ISSN: 2230-9926

Available online at http://www.journalijdr.com

IJDR

International Journal of Development Research
Vol. 10, Issue, 11, pp. 41864-41870, November, 2020
https://doi.org/10.37118/ijdr.20415.11.2020

RESEARCH ARTICLE OPEN ACCESS

COMBINATION OF SOCIAL MACHINES AND SERVICE-ORIENTED RELATIONSHIPS

***Brunno Wagner Lemos de Souza and Silvio Romero de Lemos Meira**

Centro de Informática, Universidade Federal de Pernambuco – UFPE, Av. Jornalista Anibal Fernandes, s/n, Cidade Universitária (Campus Recife), CEP: 50.740-560 - Recife – PE, Brazil

ARTICLE INFO

Article History:
Received 14th August, 2020
Received in revised form
21st September, 2020
Accepted 20th October, 2020
Published online 24th November, 2020

KeyWords:
Social Machines; Relationship; Services; Web.

****Corresponding author:***
Brunno Wagner Lemos de Souza

ABSTRACT

The dissemination of the Web as a software development platform has led to a combination of computational and social elements to deal with the complexity of existing services and operations on the network, known as Social Machines. Social Machines establish relationships between people and machines through different views of relationships and their constraints. The role of Social Machines has changed the understanding of the nature of computing in complex operations and services, increasing the interactivity and connectivity of applications, collaborative platforms, and social networks. In this research, possibilities of interactions between Social Machines from the perspective of service-oriented relationships are presented, using the method based on Design Science Research (DSR). Given the type of relationship established in this research, some existing characteristics of architectural styles that can be used for communication between machines are highlighted, and the results are expected to bring, in a certain way, a set of perspectives of the service-oriented relationship between social machines.

Citation: ***Brunno Wagner Lemos de Souza and Silvio Romero de Lemos Meira, 2020.*** "Combination of social machines and service-oriented relationships", *International Journal of Development Research*, 10, (11), 41864-41870.

INTRODUCTION

The internet has enabled the growth of systems that use computer concepts and are also guided by social processes leading to the argumentation of Social Machines (SMs) and social computing, a philosophical conception in which people and machines interact to solve problems. As a result, new applications are emerging rapidly, and new computational models and paradigms are needed to deal with them. This paradigm is called Social Machine. The Social Machine represents the insertion of the combination of computational and human elements in society. According to Hendler and Berners-Lee (2010), one of the first definitions is that the Social Machine is a computational entity that combines computational and social processes. However, some people think that the Social Machine is an artificial machine, whichdoes not correspond to reality. In this sense, internettechnology, in particular, involves interactions betweenhumans and computers through complex informationnetworks. Taking this thought to a logical conclusion, trust in such machines will be more significant when the user participates in the design and operation, that is, the more relationship exists, better (BURÉGIO; MEIRA; ROSA, 2013).

We are in times intensely experimental with Web 3.0, that is, the Web as a programming platform in which the construction of Social Machines will be of revolutionary impact, which will reduce the level of complexity of networked systems, also known as sociotechnical systems, and that is already around us, creating their specific applications and providing innovations in the way of interacting, communicating and articulating on the Web (MEIRA *et al.,* 2011). As a result, it is understood that it is feasible and necessary to describe a Social Machine using existing Software Engineering models and instruments because Social Machines are concepts of a model representing entities connected to the Web (MEIRA *et al.,* 2011).

Currently, it is known that collaborative platforms such as Wikipedia, Salesforce, Social Networks in general (LinkedIn, Twitter, Facebook, ...), applications (Uber, Waze, ...) are characterized as Social Machines since they constitute Information Systems (IS) relating human and computational elements, and numerous factors must be taken into account when developing mechanisms of access to Social Machines, as there will be levels of complexity, limitations, and restrictions in the relationship.

One of the objectives of Social Machines, whether in the governmental sphere: health, education, and safety, is the heterogeneous combination of man and machine to solve problems for society. Thus, it was observed that there is a great need for further studies that deepen the theme of Social Machines with possible types of relationships. Therefore, it is intended to answer the following research question: how do we combine service-oriented relationships with social machines? Thus, this research aims to present a combination of Social Machines with one of the existing relationship types, in this case, service-oriented relationships.

The relevance of this research's contribution because it is an emerging area of study is to present a combination of Social Machines, mainly in its construction with the existence of service-oriented relationships, producing different points of view about architectural styles.

MATERIALS AND METHODS

For the continuity of this study's development, a theoretical-conceptual methodological approach based on a broad literature review is used. From the literature review, it was possible to verify that the concepts of the methodological proposal associated with Design Science research are relevant and applicable to Computer Science, specifically, to the area of Software Engineering. There is a need to adopt a research method centered on the evolution of a "Project Science" (Design Science), meaning, and operationalization forms. Design Science Research (DSR) is a method that operationalizes research conducted under the Paradigm of Design Science. For Vaishnavi, Kuechler, and Petter (2009), DSR is a new perspective with a set of analytical techniques that allow the development of research in various areas, including engineering and computing. DSR aims to study, research, and investigate the production (artificial) and its behavior, both in the academic and organizational aspects (BAYAZIT, 2004). It can be affirmed that DSR is a research method directed at problem-solving (MARCH; STOREY, 2008). This method seeks, from the understanding of the problem, to construct and evaluate artifacts that allow transforming situations, changing their conditions, to better or desirable states (MARCH; STOREY, 2008). These artifacts produced or considered by DSR are classified into: constructs, models, methods, instantiations (MARCH; SMITH, 1995), and may also result in an improvement of theories (HEVNER; CHATTERJEE, 2010; VENABLE, 2006).

Finally, a fundamental characteristic of the research that uses DSR as a method is that it is oriented to problems seeking a satisfactory solution to a given situation instead of an optimal solution. One reason for applying the DSR in research is the possibility of reducing the existing space between theory and practice (AKEN, 2004; AKEN, 2005; ROMME, 2003). This is because it is a method oriented to problem-solving, but it can reference improving theories since it is a knowledge-producing method. Figure 1 presents the DSR its relationship with two fundamental factors for the research's success: rigor and relevance.

As shown in Figure 1, DSR should consider the relevance and accuracy of the research. After all, it is the professionals of the organizations who will make use of the results of investigations and the knowledge generated to solve their practical problems and with fundamental concern for the research to be valid and reliable, thus contributing to the increase of the existing knowledge base in a given area. Also, Figure 1 relates to the environment as the place where the problem is being observed, where the researcher's event of interest is located. This environment consists of people, organizations, and technologies (HEVNER *et al.,* 2004).

Based on the researcher's interests depending on organizational needs, DSR supports the development and construction of artifacts and the improvement of the existing theory. These artifacts will later be evaluated along with the justifications for their importance. The current knowledge base needs to be consulted and used to support this development, construction, explanation, and evaluation activities. This knowledge base consists of fundamentals and methods consolidated and recognized by the academy, according to Figure 2. These methods mainly support the justification and evaluation of constructed artifacts or improved theory (HEVNER *et al.,* 2004).

Evaluation Methods	
1. Observational	Case Study: Study artifact in depth in business environment
	Field Study: Monitor use of artifact in multiple projects
2. Analytical	Static Analysis: Examine structure of artifact for static qualities (e.g., complexity)
	Architecture Analysis: Study fit of artifact into technical IS architecture
	Optimization: Demonstrate inherent optimal properties of artifact or provide optimality bounds on artifact behavior
	Dynamic Analysis: Study artifact in use for dynamic qualities (e.g., performance)
3. Experimental	Controlled Experiment: Study artifact in controlled environment for qualities (e.g., usability)
	Simulation – Execute artifact with artificial data
4. Testing	Functional (Black Box) Testing: Execute artifact interfaces to discover failures and identify defects
	Structural (White Box) Testing: Perform coverage testing of some metric (e.g., execution paths) in the artifact implementation
5. Descriptive	Informed Argument: Use information from the knowledge base (e.g., relevant research) to build a convincing argument for the artifact's utility
	Scenarios: Construct detailed scenarios around the artifact to demonstrate its utility

Source: Hevner et al. (2004)

Figure 2. Usable evaluation methods in Design Science

To assist in the driving of DSR, Hevner et al. (2004) define seven criteria that should be taken into account by researchers, according to Figure 3. These criteria are fundamental since DSR seeks to create a new artifact (criterion 1) for a particular problem (criterion 2). Once this artifact is presented, its usefulness must be explained. For this, it must be adequately evaluated (criterion 3). Besides, the research contributions should be clear, both for professionals interested in the resolution of organizational problems and academia, to contribute to the evolution of knowledge in the area (criterion 4). To ensure the validity of the research and expose its reliability, investigations must be conducted with the proper rigor, demonstrating that the constructed artifact is appropriate to the proposed use and that it met the criteria established for its development (criterion 5). Besides, for the construction, or even evaluation of the artifact, the researcher must understand the problem and seek possible ways to solve it (criterion 6). Finally, the study results must be duly communicated to all interested parties (criterion 7) (HEVNER *et al.,* 2004).

The authors Hevner et al. (2004) systematized a set of seven guidelines that became references for researchers, reviewers, editors, and readers to understand and evaluate the design science research method in the field of IS. The term design science is chosen to highlight the orientation in knowledge for design (for real-world problem solving) and the tools needed

for appropriate actions, which are professionals' domain, characterizing a strong link between the design science approach with the area of Information Systems.

Design-Science Research Guidelines	
Guideline	**Description**
Guideline 1: Design as an Artifact	*Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.*
Guideline 2: Problem Relevance	*The objective of design-science research is to develop technology-based solutions to important and relevant business problems.*
Guideline 3: Design Evaluation	*The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.*
Guideline 4: Research Contributions	*Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.*
Guideline 5: Research Rigor	*Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.*
Guideline 6: Design as a Search Process	*The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.*
Guideline 7: Communication of Research	*Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.*

Source: based on Hevner et al. (2004)

Figure 3. Criteria for conducting research using Design Science Research

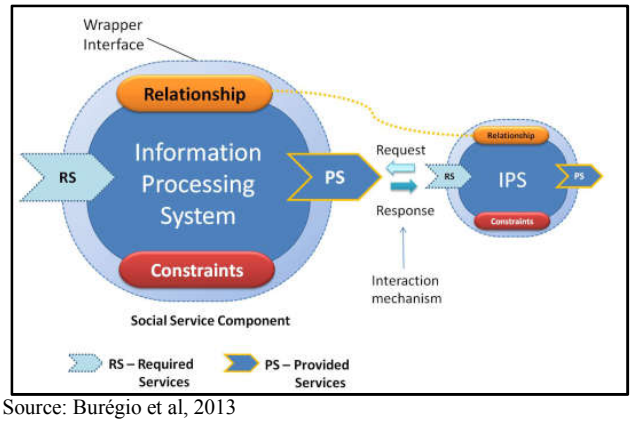
This research can also be described as qualitative, exploratory, and descriptive since it aims to generate a greater understanding of the research object, gathering information about the design science approach, describing the nature of such behavior and evolution, and how to conduct, evaluate and present research guided by this approach.

RESULTS AND DISCUSSION

A definition presented by Meira et al. (2011) says that a Social Machine is like a connectable entity that contains an internal processing unit and an interface that waits for requests and responds to other Social Machines (SMs). The Social Machines represent a connection system to deal with the complexity that the internet suggests since for Meira et al. (2011), the internet today is a programmable, open platform, where applications and services are increasingly used to transform industry and society.

The Social Machine as a component of social service, presented in Figure 4, was defined as: "a connectable and programmable building block that involves communication interface (wrapper interface), an information processing system and defines a set of services required and provided, dynamically available under restrictions, which are determined, among other things, by their relationships with others" (BURÉGIO *et al.,* 2013). Emphasizing that a component of social service is built on three main concepts: computing, communication, and control, and that it is fundamental to understand the role that each one plays in understanding Social Machines.

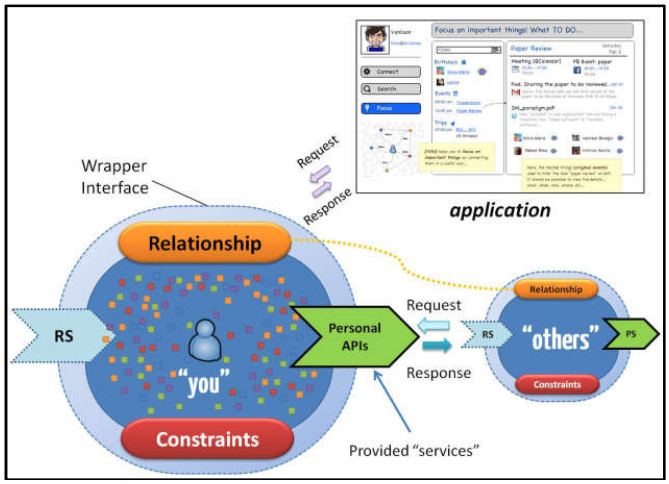
In practice, a relationship between two Social Machines can be obtained by establishing the ongoing relationship between them in advance.



Source: Burégio et al, 2013

Figure 4. Social Machine as a social service component

For example, having specific types of app interactions, such as Twitter, Facebook, Dropbox, Instagram, a client app needs to be registered before calling its rendered services, and in most cases, different restrictions are associated with those relationships to determine the specific interaction view. Other types of relationships can also be considered. However, regardless of styles, the main idea to highlight is the relationship as a key to determining the different sets of interaction views. The concept of relationships between Social Machines is similar to that of relationships between people; we can see them as relationships of trust between different Social Machines, satisfying the established restrictions, as shown in Figure 5 (BURÉGIO *et al.,* 2014).

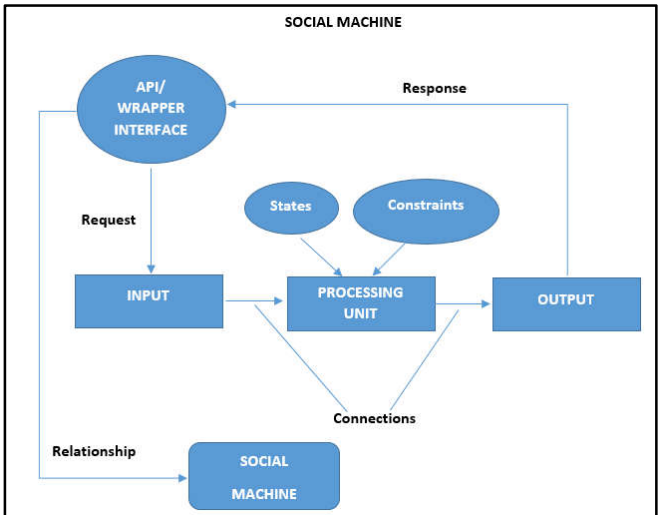


Source: Burégio *et al.,* 2014

Figure 5. People as "relationship" of Social Machines

The communication interface (Wrapper Interface) abstracts any communication layer through which a Social Machine externalizes its services to interact with other Social Machines. For example, considering Twitter as a Social Machine, the API provided can be regarded as a type of communication interface because, through the Twitter API, a client application can interact with its primary services (search, tweet, direct messages, retweet). This communication interface may also be responsible for oncoming the Social Machine's interaction visions according to the restrictions and existing relationships with other Social Machines (BURÉGIO *et al.,* 2014). Figure 6 represents the interaction between Social Machines through their communication interfaces through an existing relationship between them. One of the Social Machines is described in a detailed way internally, and the other is not, whose intention was to show the connection between the two.

Thus, the Social Machine is an information system related to other systems, containing significant and restrictive elements in the relationship.



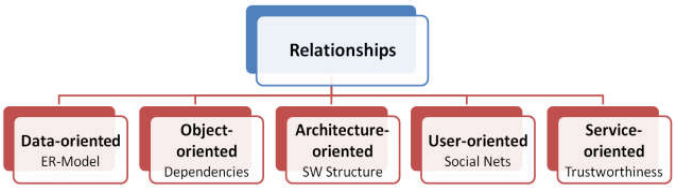
Source: (Lemos de Souza & Meira, 2020)

Figure 6. Representation of relationship between Social Machines in the System approach

According to Elmasri and Navathe (2005), these restrictions are established according to the modeled reality. That's when Meira et al. (2011) defined a Social Machine stating that it contains relationships, interfaces, requests, responses, state, constraints, input, processing, and data output. The concept of relationships to which Meira et al. (2011) refer is analogous to the relationship concept of Silberschatz, Korth, and Sudarshan (2011) and Elmasri and Navathe (2005); these last two authors use the word "entity" as "something" that relates. In the case of Meira et al. (2011), these relationships can occur with people to deal with the relationship between Social Machines, that is, a Social Machine that can communicate with another Social Machine, followed by any well-defined communication protocol. Interfaces are defined as a communication layer through which a Social Machine externalizes its services and allows interactions with other Social Machines existing on the internet. Requests are defined as a remote procedure call for services provided by the Social Machine interface. Responses are defined as a private response to other machines, also through the interface. In turn, the state is defined as the current situation of the Social Machine. Restrictions are defined as rules to be considered during the establishment of relationships between different Social Machines. Finally, Meira et al. (2011) state that every Social Machine should receive input data to perform processing and generate output data. The operation of a Social Machine is to receive requests (inputs) from other SMs, these are processed and return responses (outputs). Also, rules will define relationships with other SMs under a specific set of constraints.

The interaction between users and applications with a system can be described as a relationship defined as a dynamic set between databases and services. In Social Machines, these interactions are driven due to the large number of possible relationships established between users, applications, and Social Machines provided through APIs (SHADBOLT; KLEEK; BINNS, 2016). When it comes to relationships, it creates a connection with something (components), according to Dicio (2019), and that expresses the dependencies and requirements between them, that is, restrictions, and that can also be a connection between theory and practice. A relationship must contain its function, representation, rules, and exceptions of its establishment, its occurrence, and when it can cease to exist. For Silberschatz (2011), one relationship is an association between several entities. Elmasri and Navathe (2005) say that the relationship between two or more entities shows an association. Yet Elmasri and Navathe (2005) state that "the relationship types of relationships usually have certain restrictions that limit the combinations of entities that can participate in the corresponding set of relationships".

The relationship is the Web's centerpiece, which can be seen as a "dynamic set of relationships" between information collection and services (BURÉGIO *et al.,* 2013). The semantics of the relationship are now explicit, and, in addition to representing static connections and dependencies, it establishes constraints that are influential in the way Social Machines interact dynamically (BURÉGIO *et al.,* 2013). According to Burégio et al. (2013), the relationship can generally be defined as "the way things are connected" and, in this sense, is often used as an interchangeable term as "connection", "association", "link", "relationship". The authors also state that the relationship is an essential element in the Social Machine model. Thus, it was defined: "a relationship is a particular type of connection that restricts the way two or more Social Machines are associated or interact with each other" (BURÉGIO *et al.,* 2013). Figure 7 shows the different types of relationships addressed in the area of computing. They are data-driven relationship, object-oriented relationship, architecture-oriented relationship, user-oriented relationship, and service-oriented relationship. The latter will be detailed in this research because the Social Machine uses various services and applications that involve the internet.



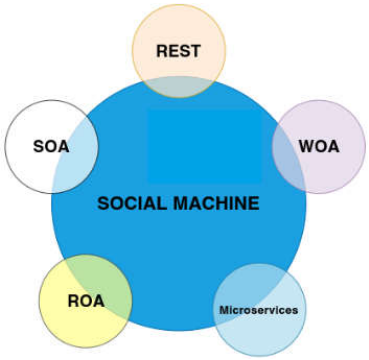
Source: Burégio et al. (2013)

Figure 7. Different relationship views

In the service-oriented relationship view concerning distributed systems that are service-oriented, the relationship underlies the reasoning about reliability, as shown in Figure 7.In these systems, trust relationships are used to infer access to reputation, control of services, and resources (Suryanarayana *et al.,* 2004). Control of services and resources (Figure 8 represents the association between two Social Machines through a service-oriented relationship.

Figure 8. Service-oriented relationship of Social Machines

Although there are different architectural styles, as shown in Figure 9, the question of trust in decentralized environments is explicit, and the architectural style is the combination of distinct characteristics in which architecture is executed or expressed (OPEN GROUP, 2006).



Source: adapted from Lemos de Souza & Meira (2020)

Figure 9. The Social Machine and service-oriented relationships

Given the type of relationship established, some existing characteristics of architectural styles are highlighted in this research.A service architecture follows in general to identify all the links between business and IT from the context of people, processes, and technology (OPEN GROUP, 2006). A service is a logical representation of repeatable business activity with a specified, self-sufficient result composed of other benefits. A software program runs a software service. It produces effects that have value for the people or organizations that are its consumers. It has a provider - a person or organization responsible for running the program to produce these effects. And there is an implicit or explicit service contract between the provider and consumers that the program will have the impact expected by consumers. The purpose of a service is to represent what the business does and put a limit on all parties, but predominantly in which the business can agree, and it is in the representation of the place that the creation of a service architecture should be focused because technology becomes a secondary element (KISTASAMY; MERWE VAN DER; DE LA HARPE, 2010).

The Service-Oriented Architecture (SOA) definition produced by the Open Group team states that it is an architectural style that supports service guidance. For Kistasamy, Van der Merwe, and De la harpe (2010), SOA is a business tool that allows companies to architect their processes independently, thus leading other architectural domains to follow the same logic. This is a way of thinking about services and service-based development and service outcomes (OPEN GROUP, 2006). These services are identified and defined in the form of a contract by business functionality modules or applications with exposed interfaces that allow the use of techniques such as service composition, message-based communication, and model-oriented implementation, which provide rapid development of effective and flexible solutions (OASIS, 2016). The real effect of SOA is the replacement of large, monolithic applications that have small interoperability interfaces, reluctantly provided and not guaranteed, by trimmer, modular services that have interface descriptions and contracts (OPEN GROUP, 2006).

Representational State Transfer (REST) is a Web-inspired architectural style for distributed hypermedia systems where existing Web principles and protocols are sufficient to create robust Web APIs services, not in need of SOAP protocol, although it takes the use of standards (HTTP, URI, XML/HTML, etc.) (CHEUNG, 2012). This style describes some architectural constraints that exemplify how features you can think of as a customer and have become famous and the frameworks that helped create web services called RESTFul (FIELDING, 2012). While REST is a set of architectural guidelines applicable to various computing infrastructures, Resource-Oriented Architecture (ROA) is web-bound only. Therefore, this architecture is primarily useful for companies that consider the Web as their preferred computing/publishing platform (FIELDING, 2012).ROA is a software architecture style and programming paradigm for designing and developing support software in features with RESTful interfaces. There is a lot of overlap with service-oriented architecture (decentralization and small interoperability services), but that means that instead of treating functionality and data as service calls, it's like restful resources (ONTHEROA, 2009). These features are software components (discrete pieces of code or data structures) that can be reused for different purposes since ROA supports the interconnection of elements whose principles and guidelines are used in software development and systems integration (FIELDING, 2012).

The Web-oriented architecture (WOA) is a type of software architecture designed to support to be used in websites, Web-based software services, and applications, and is simply a way to implement SOA by creating services, which are RESTful capabilities, allowing any service to be accessed with a URI (ROSENBERG, 2008). WOA is based on SOA, adding support for web-based software applications and services. The main difference between SOA and WOA is using REST APIs by WOA instead of SOAP by SOA (ROSENBERG, 2008). WOA = SOA + WWW + REST, being a non-ideal approach for all scenarios, but new applications, websites, APIs, and other services are more focused on the use of this technology with the help of simplified Web protocols such as REST and JSON (JavaScript Object Notation) (DONG; PAUL, PAUL, 2010 ZHANG, 2009; MARKS; BELL, 2006). These protocols are easier for Web developers and are more recognized because of large social platforms such as Facebook, Amazon, and Twitter, etc., which use them (MARKS; BELL, 2006).

In recent years, microservices architecture has emerged to describe a particular way to design software applications as independently deployable service suites, written in different programming languages, and use diverse data storage technologies (FOWLER; LEWIS, 2014). Microservice is an architectural style that uses an approach to developing a single application as a set of small standalone services, each running in its process and communicating with lightweight mechanisms, often an API of HTTP resources, and which can develop better ways to have machines talking to other machines (AECE, 2009; NEWMAN, 2015). The proposal of microservices-oriented systems architecture is to create more flexible, scalable, and more maintenance designs than the monolithic system architectures commonly used (MACHADO, 2017). This is one of the main reasons why organizations like Amazon and Netflix use these architectures to remove as many impediments as possible. The idea is to obtain smaller and smaller services because the service is shorter, maximizes the benefits around the interdependence and disadvantages of architecture, and vice versa.

CONCLUSION

It is already known that the Social Machine can be defined as a programmable structural set that involves an information processing system consisting of a group of requirements and

services provided, dynamically available under constraints that are determined by relationships. These restrictions can be specified as a set of rules between the Social Machines involved. Research on Social Machines is not yet mature since it is needed to standardize the concepts, consequently, a higher level of understanding and maturity. This represents that, in addition to the ideas not being aligned, it is much more challenging to perceive evolution in behavior and related perspectives and the construction of relationships through different views, which points out that there is a tendency for future studies on Social Machines.According to what was presented about the Social Machine characterization, aspects of the constituent elements were expanded and discussed so that social machines can be implemented and have restrictions through relationships.Given the fact already stated, because it is found that a Social Machine architecture can become functional, it is necessary to be composed of concrete architecture, in this case, an architecture that represents a service-oriented relationship. Knowing that a Social Machine is an information system that uses relationships and constraints, then, in this research, from the view of the types of relationship is and being service-oriented, we highlight the types of existing architectural styles and their characteristics.

ACKNOWLEDGMENT

The Coordination supported this work for the Improvement of Higher Education Personnel (CAPES) and the Informatics Center of the Federal University of Pernambuco (Cin-UFPE).

REFERENCES

Aece, I. Service Granularity. 2009. Available at: <http://www.linhadecodigo.com.br/artigo/2599/granularidade-de-tigoservicos.aspx>. and <https://israelaece.com/2009/10/29/granularity-de-servi-os/>.

Aken, J. E. VAN. Management Research Based on the Paradigm of the DesignSciences: The Quest for Field-Tested and Grounded Technological Rules. Journal of Management Studies, v. 41, n. 2, p. 219-246, 2004.

Aken, J.E. VAN. Management Research as a Design Science: Articulating the Research Products of Mode 2 Knowledge Production in Management. British Journal of Management, v. 16, n. 1, p. 19-36, mar 2005.

Bayazit, N. Investigating Design: A Review of Forty Years of Design Research. Massachusetts Institute of Technology: Design Issues, v. 20, n. 1, p. 16-29, 2004

Burégio V, Meira S, Rosa N. (2013). Social Machines: A Unified Paradigm to Describe Social Web-Oriented Systems. *International World Wide Web Conference Committee (IW3C2)*.

Burégio V, Nascimento L, Rosa N, Meira S. (2014). Personal APIs as an Enabler for Designing and Implementing People as Social Machines. *International World Wide Web Conference Committee (IW3C2)*.

Burégio, V.A; Meira, S.L; Rosa, N.S; and Garcia, V.C. (2013). "Moving towards 'relationship-aware' applications and services: A social machine-oriented approach," *IEEE International Enterprise Distributed Object Computing Conference Workshops Moving*, pp. 43–52.

Cheung Anson. Web APIs - Resource Oriented Architecture. 2012. Available at: <http://www.ansoncheunghk.info/article/web-apis-resource-oriented-architecture>.

Dicio. Relacionamento. Disponível em: https://www.dicio.com.br/relacionamento/.

Dong, Jing; Paul, Raymond & Zhang, Liang Jie (2009). "Chapter 12 : Specifying Enterprise Web-Oriented Architecture". *High Assurance Services Computing. Springer.* ISBN *038787657X.*

Elmasri, R.; Navathe, S.B. Database systems 4th edition. technical reviewer Luis Ricardo de Figueiredo. -- São Paulo : Pearson Addison Wesley, 2005.

Fielding, Roy T. Chapter 5 dissertation is "Representational State Transfer (REST)". 2012. Disponível em:<https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm> e em <https://www.revolvy.com/page/ Resource%252Doriented-architecture>.

Fowler, M.; Lewis J. Microservices a definition of this new architectural term. 2014. Disponível em: <https://martinfowler.com/articles/microservices.html>.

Hendler J, Berners-Lee T. (2010). From the Semantic Web to social machines:A research challenge for AI on the World Wide Web. *Artificial Intelligence*. https://www.sciencedirect.com/science/article/pii/S00047020900140.

Hevner, A. R.; Chatterjee, S. Design Research in Information Systems: Theory and Practice. New York: Springer, 2010.

Hevner, A. R.; March, S. T.; Park, J.; Ram, S. Design Science in information systems research. MIS Quaterly, v. 28, n. 1, p. 75-105, 2004.

Kistasamy, Christopher & Van der Merwe, Alta & De La Harpe, Andre. (2010). The Relationship between Service Oriented Architecture and Enterprise Architecture. 129 - 137. 10.1109/EDOCW.2010.12.

Lemos de Souza, B. &Meira, S. (2020). Social Micromachine: Origin and Characteristics. In Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 1, pages 788-796. ISBN: 978-989-758-423-7. DOI: 10.5220/0009580507880796.

Machado, M.G. (2017). Micro services: What is the difference for monolithic architecture. Available https://www.opus-software.com.br/microservicos-diferenca-arquietura-monoliticas/.

March, S. T.; Smith, G. F. Design and natural science research on information technology. Decision Support Systems, v. 15, p. 251-266, 1995.

March, S. T.; Storey, V. C. Design Science in the Information Systems Discipline: An Introduction to the Special Issue on Design Science Research. MIS Quaterly, v. 32, n. 4, p. 725-730, 2008.

Marks, Eric A. and Bell, M. Service-Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology: Wiley, 2006.

Meira S., Burégio V., Nascimento L., Figueiredo E., Neto M., Encarnação B., Garcia V. (2011). The Emerging Web of Social Machines, https://arxiv.org/ftp/arxiv/papers/1010/1010.3045.pdf

Newman, S. Building Microservices, First Edition (2015, Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Oasis. Reference Model for Service Oriented Architecture 1.0. 2006. Available at: <http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf>.

Ontheroa. On the ROA. Resource Oriented Architecture at the U.W (University of Washington. 2009. <https://blogs.uw.edu/ontheroa/2009/03/24/what-is-roa-resource-oriented-architecture/>.

Open Group. Service-Oriented Architecture. 2006. Available in: < http://www.opengroup.org/soa/source-book/soa/index.htm>.

Romme, A. G. L. Making a Difference : Organization as Design. Organization Science, v. 14, n. 5, p. 558-573, 2003.

Rosenberg, D. Web-oriented architecture and the rise of pragmatic SOA. 2008. Disponível em: <https://www.cnet.com/news/web-oriented-architecture-and-the-rise-of-pragmatic-soa/>.

Shadbolt, N.; Kleek, M.V.; and Binns, Reuben. The Rise of Social Machines. The development of a human/digital ecosystem, 2016.

Silberschatz, A.; Korth, H. F.; Sudarshan S. Database System Concepts, 6th Edition (2011, McGraw-Hill).

Suryanarayana, G; Erenkrantz, J. R; Hendrickson, S. A; and Taylor, R. N. "PACE: an architectural style for trust management in decentralized applications," Proceedings. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), pp. 221–230.

Vaishnavi, V.; Kuechler, W. and Petter S. Design Research in Information Systems, 2009. Disponível em: <http://desrist.org/design-research-in-information-systems>. Acessoem: 20 mar. 2020.

Venable, J. R. The Role of Theory and Theorising in Design Science Research. DESRIST, v. Feb. 24-25, p. 1-18, 2006.

****