International Journal of Computer Integrated Manufacturing

ISSN: 0951-192X (Print) 1362-3052 (Online) Journal homepage: http://www.tandfonline.com/loi/tcim20

A Cloud-Based Architecture with embedded Pragmatics Renderer for Ubiquitous and Cloud Manufacturing

Luís Ferreira, Goran Putnik, Maria Manuela Cruz- Cunha, Zlata Putnik, Hélio Castro, Catia Alves, Vaibhav Shah & Leonilde Varela

To cite this article: Luís Ferreira, Goran Putnik, Maria Manuela Cruz- Cunha, Zlata Putnik, Hélio Castro, Catia Alves, Vaibhav Shah & Leonilde Varela (2017) A Cloud-Based Architecture with embedded Pragmatics Renderer for Ubiquitous and Cloud Manufacturing, International Journal of Computer Integrated Manufacturing, 30:4-5, 483-500

To link to this article: https://doi.org/10.1080/0951192X.2017.1291995

Published online: 01 Mar 2017.

International Journal of Computer Integrated Manufacturing, 2017
Vol. 30, Nos. 4–5, 483–500, http://dx.doi.org/10.1080/0951192X.2017.1291995

A Cloud-Based Architecture with embedded Pragmatics Renderer for Ubiquitous and Cloud Manufacturing

Luís Ferreira[a], Goran Putnik[b,c]*, Maria Manuela Cruz- Cunha[a,c], Zlata Putnik[c], Hélio Castro[d], Catia Alves[c], Vaibhav Shah[c] and Leonilde Varela[b,c]

[a]*School of Technology, Polytechnic Institute of Cávado e Ave, Barcelos, Portugal;* [b]*School of Engineering, Campus de Azurém, University of Minho, Guimarães, Portugal;* [c]*ALGORITMI Research Centre, University of Minho, Campus de Azurém, Guimarães, Portugal;* [d]*ParallelPlanes, Lda., Braga, Portugal*

(*Received 2 February 2017; accepted 2 February 2017*)

The paper presents a Cloud-based architecture for Ubiquitous and Cloud Manufacturing as a multilayer communicational architecture designated as the Communicational Architecture. It is characterised as (a) rich client interfaces (Rich Internet Application) with sufficient interaction to allow user agility and competence, (b) multimodal, for multiple client device classes support and (c) communicational to allow pragmatics, where human-to-human real interaction is completely supported. The main innovative part of this architecture is sustained by a semiotic framework organised on three main logical levels: (a) device level, which allows the user 'to use' pragmatics with the system, (b) application level which results for a set of tools which allows users pragmatics-based interaction and (c) application server level that implements the Pragmatics renderer, a pragmatics supporting engine that supports all pragmatics services. The Pragmatics renderer works as a communication enabler, and consists of a set of integrated collaboration technology that makes the bridge between the user/devices and the 'system'. A federated or community cloud is developed using a particular cloud RESTful Application Programming Interface that supports (cloud) services registration, composition and governance (pragmatics services behaves as SaaS in the cloud).

Keywords: manufacturing systems; integration; cloud and ubiquitous manufacturing; pragmatics

1. Introduction

In Cloud Manufacturing (CM), all stakeholders (suppliers, customers, providers, producers, etc.) must benefit from the venture and must receive additional value on top of the value received in traditional manufacturing relationships (Wu et al. 2013). To handle the required effectiveness of this business models, mechanisms to support direct collaboration and cooperation are needed. Having a new, dynamic and global business model inherent to cloud infrastructure, we see the change of traditional production processes to adapt to this global and virtual chain of resources and stakeholders. Another relevant concept is Ubiquitous Manufacturing (UM). There are several models developed, see for example, (Cha and Yoo 2008), (Kim et al. 2012), (Kim and Song 2008), (Onosato et al. 2001). While the CM is related more to the properties of virtually 'infinite' resources and 'elastic' scalability, the UM is related to the properties of availability of management, control and operation functions of manufacturing systems and enterprises anywhere, anytime, using direct control, notebooks or handheld devices, that is, to the devices that provide ubiquity of the functions (these devices are not relevant as these are only hardware platforms for the services, which (the hardware) interoperability is the issue of the operating systems which is not considered within the proposed architecture).

Combining properties of both approaches, that is, of CM and UM, could be defined as the Ubiquitous and Cloud Manufacturing (UCM) concept.

This UCM business model brings the quick reaction to market changes, and the high availability and capacity to effectively support changed requirements, as one of the main sustainability criterion (Ferreira 2013; Putnik 2010a).

In the 'classical' literature, the both concepts of CM and UB, and of their derivation UCM, are primarily related to use of the Cloud Computing (CC) and Ubiquitous Computing (UC) systems (respectively). Many of the existent infrastructures are already cloud based or changing towards that virtual architecture.

However, there are two quite different approaches to both the concepts of CM and UM, that is, of UCM, for which it is possible to say that these two approaches in fact represent two different paradigms of CM and UM, and of UCM:

- The first approach, considers CM and UM, that is, UCM, as the manufacturing systems (MS) based on extensive use of cloud and ubiquitous computational systems (CCS and UCS).

*Corresponding author. Email: putnikgd@dps.uminho.pt

- The second approach, which is the authors' original approach, considers 'cloudiness' and ubiquity of the MS as a homomorphism, that is, as a mapping, of the CCS and UCS.

Concerning the second approach, that is, the second paradigm, it is inspired by the phenomenon of UM that has been known for years but not as an application of UC. Alfred Weber (cited in (Foust 1975)) has defined the UM:

> Ubiquity naturally does not mean that a commodity is present or producible at every mathematical point of the country or region. It means that the commodity is so extensively available within the region that, wherever a place of consumption is located, there are ... opportunities for producing it in the vicinity. Ubiquity is therefore not a mathematical, but a practical and approximate, term (praktischer Naherungsbegriff).

That is, 'the term "ubiquitous"' is 'explicitly defined to be functional in an empirical context ...The types of manufacturing which are both market oriented and have a **frequency of occurrence greater than** [bold by the authors] a specific limit which can be empirically defined are ubiquitous ...'. The importance of the referred work is in contributing to the understanding of the phenomena of ubiquity, and related scalability, and in that they may not need necessarily the UC technologies (Putnik et al. 2013).

The similar idea was referred in Murakami and Fujinuma (2000), (ref. in (Serrano and Fischer 2007)). This approach is referred as well as 'Ubiquitous networking' that 'emphasises the possibility of building networks of persons and objects for sending and receiving information of all kinds and thus providing the users with services anytime and at any place'.

Also, for instance, Putnik and Putnik (2010), Putnik (2010a) and Xu (2012) suggest a manufacturing version of ubiquitous and CC (respectively) – UCM – and manufacturing with direct adoption of ubiquitous and CC technologies and their structural and organisational properties, generating a manufacturing service-oriented network. This manufacturing service-oriented network can stimulate from production-oriented to service-oriented manufacturing (Cheng et al. 2010). To use efficiently those infrastructures, the applications must be transformed and follow services-oriented applications patterns. Thus, the intended UCM demands information systems that ensure sufficient ubiquity and availability of any stakeholder (Li and Mehnen 2013).

Following the second paradigm referee above, which implies that UCM has to have a large number of redundant resources, or entities, or components, to be able to provide ubiquity, that is, availability of resources 'anywhere, anytime', as well as virtually 'infinite' 'elastic' scalability. As the large number of redundant resources is not likely to be employed by a single enterprise, it implies a necessary resource to the networked organisation.

In other words, in this paper UMS is defined as a MS based on a 'hyper'-sized manufacturing network, consisting of thousands, hundreds of thousands or millions of 'nodes', that is, of manufacturing resource units (in principle of any type), freely accessible and independent, that represent a homomorphism of the CC and UC systems, and which may use extensively CC and UC systems as technological support and enablers.

However, Ubiquitous and Cloud Manufacturing Systems (UCMS) can only be possible if the efficient interoperability between resources (people, machines, time, services, etc.) is ensured. To assure the resources workflow (composition) it requires their efficient and **effective** integration and mechanisms to coordinate that process.

Having multiple service types (SaaS – applications, IaaS – infrastructure, PaaS – platform, HaaS – hardware) and deployment models (Private, Community, Public, Hybrid), the cloud technologically faces demanding interoperability requirements. *'(...) the resources that are available'* in the cloud *'must be accessible to users of a cloud-based service, which requires that such a service be interoperable with the devices, operating systems, and applications of its user base ... as more businesses, organizations, and individuals use the cloud for their operations, a more diverse range of cloud-based applications will need to interoperate smoothly and efficiently ... the optimal level of interoperability in cloud services may not develop spontaneously or naturally. (...)*' (Becker 2012).

Nevertheless, several facts show that the more independence from technology, the best architecture robustness and flexibility we get. Since technology does not stop evolving, everything which depends on it needs to change too.

UCMS supporting architectures must not be only a technological-based solution. Persons are increasingly essential resources, better actors and they must be considered active users in all system. The user participation is essential and his eventual co-decision is a must. Thus, for UCM support, a communicational architecture model that integrates mechanisms to allow pragmatics is required.

We state that a traditional transactional or existing cloud-based architectures, enriched with communicational services, strengthened with pragmatics support and cloud services reliability is more effective (and therefore more sustainable) since it refers real users and not 'abstracted users', as has happened with others pure technological architectures. This is *the underlying research hypothesis* of the UCM supporting information system architecture presented in this paper.

In summary, nowadays interoperability needs to frame technology (that covers syntactic and semantic integration) as well as pragmatics-based interoperability, which could essentially mean human-to-human communication capacity.

This paper addresses the development of a novel and original effective information system architecture, as the paradigmatically innovative architecture for effective interoperability of UCM to support UCM that integrates the pragmatics services through the implementation of the *Pragmatic renderer*, which represent the main theoretical contribution.

In accordance with the UCM definition above, used in this paper, as UCM is considered as a networked organisation which components, or elements, are freely accessible and independent, UCM is necessarily composed, or integrated, by homogeneous or heterogeneous components (apps and other type of services). The case of the UCM composed of the homogeneous components, whether proprietary or standard, could be considered as a reductionist's version of UCM and this is not of a stronger scientific interest. However, the case of the UCM composed of the heterogeneous components, which could be considered as a true UCM, is of the strong scientific interest as up today there are no effective solutions for such type of UCM. This is mainly because of the interoperability problems between the heterogeneous components. In fact, it could be stated that the true UCM is not possible without embedded instruments for enabling full and effective interoperability.

In this context, actually the concept of UCM itself implies the architecture that enables full and effective interoperability of heterogeneous components of an UCM. Otherwise, the concept which would be called UCM would be a 'weak', or 'pseudo-' UCM.

In these terms, the innovative component of the proposed architecture is embedment of, the authors called it, 'pragmatic renderer' as the instrument for enabling full and effective interoperability of heterogeneous components of an UCM. The 'pragmatic renderer' provides capability of employing pragmatics effects during the process of establishing the interoperability of applications within the cloud environment. The 'traditional' architectures for interoperability up to date are limited to the semantic levels (e.g. employing (international) standards for data formats, ontologies, and similar, such as STEP, STEP-NC (ISO 10303, 14649), MT-connect) which is proved in linguistic theory that are not capable to establish fully effective interoperability. It could be said, that while the 'traditional' architectures are 'transactional' (information transaction) in their nature, the architecture presented in this paper is 'communicational' or 'co-constructive' in its nature. Further, while the 'traditional' architecture is focused on the interoperability interfaces' ability to correctly transmit the information, the architecture presented in this paper is focused on the co-constructive (or co-designing, co-creating) process of establishing the interoperability mechanism. Using other words, that is, in another interpretation, it could be said that the proposed architecture is conceived to overcome the integration problems that arise from the technology-based solutions only. The main existing proposals for the UCM architectures explore technological platforms as solution, essentially. Since UCM demands information systems that ensure sufficient ubiquity and availability, and even known that connectivity is not yet ubiquitous, the proposed architecture enriches those technological perspective with social perspective where the user can interact easily with the system and naturally with other users in the process of co-creation. That is, the 'pragmatic renderer' adds real effectiveness to existing non-integrated proposals.

In this way, the presented architecture actually proposes a model for a *new generation of architectures* with the objective to provide true effective interoperability required in ubiquitous and cloud systems, and in our case, in UCM.

Although it is obvious that there is a 'human interface' embedded in the architecture, employment of the human interface is also paradigmatically different than in 'traditional' architectures. While in 'traditional' architecture human interfaces are employed within the concepts of 'Graphical User Interface' (GUI) and 'User Experience' (UX), in the presented architecture the role of human interface is on the higher level than in GUI and UX, as in the presented architecture the human interface has the communicational, that is, the co-creation function, which could be considered as a theoretical contribution too.

The phenomenological, and theoretical, basis for the pragmatics-based interoperability, that is, the concept of 'semiotic framework for manufacturing systems integration' which is considered by the authors of this paper as the fundamental theoretical basis for the new interoperability paradigm, that led to the conceptualisation of the 'pragmatic renderer', is presented in (Putnik and Putnik 2010).

While there could be found in the literature examples of the pragmatics-based interoperability of MS, that is, the concept of the 'semiotic framework for manufacturing systems integration', models and applications, see e.g. the Special Issue on 'Semiotic based manufacturing systems integration', in *International Journal on Computer Integrated Manufacturing*, Vol 23, No 8–9. (Putnik 2010b), there is not found in the literature any more detailed model of the UCM architecture with embedded pragmatics-based interoperability instruments. In this sense, this paper represents a contribution to the literature on UCM modelling.

The followings of the paper are divided into five sections. Section 2 presents the basic concepts underlying the pragmatics-based interoperability. Section 3 presents a review of typical information systems architectures for UCM, and Section 4 details the proposed architecture. In Section 5, a prototype implementation of the proposed architecture is presented and in Section 6 is given a qualitative comparison of the proposed architecture with the reviewed architectures from Section 3. Finally conclusions and future work are drawn in Section 7.

2. Pragmatics-based interoperability – the emergent semiotic framework requirements

The capacity that allows persons to interact with each other, in a particular context of space, time and any other criteria, belongs to Pragmatics, a field of semiotics (semantics and syntactic are other two fields of semiotics) that concerns the relation between 'signs' and their interpreters. The terms syntax, semantics and pragmatics were introduced in linguistic and semiotics theory (Saussure 1916; Morris 1938).

Any information system that aims to consider the true needs of a particular customer, that is, the needs closest to the real customer's needs, with as less as possible abstractions, should consider pragmatic dimension of communication with him. To clarify this, one must distinguish clearly the properties of syntactic, semantic and pragmatics aspects. '(…) if in an investigation explicit reference is made to the speaker, or, to put it in more general terms, to the user of language, then we assign it to the field of pragmatics… If we abstract from the use of the language and analyse only the expressions and their designate, we are in the field of semantics. And if finally, we abstract from the designate also and analyse only the relations between the expressions, we are in (logical) syntax' (Carnap 1942, p.9) (cited in (Recanati 2006)).

The emergence of 'Pragmatic Web' (an approach to support pragmatics dimension), succeeding the syntactic web (common web) and the semantic web, tried to get relevant information applying human interaction, that is, concern not only with the form but also with the meaning of the information. Since pragmatics is a field, rather than a discipline, and, additionally, belonging to the human communication, the tentative to *implement the pragmatics in an information system as its part is a paradox* (Ferreira 2013).

The production manager of a MS or company, has to deal, every time, with unpredictable behaviour of the MS/company organisation. Besides technological problems represent relevant causes for the deficient alignment among business and IT, personal factors represent the strongest argument, most of them related with the ability to well communicate (in sense of to be able to transmit und understand a message) or with the dynamic behaviour of the person. This complex and unpredictable behaviour is related to three main particularities (not by order of importance): (a) the linguistic competence on communication; (b) the behaviour of the responsible (manager, etc.) during context evolution and (c) the technological conditions. So, the capacity to communicate by itself could not be sufficient, unless it is accomplished with the ability to understand another speaker's intended meaning (which is the linguistic competence, and in particular the pragmatics competence).

However, this dynamic collaborative behaviour might be further enhanced by the emergent technological opportunities. However, paraphrasing Begnis (2010) 'the behaviour of the collaborators and the collaborative artefacts are affected by the ability of the infrastructure to facilitate desired and appropriate behaviours'. Since pragmatics is possible when human beings can share and react directly among themselves, if the information systems support it, they will be better aligned with user's interests and improve the result of the collaborative effort.

A pure technological process that take into account ubiquity and User Interfaces multimodality requirements need to transform and to enrich applications with real-time communication (RTC) services support (chats, videoconference, conference rooms, others). However, this new communicational capability by itself will never result in real effective systems unless used in a live and generative (not only transactional) communicational process involving pragmatics effects (Putnik et al. 2012; Ferreira et al. 2012). Paraphrasing Putnik et al. (2011) '(…) the human-to-human synchronous collaboration (video, audio, etc. and related auxiliary tools) which allows the natural involvement of the user on the co-creation/co-design (co-management) processes with other agents (humans) is the responsibility of the Pragmatics (…)'. Live communication, involving pragmatics effects represents a true meaning (semantics) co-creation (also, co-design) process. This co-creation process, that *connects* semantics and pragmatics, that is, the communication process producing the *effect*, of positive effective ascription of the intended semantics on other party, and therefore the effective interoperability between the two parties (at the ends of the communicational 'chain' are always humans, that is, users, even when considering an automatic technological equipment or systems), is described by Kaplan (1989) as:

> The fact that a word or phrase has a certain meaning clearly belongs to semantics. On the other hand, a claim about the basis for ascribing a certain meaning to a word or phrase does not belong to semantics. … Perhaps, because it relates to how the language is used, it should be categorized as part of … pragmatics … or perhaps, because it is a fact about semantics, as part of … Metasemantics (Kaplan 1989) (cited in Recanati (2006)).

In other words, the effective interoperability is based in conversation (discussion, dialogue, learning), meaning negotiation, consensus searching and similar.

Use of the term 'co-creation' is because the pragmatics aspects of communication (i.e. of effective interoperability) process result in a kind of new design of shared meaning between two actors (users), the design of interoperability mechanism, that is, for the case when interoperability is presented as the ontologies interoperability, in the new interpretation ontology as the instrument for two independent ontologies interoperability. In other words, paraphrasing Guiraud (1975), *pragmatics* could be considered as the 'maker' and 'inventor', figuratively, the *poiéte* ('maker') of new integration and/or interoperability instruments, providing higher levels of coherence with the environment, and social reality (Putnik and Putnik 2010).

Recently Zhang, Zhang and Cai (2015) still explore a technologically based solutions, instead. Although technology can offer relevant important information about these (and others) details, all these features are only effectively supported and assured if pragmatics instruments (like conversation) are effectively available in the system. Pragmatics instruments sustain a generative integration and interoperability of users and supports UMS concept (Putnik and Putnik 2010).

The interoperability surpasses the technological aspect of interoperability.

3. A review of typical information systems architectures state-of-the-art for UCM

3.1. Cloud-based manufacturing information systems architectures

There are already several proposals of architectures and frameworks to support CM (Wu et al. 2013). Being all multilayered architectures, their main difference comes essentially from the use of different terminologies, organisation of their layers and their logical components or modules and small functional details. In this paper, some of them are highlighted in order to present the actual state-of-the-art approach to the UCM underlying information system architecture.

Xu (2012) proposes a Cloud-based model layered (CBML) framework with four layers (Figure 1), having the physical resources (Manufacturing Resource Layer) brokered (identified, selected and organised) by the Virtual Services Layer and managed and governed by a Global Services Layer. Finally, the user has the Application Layer to construct manufacturing applications from the virtualised manufacturing resources.

Wu et al. (2012) proposes a *Cloud-Based Design and Manufacture* (CBDM) *Model* (Figure 2) where a cloud broker is an intermediate between the consumers and providers, and manages the use, performance and delivery of services.

Schaefer et al. (2012) propose a CBDM-*Distributed Infrastructure with Centralised Interfacing System* (DICIS) (Figure 3), where a communication asset already evidences the preoccupation with human interaction.

A more complex architecture for Cloud Manufacturing System (CMS) (Figure 4) comes from Tao et al. (2011) with 10 layers, where the 10 layers are: (1) Resource layer,

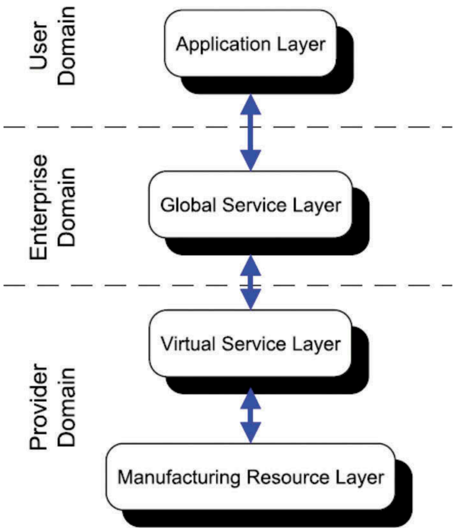

Figure 1. CBML four layer CM Architecture (Xu 2012).

(2) Perception layer, (3) Virtual resource layer, (4) Cloud service layer, (5) Application layer, (6) Portal layer, (7) Enterprise cooperation application layer, (8) Knowledge layer, (9) Cloud security layer and (10) Wider network layer (Internet, intranet, internet of things).

However, even being a relevant joint between CC capacity and cloud services, where users can easily communicate (within the transactional communication paradigm), we think it does not allow human active participation and consequent co-design of interoperability and on collaboration processes. The users use the system, passively, in the sense that they use what is given and the users can not influence the provided service design to improve the interoperability.

3.2. A generalised model for traditional transactional architectures

Information systems, distributed or not, are sustained by architecture models. Ferreira (2013) presented a generalised model for a traditional Transactional Multilayer Architecture (Figure 5) where layers, patterns and standards Application Programming Interface (API) prepare the system to sufficiently support any specified requirement and easily integrate future ones, granting its flexibility, robustness and interoperability. Interoperability here means interaction through transaction essentially, offering enhanced rich interfaces to multiples devices (multimodal systems), and the human behaves as passive user, outside the architecture.

The integration of new applications in the supporting platform is assured by specific middleware that grant data, business rules or front-end integrations, essentially. Besides that technological detail, if the users need to

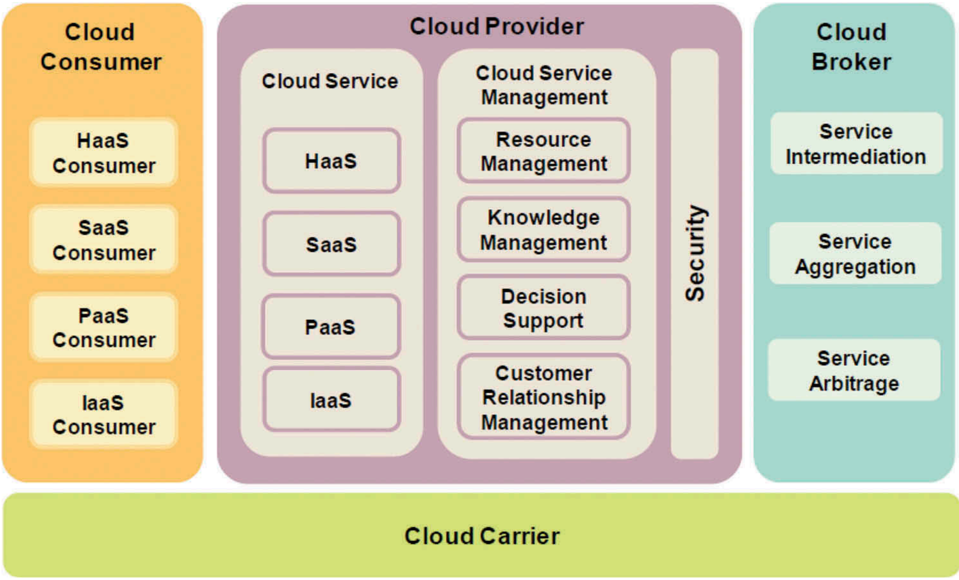

Figure 2. CBDM Conceptual Reference Model (Wu et al. 2012).

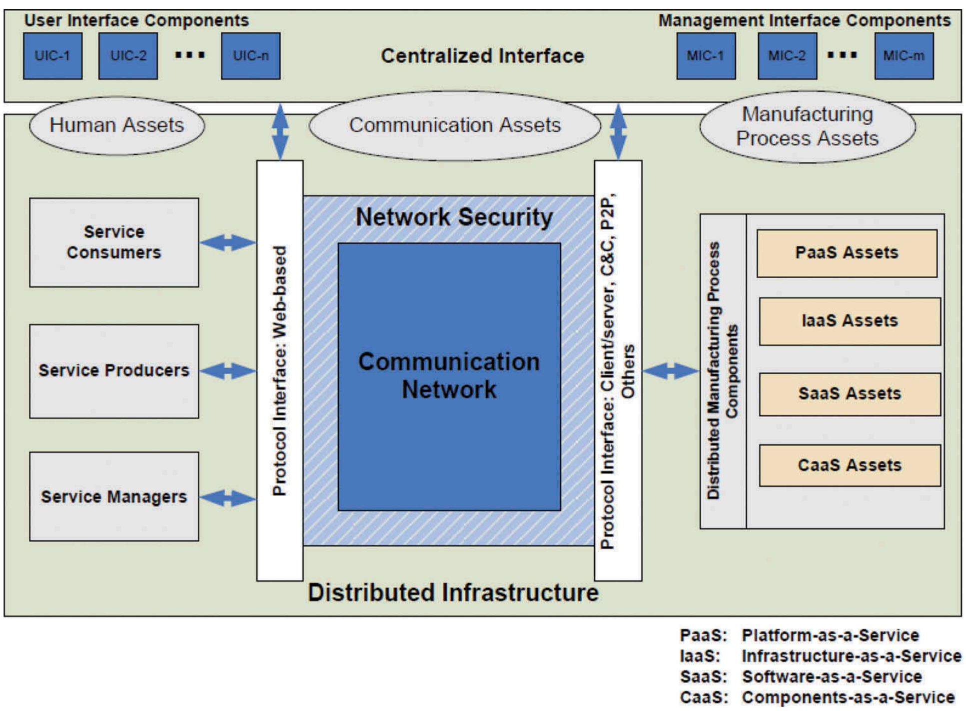

Figure 3. The DICIS model for CBDM (Schaefer et al. 2012).

'integrate' (better, use) the systems, they need to learn or to adapt to it, following wizards, for instance.

Although traditional transactional architectures demonstrated their capacity to handle efficient information systems integration, the requirement to deal with not initially specified applications (or APIs) revel their handicaps. The business globalisation brings new stakeholders and their applications to participate in relations. The UCM, as well as eBusiness and eCommerce as the business 'layers' of UCM, represents such step and the need to assimilate and follow it was (and still is) a delicate and hard decision to take. The existent architectures are not prepared to handle it. So this capacity is a requirement and Pragmatics embedded in Services-Oriented Architecture represents the necessary shift.

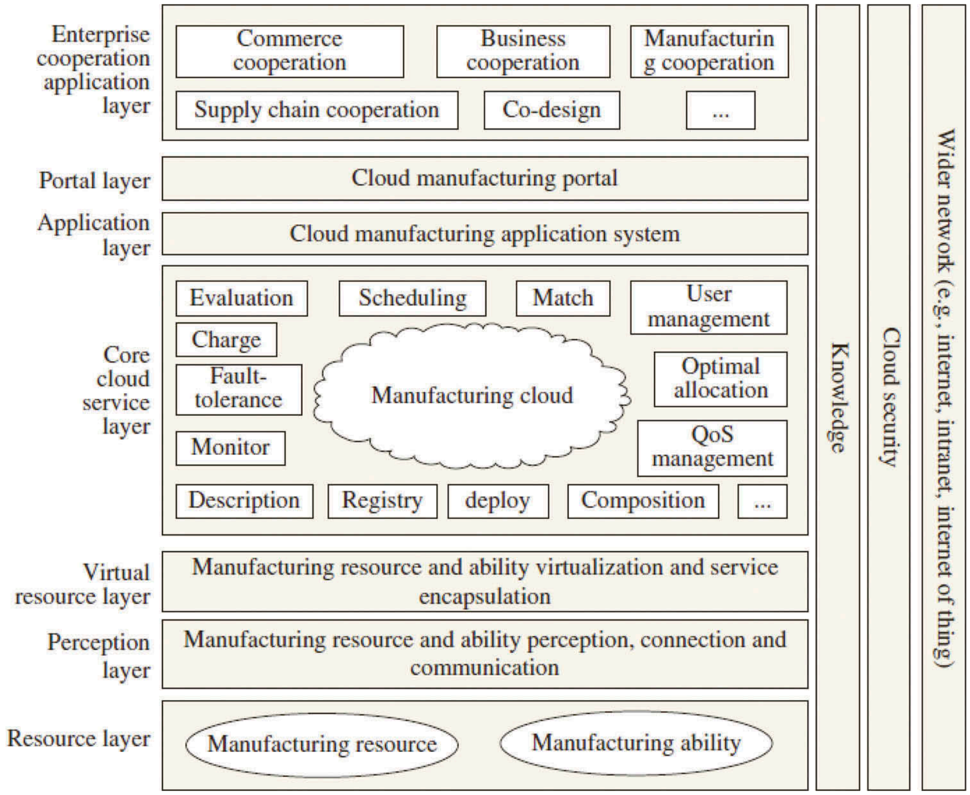

Figure 4. Architecture for Cloud Manufacturing System (CMS) (Tao et al. 2011).

4. Extending the cloud-based architecture for UMS with pragmatics renderer

4.1. The required communicational architecture

The traditional (and transactional) architecture (presented in previous Figure 5) is based on suitable middleware that sustains the needed (technological) interoperability. In this architecture, the human behaves as mere user from outside the system, making impossible to have an effective human participation. Indeed, the real (human) user requirements are not well supported by the existing systems, because they are not easily tangible and technically specifiable. The user needs to adapt continuously to the system and to follow the eventually existing system wizards. He cannot have his own reasoning and interact humanly with the system in order to adapt its design to fit better and effectively his true needs. The Pragmatics renderer, may also include the widely used collaborative design tools, such as the tools used in concurrent and collaborative engineering for the purpose of co-design, or co-creation, of the interoperability instruments (the objective of the architecture is not providing more efficient 'concurrent engineering' but interoperability for different applications for concurrent engineering services. However, the concurrent engineering tools could be used for co-design of the interoperability instruments as referred).

To align the system to human, the system architecture needs to support human-to-human real and synchronous collaboration that allows the co-exploration (co-creation, co-design) of the system with other agents (humans) (including the system designer and/or provider). Thus, the required architecture needs to be effectively communicational. Not to support communication services as transactions only (previous Figure 5), where any device used to explore existing applications behaves as a mere tool to interact with the system and not as pragmatic supporting mechanism, but to have direct human participation and collaboration in any interested phase instead. Assuming this, Pragmatics and Collaboration engines allied to effective brokering mechanisms need to be implemented. (The evidence of this comes from social networks success and their use for our own interest in a completely autonomous way.)

The proposed architecture, that includes the pragmatics supporting modules, is, in fact, a semiotic-based communicational architecture (Figure 6). We name it as *Communicational Architecture with Pragmatics Renderer* for UCM (CAPR).

It basically needs to support the transactional behaviour (using nowadays supporting technologies) and Pragmatics services, that means: (a) rich client interfaces with sufficient interaction to allow user agility and

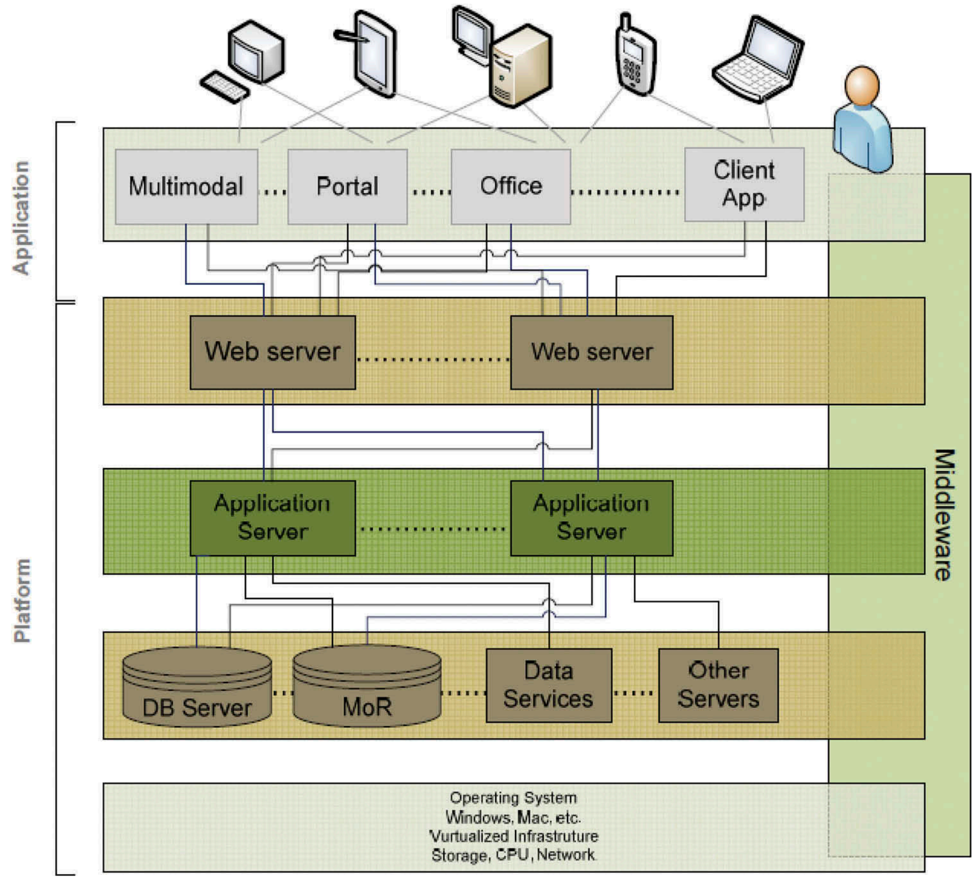

Figure 5. Traditional transactional multilayer architecture.

competence, (b) multimodal, for any client device classes (smartphones, tablets, small widgets, etc.) support and (c) communicational to allow pragmatics, where human-to-human real interaction is completely supported.

Concerning the need for communicational channels, as the instruments of the Pragmatics renderer, considering usual request for 'real-time communications among the parties involved' it should be called for attention that the real-time automated communication among the parties is only possible if exist standards for data format and respective shared ontologies. This case, of the 'real-time automated communication among the parties', is considered under the so-called transactional architectures and these architectures are not to eliminate. However, if there are no standards, or applications do not use the same standards, or do not share the ontologies, then any interoperability is impossible, and the request for the 'real-time automated communication among the parties' is simply impossible, which is the case of virtually all cloud and UC based applications, and the question of 'time', that is, the request for 'efficiency', is irrelevant. We witness this situations daily when attempt to use different applications. In this situation, what we need is effectiveness, that is, we need to achieve the effective interoperability.

With the proposed architecture with embedded Pragmatics renderer, the active user participation is essential to overcome technological interoperability problems. To ensure effectiveness, Pragmatics must be supported and generating effective middleware-like mechanisms should be possible. This is not only offering chat or videoconferences like services, but act immediately if needed, promoting co-decision and co-creation, or co-design, of the interoperability mechanism.

Following, Section 4.2 presents the organisational model, that is, presents the organisational elements of the communicational architecture in Figure 6, while Section 4.3 explores the model, that is, the logical and functional elements of the model and Section 4.4 explores its implementation. As it happens with any system, the implementation aggregates several functionalities in components. (Figure 8 represents the technological components, only!)

4.2. *Organisational model*

Besides the common transactional architectures components (operating systems, web servers, etc.), the proposed architecture is globally structured in three main components: (a) Platform, (b) Application and (c) Communication.

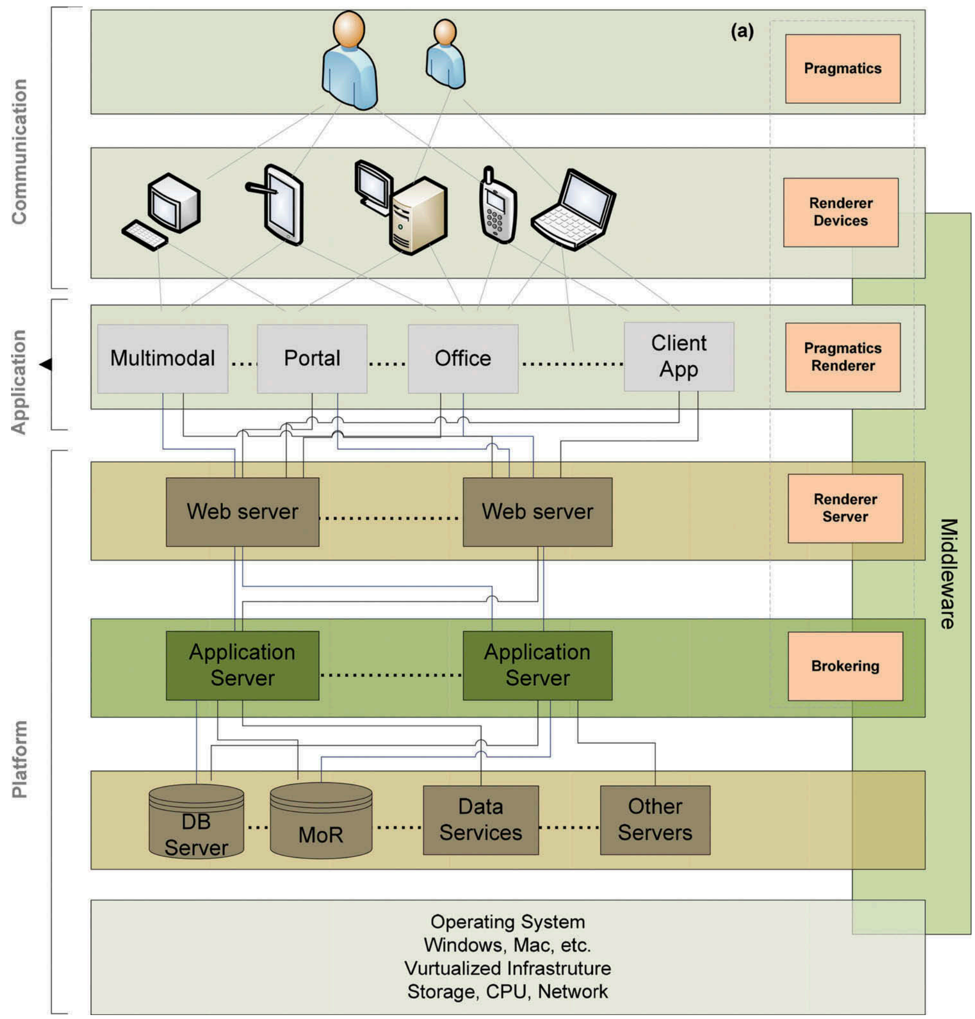

Figure 6. Communicational architecture where devices are Pragmatics renderers.

The *Platform component* represents the base technological infrastructure that sustains the required network services, data management and services integration. Being a web infrastructure that have CC as its main support, the typical transactional behaviour depends on specific middleware that assures services interoperability, composition and coordination. To assure efficient portability and multimodality, RESTful services with JSON/XML data integration are the main integration patterns to follow here. The critical brokering processes depend on these interoperability capacity.

The *Application component* represents the several client applications that will be enabled to integrate the supported system. The architecture is prepared: (a) to handle multimodality, that means, any service can be supported by any existent platform or device, being mobile or not; (b) to offer Rich Internet Application (RIA) services since it is essential to follow the emergent social networks user experiences and (c) to handle responsive frontends, that means, the interaction services must be prepared to be used in (different screen resolution, for instance inherent to) multiple existing technologically heterogeneous devices.

The *Communication (Communicational or Semiotic) component* represents the main innovative part of this communicational architecture. It is organised in three levels: (a) device level – *renderer device* – which allow user 'to use' pragmatics with the system, (b) application level – *pragmatics renderer* – a set of tools which allow to the user a pragmatics based interaction and (c) application server – *renderer server* – level which is responsible to implement the Pragmatics Engine, the entity which will support all pragmatics services.

4.3. *Logical and functional model*

The proposed architecture (a) must be independent from the supporting technological platform, must consider the needed, (b) to allow participants to decide autonomously and (c) the overall management of the network of participants must be sustainable, since resources management are managed by their promoter, mainly. These intended requirements come from the emergent web 3.0 applications and social networks behaviour, that is, their sustainability is the 'responsibility' of its users by themselves.

Functionally, the infrastructure has mechanisms that allow membership registration autonomously and independently. A new service, seen also as a resource (it can be people, machines, time, a set of tasks, a simple task, etc.) exists in the platform if any promoter or provider has registered it. The architecture allows the registration (integration in the platform using meta-information) of new services, where its management is partly the responsibility of the promoter, having the platform the responsibility of its integration into search engines (brokering) and rating (rating), basically. Multiple integration tools allow services composition and coordination, where co-creation (co-design, co-decision – on interoperability and integration) 'tools' support the required effectiveness off all the system, that is, *resource's owner – resource's costumer* in-time communication and co-decision.

Considering all the architecture's components, three of them must be highlighted: *Pragmatics Renderer, Repository* services and the *Brokering*.

The *Pragmatics Renderer* works as a co-creation (co-design, co-decision decision – on interoperability and integration) enabler. Basically it is consisted of a set of integrated communicational channels that makes the 'bridge' between the user/devices. It is provided a set of collaboration mechanisms, under synchronous bidirectional channels, and multiuser sessions with recording and historical support. Real-time video, chat, direct visual talking, rooms, spaces, etc., are some of the enabled services. In practice, in the case of complementary information needed for a particular resource (if it is operational, busy, etc.), it can be obtained by interacting directly with his owner or provider.

Each resource needs to be previously and properly registered in the *Repository* (the Market of Resources base), being the registration process autonomous and completely independent (in time and context) of the selection process. The registration follows automatic or explicit mechanisms (manuals) of cataloguing or classification, using as meta-information terms from the domain ontology. For the classification process, the user feedback can be important too. Each registered resource is accomplished via attributes/values tuples, complemented with other semantic data. For example, the power of a machine has an attribute 'watts'; or a travel has an attribute 'distance' to which is associated a numerical value.

Pragmatics requires that two users can participate in the system in a natural way, that is, talking face-to-face with the other(s), at that time, being the system only the medium (mediation tool) for this to be possible. So, the architecture sought to ensure pragmatics on the system in two ways: (a) mediating existing communication channels and (b) implementing new channels of communication. Communicational channels mean video chat, video conference, audio chat, email and several others.

In mediation, the system allows each resource owner to appeal the registration and publication of the communicative channel that has and wants to make public (Figure 7). Although their state (on, off, faulted, etc.) is moderated by the owner, the system offers the user the possibility to use them whenever they are available.

Thus, the system 'finds' the active channels resource and offers them. In practice, analysing the e-mail service, for instance, the application prepares the message to send and dispatches it to the daemon service responsible for sending it. If you need to connect video (video streaming) with the owner of the resource, this communicational channel is established.

The search and selection of a desired resource is only possible through appropriate selection procedures – *Brokering* – in the resources repository. This selection is conditioned by multiple factors, from the availability and quality of performed services up to the user's requirements. To bridge the gap between the interest of the user and the information that it wishes for, the search engine is still sensitive to context and to the own profile of who is searching. For instance, the decision on which resource to choose can be moderated by the end user. Thus, the brokering service works: (a) as a tool to help the selection and composition of resources that best satisfy the user's requirements and expectations and (b) as a tool to help on the dynamic reconfiguration management, inherent to the constant change of requirements and the state of resources, ensuring the better alignment between the virtual enterprise (VE) thus formed and the task(s) to execute.

The quality of service is virtually the most important factor in resource selection. In practice, the user wants to

Figure 7. Communicational channels.

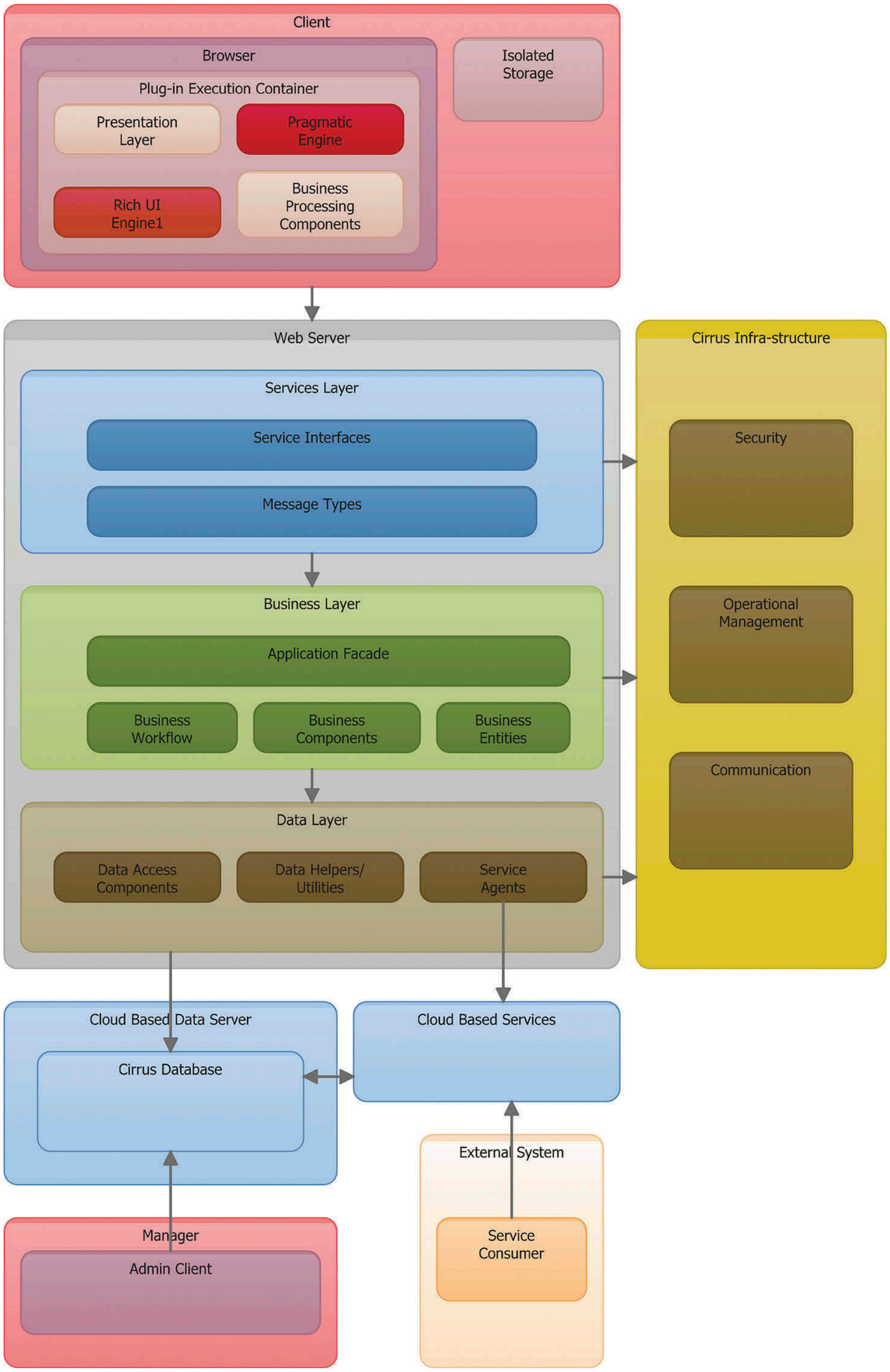

Figure 8. Architecture's Technological Components – MVC/RIA Pattern.

see well executed the given task and do not want to worry about how it is performed. To register this quality, additional semantics information must be appended to each resource.

It is important to note that the dynamic reconfiguration of resources is due mainly to: (1) performance and availability of providers; (2) changes in user requirements (and therefore changing the task as a whole) and (3) not controllable external factors. Faced with unforeseen situations (external factors), the system should reconfigure itself to ensure the total satisfaction of user requirements.

4.4. Technological platform

In a global technology perspective, the architecture must support secure real-time collaboration and synchronous or asynchronous integration between processes, based on open-source technologies mainly. Since we want a scalable solution (see Putnik et al. 2013), where the key word is ubiquity of services, and whose intended sustainability is derived from the user participation, that is, cannot be fully controlled by us, a cloud-based infrastructure with social network like platform, is required.

Since we want to maintain some flexibility and agility, mainly in the stakeholder's side, we are conscious that not all services will be cloud-based. So a bet on a hybrid strategy guarantees, on the one hand, and economy of scale and ubiquity, and some control considered essential, on the other. The existing information systems of the multiple stakeholders are not preventing accession and the decision to 'be or get out' is for them. We focus a little on buzzword *lowering transloading cost* in the context of software architecture: *Localised Optimisation through Selective Specialisation* (LOtSS) in which the company optimises its services, deciding what to develop internally or adopting existing solutions.

A relevant technological implementation detail lies on the brokering of resources process. It will be mainly supported by cloud services provided as a RESTful and SOAP API that must allow synchronous or asynchronous interoperability, transactional or communicational, and ready to be integrated in any external application. In addition, the API still has complementary resource management services (registration, removal and update) for the repository.

In the situation where a user needs to interact with any other(s), there are integrated RTC services (video, audio, etc.), whiteboards for collaboration, etc., that each resource makes available, ensuring the proper communication between people.

The layered structure of the technological architecture resulted from the combination of the Model-View-Controller (MVC) pattern and RIA. Figure 8 shows the proposed technological components, the responsibilities they have and the interaction they establish among them. It can be seen clearly a part that is supported on the server side (Web Server and Cloud-based Server components) and another that is supported on the client side, being it a mobile application or not, to use (Client) or to manage the system (Manager).

Whenever it is necessary to integrate the various layers, a transverse layer takes care of global services namely Security, Operational Management and Communication. The integration of external applications (Service Consumers) is possible thanks to the developed web services (Cloud-Based Services).

System users (customers) interact with the application through the Presentation Layer, and directly between them through the services provided by Pragmatics Engine component. External systems interact through the layers of services, whether they are in the cloud (Cloud Services) or in the application itself (Service Layer). Both layers, Presentation and Services must 'comply with' the rules implemented in the Business Layer. This rule is maintained by the MVC component.

An external system can be another web application, a mobile application, a mash-up to integrate in a dashboard, etc.

The communication services will be mainly supported by existing P2P technology and the direct interoperability needs to integrate existing communication network solutions such as Skype, Google Talk, MSN or others, all at once. The innovative communication services are explored using OpenCV image processing library and RTC technologies.

The registration and services discovery is in charge of the API cloud engine. It supports an advanced brokering mechanism over registered services which represent the Market of Services or Market of Resources. The dynamic reconfiguration and inherent resource's ranking are two of the multiples features that the broker needs to support. This broker is implemented using CC model and the code behind should be used following the cloud services programming model. In practice, this Market of Resources will behave as a PaaS.

Having cloud services as the main supporting architecture, the use of cloud engine API will be determinant to develop a federated or community cloud. The cloud services will provide a RESTfull API to support their use and composition. In this context, pragmatics supporting services will behave as SaaS in the cloud.

Figure 9 presents the main technologies that can support each of the layers/components, highlightening the parts that support the *Pragmatics Engine*. This will be a set of services that each resource offers to allow the establishment of a direct synchronous communication channels (*email, chat, video conference, chat rooms, white boards*, etc.). Considering the main existing technologies, and focusing on the main ones that offer real-time synchronous communication, such as *XMPP, SignalR* and *WebRTC* (*HTML5 WebSockets Protocol*), show that it has not yet reached a satisfactory capacity of interaction, especially if we refer to non-proprietary technologies, as has been the case.

In short, the architecture can be implemented with the emerging technologies for web 3.0 development, including, Table 1.

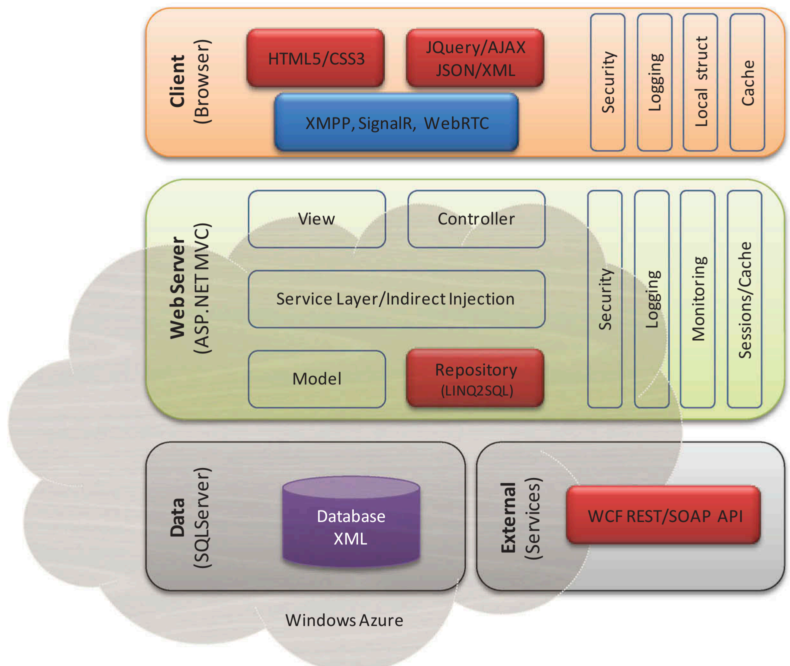

Figure 9. Architecture technological support.

Table 1. Emerging technologies for web 3.0 development for implementation of the UCM cloud-based architecture.

RIA web pages (View)	*CSS3, JQuery, AJAX, JSON, XML/CSS3, JQuery, AJAX, JSON, XML*
Asynchronous requests	*AJAX*
Parallel requests	*Threads/SignalR*
Server Business Logic (Controller)	*.NET (C#)*
RESTful cloud Services	*.NET(C#), XML, JSON*
Repository (Model)	*.NET (C#), LINQ2SQL*
Database	*MS SQL Server, XML,JSON*
Cloud 'housing'	*Windows Azure*

5. Prototype implementation

The dynamic charisma and complexity of manufacturing economic activity is sensible to continuous changes due to multiple internal (resources failure, insufficient resources, production errors, organisational dynamics, etc.) or external (climatic conditions, legal regulations, others) application factors, implying the production process reconfiguration. Three main entities were identified in the process: (a) the Client (which requires the product); (b) the Company (that provide a set of resources) and (c) the Market of Resources (that mediate the company/client relation, ensuring the resources supply) hosted in a cloud.

Given the complexity of the area and multiplicity of contexts, and considering the architecture applicability as the main goal, it is decided to model the entities only with the information considered essential to the services that we wanted to implement. The innovative ones are: (i) resource registration; (ii) communicational channels registration; (iii) tasks that resources can execute; (iv) searching of resources by state, classification, sector, geo-referenced information; (v) changing the status of the resource; (vi) production plan reconfiguration; (vii) resources geo-referencing; (viii) production plan geo-referencing and (ix) use of integrated communicational channels. In practice, all these integrated services sustaining all public information of any resource (Figure 10).

It is assumed that there is an instance of the Manufacturing Market of Resources (MMR) hosted in the cloud. Any interaction with it is done via a MMR API it provides. Following cloud service patterns, the MMR API is operational for several other applications. Having this, three main components are implemented: (a) the Market of Resources; (b) the broker and (c) the Pragmatics Engine. All of them are explored using several services integrated in two distinct applications: a Web Portal and a Mobile Windows Phone. The use of (a) geo-referenced (map-based) data on client resource monitoring and dynamic reconfigurations and (b) integrated communication channels to allow participants' direct communication represent the main and innovative features on the Web Portal. Furthermore, the Mobile application, behaving as an app, allows the resource promoter to register their own resources, essentially.

For simplification, it is decided to 'join' on this website two profiles: (a) the customer profile (registered or not) and the profile of the system manager. In practice,

Figure 10. Resource public information.

offering *back-office* services (to the Manager) and *front office* for the normal user.

Figures 11–15 show the main implemented features. The prototype figures represent some possible scenarios of intended services implementation. No relevance to manufacturing resources interaction was particularly done, but the enrichment of the services with communication channels instead.

5.1. *Support to effectiveness*

As described previously, technologically this base architecture follows MVC pattern and the interface follows the RIA Presentation Design Pattern (Cunningham 2003), having resources and their governance services (Model) hosted in cloud, cloud-based Representational State Transfer (REST) services to support business rules and actions (Controller) and multimodal RIA Presentation Layer (View) to allow multimodal device interaction with.

Functionally each manufacturing resource represents a service (or many) that is hosted on cloud. It has an interface description language that allows its discovery, an interoperability specification to follow and an (REST) API that allows its use and integration with (or, use by) other resources. Since they 'Reside' in cloud they are

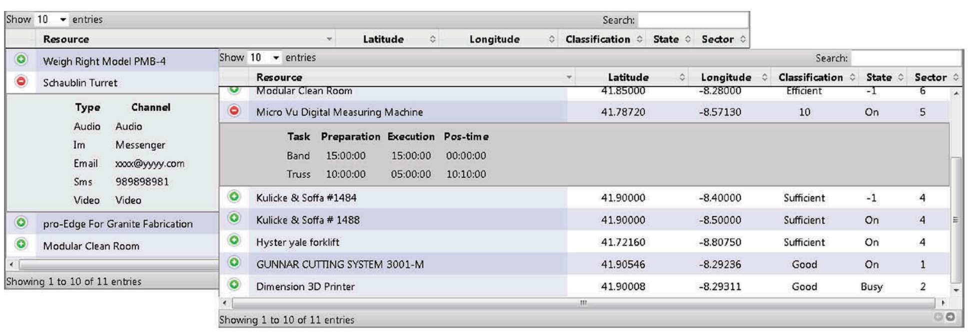

Figure 11. Communication channels and affected tasks for a particular resource.

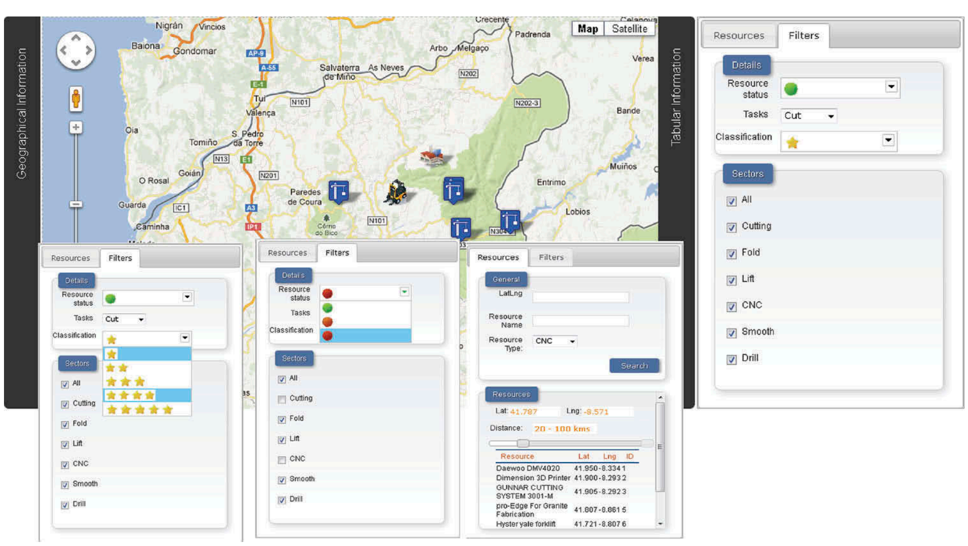

Figure 12. Broker filtering service.

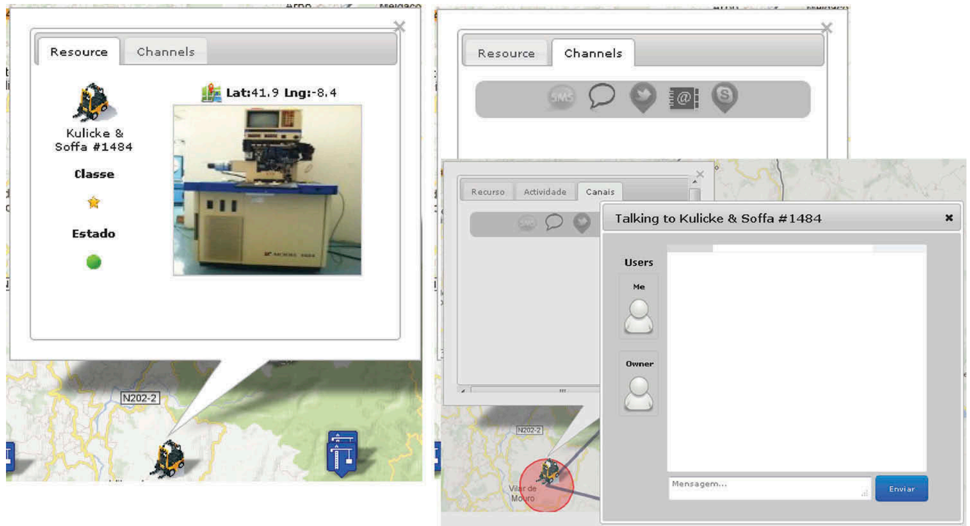

Figure 13. Detailed resource information and web chat with the owner of the resource.

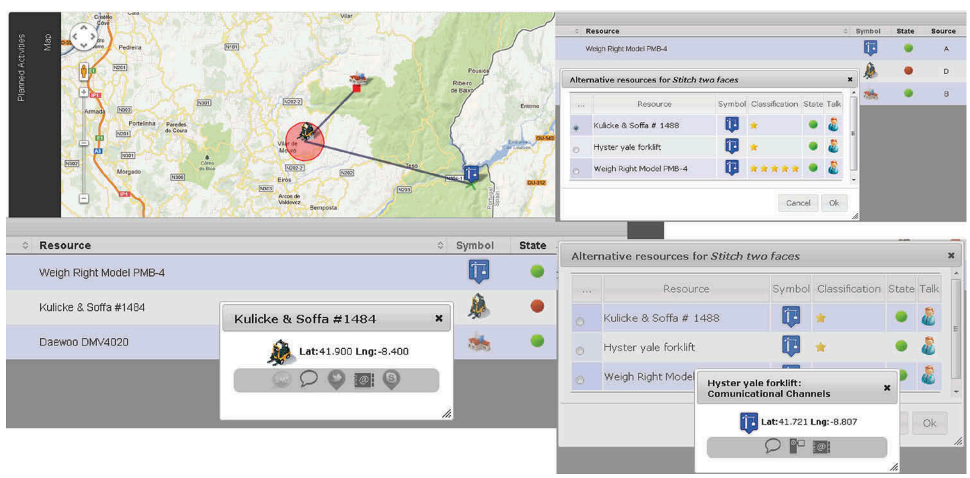

Figure 14. Dynamic reconfiguration and selection of alternative resources.

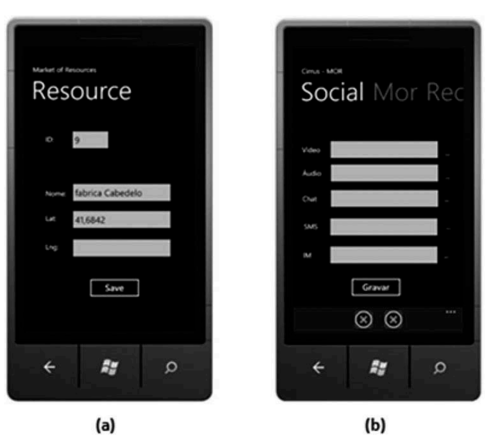

Figure 15. Resources registration on the mobile application.

named *cloudlets*. A RIA dashboard (Figure 16) using Web 3.0 technology allows web components (apps) integration, and open-source communication technology, such as WebRTC and Web Media Capture. Thus, this real alignment with personal point of view sustains the effectiveness of the system.

6. Comparison of the proposed architecture with the reviewed architectures

Table 2 presents a comparison between multitier cloud-based architectures that, as described before, represent relevant initiatives to support CM. They are faced against the main requirements that a communicational architecture with Pragmatics renderers must support. Table 2

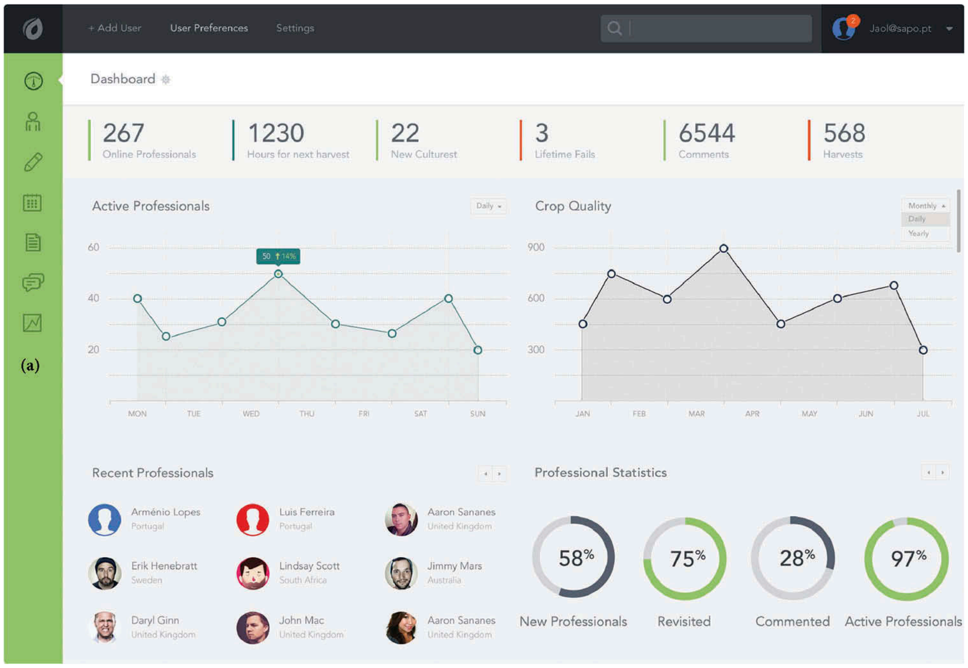

Figure 16. Dashboard for effectiveness.

Table 2. Comparison of information systems architectures for UCM.

	Requirements									
Architectures	Reconfiguration	Cloud based	RIA	Co-creation	Cloud services	Communications Assets	Human assets	Opened API	Effectiveness	Pragmatics
CBML (Xu 2012)		✓			✓					
CBDM (Wu et al. 2012		✓			✓					
DICIS (Schaefer et al. 2012)		✓			✓	✓	✓			
CMS (Tao et al. 2011)	✓	✓		✓	✓	✓	✓			
CAPR (Ferreira 2013)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

demonstrates the innovativeness of the architecture proposed in this paper (CAPR) and its much richer potential for supporting effectiveness in interoperability and integration, and sustainability, of UCM

All reviewed architectures are interesting and well-documented technological proposals. The base selection criteria were cloud-based and cloud services support. However and because our main focus is pragmatics support, criteria like RIA, communication and human assets and effectiveness were also considered. Having an integration opened API that allows reconfiguration and middleware development, like a Software Development Kit, are convenient too.

It is worth to refer the MVC-based architectures too (Wang, Shen and Lang 2004; Wang 2008). The main relevance of these architectures is not its implementation pattern, as is MVC. Thus, the reference to others MVC-based architectures was not the main scope of this project. Furthermore, Wise-ShopFloor is another technology-based proposal only, where the user acts as a passive participant. We want to emphasise the active use participation, instead.

7. Conclusions

Usually it is believed that technology is the base to solve many of the global business (and any others) problems.

Considering the integration (or interoperability) problems that arises when new technological solutions need to 'cooperate', the question is mainly focused on the lack of new standards, patterns or efficient middleware. If the problem goes to semantic contexts (different terminologies or meanings), the use of ontologies, thesaurus, taxonomies or others, will be enough.

However, this research is based on the assumption that semantics by itself cannot ensure the correct relation between concepts, and to effectively integrate semantics there are necessary mechanisms that allow direct participation of all stakeholders. Co-creation processes are essential. Communicational channels are the bases.

Although traditional architectures sustain that technologies by themselves are able to support any interoperability requirement, we defend the position that communicational architectures, where exist efficient and integrated communications mechanisms that allow humans to directly interact with each other, are more efficient to ensure effective integration on dynamic reconfiguration scenarios. Besides the evidences reported in literature (in particular in linguistics as the fundamental scientific discipline for semiotics and pragmatics-based approach), this position is sustained through a specification of a communicational architecture model and through the prototype developed to validate the proposed communicational architecture model, following the Semiotics-based Manufacturing System Integration paradigm (Putnik 2010b; Putnik and Putnik 2010).

Relatively to UCM business activity: (a) a distributed platform technology independent based on cloud an their cloud services; (b) the existence of integrated communication mechanisms to allow pragmatics where human behave truly effective, and (c) a communicational architecture with effective brokering mechanisms represent the main characteristics to sustain a business activity that faces dynamic and reconfigurable scenarios, and ubiquitous information system.

As final remark, the architecture proposed be used to help systems architects and managers to understand the relevance of more than technological aspects, namely social and human ones, to increase the effectiveness of information systems for UCM and information systems in general.

The future work could be developed in a number of dimensions. For sure one of the dimensions is further technological development of the proposed architecture, in order to follow the development of underlying information and communications technologies. One of the directions will be enriching the proposed architecture model with emergent federated social networks where effective federated ubiquitous services must be granted.

However, virtually more important dimension for development is further exploration of communicational systems as the co-creation instruments and considering an UCM as the complex system. The complex systems are not technology dependent but human centred.

The third dimension of the future work should address the techno-economical aspects of the proposed architecture as the architecture is not 'efficiency' oriented but oriented to effectiveness. In other words, it is necessary to explore the economic aspects as well as the corresponded innovative business models in order to make the underlying paradigm sustainable in economic terms.

In a more operational context, the future work must focus on implementation of the proposal architecture and quantitative analysis of empirical data.

Acknowledgements

The work is supported by the Portuguese National Funding Agency for science, research and technology (FCT), (1) Grant No. UID/CEC/00319/2013, and (2) 'Ph.D. Scholarship Grant' reference SFRH/BD/85672/2012.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The work is supported by the Portuguese National Funding Agency for science, research and technology (FCT), (1) Grant No. UID/CEC/00319/2013, and (2) 'Ph.D. Scholarship Grant' reference SFRH/BD/85672/2012.

References

Becker, M. B. 2012. *Interoperability Case Study-Cloud Computing*. Berkman Center for Internet & Society Research. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2046987

Begnis, G. 2010. Behavioral Facilitation. http://encyclopedia.jrank.org/articles/pages/6675/Behavioral-Facilitation.html

Carnap, R. 1942. *Introduction to Semantics*. Cambridge, MA: Harvard University Press.

Cha, S.-K., and -J.-J. Yoo. 2008. "U-Manufacturing Model & Application System Using RFID/USN, Mobile and Internet Technology." *10th International Conference on Advanced Communication Technology, ICACT 2008*, 79–83. IEEE, Phoenix Park, Korea (South).

Cheng, Y., F. Tao, L. Zhang, X. Zhang, G. H. Xi, and D. Zhao. 2010. "Study on the Utility Model and Utility Equilibrium of Resource Service Transaction in Cloud Manufacturing." *Industrial Engineering and Engineering Management (IEEM), 2010 IEEE International Conference on*. IEEE, Macao.

Cunningham, W. 2003. *Enterprise Solution Patterns Using Microsoft .NET (Vol. 2.0)*. Redmont, Washington: Microsoft.

Ferreira, L. 2013. *Architectures for Integration of Information Systems under Conditions of Dynamic Reconfiguration of Virtual Enterprises. Production Systems*. Braga: University of Minho.

Ferreira, L., G. D. Putnik, M. M. Cruz-Cunha, and Z. Putnik. 2012. "Towards Effective Tourism Dynamic Packages."

Information Resources Management Journal (IRMJ) 25: 1–21. doi:10.4018/IRMJ.

Foust, B.-J. 1975. "Ubiquitous Manufacturing." *Annals of the Association of American Geographers* 65 (1): 13–17. doi:10.1111/j.1467-8306.1975.tb01013.x.

Guiraud, P. 1975. *Semiology.* London: Routledge and Kegan Paul Ltd. (first published as: La Sémiologie, Presses Universitaires de France, 1971.

Kaplan, D. 1989. "Demonstratives." In *Themes from Kaplan*, edited by J. Almog, H. Wettstein, and J. Perry. New York: Oxford University Press.

Kim, C., R. Johnson, G. Roussos, and S.-J. Lee. 2012. "A Cross-Country Comparison of the Adoption of Ubiquitous Supply Chain Management." *Personal and Ubiquitous Computing* 16 (6): 717–727. doi:10.1007/s00779-011-0438-5.

Kim, D.-H., and J.-Y. Song. 2008. "Mobile and Remote Operation Form2mapplication in Upcoming U-Manufacturing." *Journal of Mechanical Science and Technology* 22 (1): 12–24. doi:10.1007/s12206-007-1002-0.

Li, W., and J. Mehnen (Eds.). 2013. *Cloud Manufacturing*. Springer Verlag. London, Heildelberg, New York, Dordrecht.

Morris, C. 1938. "Foundations of the Theory of Signs." In *International Encyclopaedia of Unified Science I*, edited by C. Morris, 77–138. Chicago: University of Chicago Press.

Murakami, T., and A. Fujinuma 2000. Ubiquitous Networking: Towards a New Paradigm. *Nomura Research Institute Papers*, No. 2, NRI - Nomura Research Institute, Ltd., Tokyo.

Onosato, M., N. Suppen, K. Teramoto, and K. Iwata. 2001. "From a Factory to a Person: One Approach for Sustainable and Service-Oriented Manufacturing Scheme." *Proceedings Ecodesign 2001: Second International Symposium on Environmentally Conscious Design and Inverse Manufacturing* 2001: 666–671.

Putnik, G., A. Sluga, H. ElMaraghy, R. Teti, Y. Koren, T. Tolio, and B. Hon. 2013. "Scalability in Manufacturing Systems Design and Operation: State-Of-The-Art and Future Developments Roadmap." *CIRP Annals-Manufacturing Technology* 62 (2): 751–774. doi:10.1016/j.cirp.2013.05.002.

Putnik, G. D. 2010a. Ubiquitous Manufacturing Systems Vs. Ubiquitous Manufacturing Systems: Two Paradigms. In R. Teti edited by *Proceedings of the CIRP ICME '10-7th CIRP International Conference on Intelligent Computation in Manufacturing Engineering - Innovative and Cognitive Production Technology and Systems*. University of Naples Federico II, Naples.

Putnik, G. D. 2010b. "Semiotics-Based Manufacturing System Integration." *International Journal of Computer Integrated Manufacturing* 23 (8–9): 687–690. doi:10.1080/0951192X.2010.513809.

Putnik, G. D., L. Ferreira, M. M. Cruz-Cunha, Z. Putnik, H. Castro, and V. Shah. 2012. "User Pragmatics for Effective Ontologies Interoperability for Cloud-Based Applications: The Case of Ubiquitous Manufacturing System UI." *Business Information Technology.*

Putnik, G. D., L. Ferreira, V. Shah, Z. Putnik, H. Castro, M. M. Cruz-Cunha, and L. Varela. 2011. "Effective Service Dynamic Packages for Ubiquitous Manufacturing System." Virtual and Networked Organizations, Emergent Technologies and Tools. Springer Berlin Heidelberg, 2012. 207–219.

Putnik, G. D., and Z. Putnik. 2010. "A Semiotic Framework for Manufacturing Systems Integration – Part I: Generative Integration Model." *International Journal of Computer Integrated Manufacturing* 23 (8–9, August–September): 691–709. doi:10.1080/0951192X.2010.510292.

Recanati, F. 2006. "Pragmatics and Semantics." *The Handbook of Pragmatics* 442–462, edited by Laurence R. Horn and Gregory Ward, Blackwell Publishing, Oxford.

Saussure, F. D. 1916. *Course in General LinguisticS. (A. S. Charler Bally Albert Riedlinger, Ed.)*. The Library of the University of California. Edition 1959. Los Angeles.

Schaefer, D., J. L. Thames, R. D. Wellman, D. Wu, S. Yim, and D. Rosen. 2012. "Distributed Collaborative Design and Manufacture in the Cloud—Motivation, Infrastructure, and Education." In *American Society for Engineering Education Annual Conference*, Paper #AC2012-3017, San Antonio, USA, American Society for Engineering Education.

Serrano, V., and T. Fischer. 2007. "Collaborative Innovation in Ubiquitous Systems." *Journal Intell-Manuf* 18: 599–615. doi:10.1007/s10845-007-0064-2.

Tao, F., L. Zhang, V. C. Venkatesh, Y. Luo, and Y. Cheng. 2011. "Cloud Manufacturing: A Computing and Service-Oriented Manufacturing Model." *Journal of Engineering Manufacture, Part B* 225: 1969–1976. doi:10.1177/0954405411405575.

Wang, L. 2008. "Wise-Shopfloor: An Integrated Approach for Web-Based Collaborative Manufacturing." *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews* 38 (4): 562–573. doi:10.1109/TSMCC.2008.923868.

Wang, L., W. Shen, and S. Lang. 2004. "Wise-Shopfloor: A Web-Based and Sensor-Driven E-Shop Floor, Transactions of the ASME." *Journal of Computing and Information Science in Engineering* 4 (1): 56–60. doi:10.1115/1.1647122.

Wu, D., M. J. Greer, D. W. Rosen, and D. Schaefer. 2013. "Cloud Manufacturing: Strategic Vision and State-Of-The-Art." *Journal of Manufacturing Systems* 32 (4): 564–579. doi:10.1016/j.jmsy.2013.04.008.

Wu, D., J. L. Thames, D. W. Rosen, and D. Schaefer. 2012. "Towards a Cloud-Based Design and Manufacturing Paradigm: Looking Backward, Looking Forward." In *International Design Engineering Technical Conference & Computers and Information in Engineering Conference - IDETC/CIE 2012* (pp. 1–14). Chicago, Illinois, USA: ASME.

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.

Zhang, W. Y., S. Zhang, and M. Cai. 2015. Personalized Manufacturing Service Recommendation Using Semantics-based Collaborative Filtering. *Concurrent Engineering* 23: 166–179. SAGE Publications.