.006.

Luksa, M. (2017). *Kubernets in Action.* Manning Publications, 1st edition.

Moghaddam, M. and Nof, S.Y. (2017). *Best Matching Theory & Applications.* Springer, 1st edition.

Moghaddam, M. and Nof, S.Y. (2018). Collaborative service-component integration in cloud manufacturing. *International Journal of Production Research*, 56(1-2), 677–691. doi:10.1080/00207543.2017.1374574.

Morel, G., Pereira, C.E., and Nof, S.Y. (2019). Historical survey and emerging challenges of manufacturing automation modeling and control: A systems architecting perspective. *Annual Reviews in Control*, 47, 21–34. doi:10.1016/j.arcontrol.2019.01.002.

Morelli, N. (2006). Developing new product service systems (PSS): methodologies and operational tools. *Journal of Cleaner Production*, 14(17), 1495–1501.

Nakayama, R., Spinola, M., and Silva, J. (2020). Towards I4.0: A compreehensive analysis of evolution from I3.0. *Computers & Industrial Engineering*, 144.

Nof, S.Y. and Silva, J.R. (2018). Perspectives on Manufacturing Automation Under the Digital and Cyber Convergence. *Polytechnica*, 1(1-2), 36–47. doi:10.1007/s41050-018-0006-0.

Salgues, B. (2018). *Society 5.0: Industry of the Furure, Technologies, Methods and Toolls.* Wilhey, 1st edition.

Sargsyan, G.e. (2011). OSI: Socio-economic impacts of Service Innovation. *European Union Report*, 178. doi:10.2759/48997.

Shibaoka, M., Kaia, H., and Saeki, M. (2007). GOORE : Goal-Oriented and Ontology Driven Requirements Elicitation Method. In J.L.H. et al. (ed.), *Lecture Notes in Computer Science, 4802*, 225–234. Springer.

Silva, J.M. and Silva, J.R. (2019). A New Requirements Engineering Approach for Manufacturing Based on Petri Nets. *IFAC-PapersOnLine*, 52, 97–102.

Silva, J.R. and del Foyo, P. (2012). *Timed Petri Nets*, chapter 16, 359–372. Petri Nets: Manufacturing and Computer Science. Intech.

Silva, J.R., Silva, J.M., and Vaquero, T. (2020). Formal knowledge engineering for planning: Pre and post-design analysis. In M. Valatti and D. Kitchin (eds.), *Knowledge Engineering Tools and Techniques for AI Planning*, chapter 3. Springer.

Silva, J., Afsarmanesh, H., Cowan, D., and Lucena, C. (1995). An Object-oriented Approach to the Design of Flexible Manufacturing Systems. In L. Camarinha-Matos and H. Afsarmanesh (eds.), *Balanced Automated Systems: Architectures and Design Methods*, chapter 9, 91–106. Springer.

Silva, J. and Nof, S. (2015). Manufacturing as a Service: From e-Work and Service-Oriented Approach Towards a Product-Service Architecture. *IFAC-PapersOnLine*, 48, 1628–1633.

Tan, A., McAloone, T., and Gall, C. (2007). Product/Service-Systems Development - An explorative case study in a Manufacturing company. In *Proc. of Int. Conf. on Engineering Design, ICED'07.*

Vaquero, T., Silva, J.R., Tonidandel, F., and Beck, J. (2013). itSIMPLE: towards an integrated design system for real planning applications. *The Knowledge Engineering Review*, 28(2), 215–230.

Vernadat, F.B., Chan, F.T., Molina, A., Nof, S.Y., and Panetto, H. (2018). Information systems and knowledge management in industrial engineering: Recent advances and new perspectives. *International Journal of Production Research*, 56(8), 2707–2713. doi:10.1080/00207543.2018.1454615.

Wu, D., Rosen, S., Wang, L., and Schaefer, D. (2015). Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation. *Computer Aided Design*, 59, 1–14.

Wymore, A. (1993). *Model-Based Systems Engineering.* Taylor & Francis, Boca Raton, 1st. ed. edition.

Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and Computer-Integrated Manufacturing*, 28(1), 75–86. doi:10.1016/j.rcim.2011.07.002.

Yu, C., Xu, X., and Lu, Y. (2015). Computer-Integrated Manufacturing, Cyber-Physical Systems and Cloud Manufacturing - Concepts and relationships. *Manufacturing Letters*, 6, 5–9. doi:10.1016/j.mfglet.2015.11.005.

Zhang, Y., Zhang, G., Liu, Y., and Di, H. (2017). Research on services encapsulation and virtualization access model of machine for cloud manufacturing. *Journal of Intelligent Manufacturing*, 28, 1109–1123. doi:10.1007/s10845-015-1064-2.