Social Micromachine: Origin and Characteristics

Brunno Wagner Lemos de Souza and Sílvio Romero de Lemos Meira
Centro de Informática, Universidade Federal de Pernambuco, Recife, Pernambuco, Brazil

Keywords: Information Systems. Social Machines. Relationship-aware. Microservices. Social Micromachine.

Abstract: The incorporation of computing into society through personal devices has led to the discussion of Social Machines and social computing, that is, has guided the even greater existence of relationships between people and machines to solve problems. Social Machines represent information systems that establish connections through certain constraints to deal with the complexity of services and operations because with the spread of the web as a software development platform along with increased interactivity and application connectivity, the understanding of the nature of computing has been modified. Software architecture is a description of how a software system is organized whether the large or small scale and is currently highly interconnected because interactions, relationships, and their constraints are considered in software behavior, including the service granularity that is used to measure the depth of abstraction that has been applied to existing services. In this research, some specific definitions of Social Machines are presented, extending the focus of the different relationship visions and their restrictions. Using a methodology based on technical goals, the understanding of the relationship-aware is presented, adding it to the Social Machine and to term Social Micromachine, highlighting the Microservice architecture as one of the types of service-oriented relationship.

1 INTRODUCTION

The first definitions of Social Machines were made by Hendler and Berners-Lee (2009), suggesting that a Social Machine is a computational entity that combines computational and social processes.

Next came the definition of Meira, Buregio, Nascimento, Figueiredo, Encarnação e Garcia (2011), presenting a Social Machine as a pluggable entity that contains an internal processing unit and an interface that awaits requests and responds to other Social Machines. 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.

For Sommerville (2011), a System is a significant collection of interrelated components that work together to achieve some goals. It is organized to perform a specific method, procedure, or control when processing information. Automates or supports human activities through information processing.

According Laudon and Laudon (2018) states that an information system can be defined technically as a set of interrelated components that collect (or retrieve), process, store, and distribute information to support decision making, coordination, and control of an organization. In addition to providing such support, these systems also assist managers and workers in analyzing problems, viewing complex issues and creating new products.

About relationship, it establishes a connection with something (components), according to Dicio (2009), which expresses the dependencies and requirements between them, that is, constraints and that can also be a connection between theory and practice. In a relationship, it must contain its function, its representation, rules, and exceptions of its establishment, its occurrence, and when it may cease to exist.

The term Relationship-aware, however, is part of the Relationship Awareness Theory, which is based on the premise that the traits of behavior are consistent with what is gratifying in interpersonal relationships and with concepts or beliefs about how to interact with others to achieve these gratifications (Porter, 1976).

Software architecture is a description of how a software system is organized. These business systems are distributed over several computers, which can be owned and managed by different companies

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

(Sommerville, 2011). In this context, Social Machines can be used in conjunction with service architecture. The "Service granularity" is used to measure the depth of abstraction that has been applied in services. It can be divided into two parts: fine-grained and coarse-grained granularity, where the former determines that there are services with few operations, but these operations are divided by several services; while in the second, there are few services, but each will contain a much more significant portion of operations (Pinto, 2014).

2 THEORETICAL FOUNDATION

In this section, we mention a little of the context related to systems, information systems, Social Machines, relationships, Relationship-aware Theory, relationship types and their constraints.

2.1 System

According to O'Brien and Marakas (2010), the concept of the system is implicit in the field of information systems. In this way, an explanation of how the generic concepts of systems apply to enterprises and the components and activities of information systems helps to understand many other concepts in technology, applications, development, administration of information systems, social rights.

Also, according to O'Brien and Marakas (2010), System can be defined simply as a group of interrelated or interacting elements that form a unified whole. However, the generic concept of the system presented in Figure 1 provides a more appropriate framework for describing such systems.

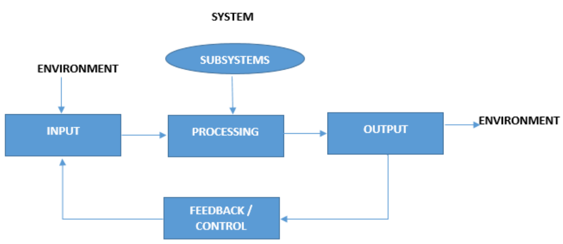

Figure 1: Generic Representation of a System.

The input stage involves the gathering and gathering of elements that enter the system to be processed. The processing step involves transformation processes (subsystems) that convert input (input) into output. The output stage involves the transfer of elements produced by a transformation process to its final destination. Finished products, human services, and management information should be passed on to their users. The Feedback / Control stage is data about the performance of a system; involves monitoring and evaluating the feedback to determine if a system is heading towards the achievement of its goal. Next, the control function makes the necessary adjustments to the input and processing components of a system to ensure that adequate production is achieved (O'Brien & Marakas, 2010).

In general, a Social Machine, Figure 2, represents a pluggable entity, containing an internal processing unit. It receives inputs, produces outputs and has states; and their connections intermittently or permanently define relationships with other SMs, these connections being established on top of a specific set of constraints and a communication interface, that is, a communication layer through which SM outsources its services and allows interactions with other SMs on the web. In this case, developed APIs can be considered a wrapper interface; these represent an instance of interactions (Meira et al., 2011).

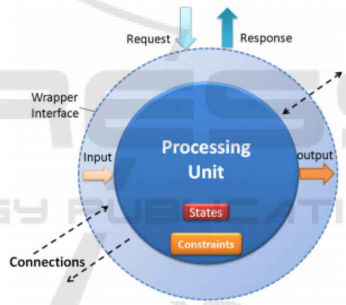

Figure 2: Representation of a Social Machine. Source: Meira, 2011.

2.2 Information System

According to O'Brien and Marakas (2010), the information system is an organized set of people, hardware, software, communication networks, and data resources that collect, transforms, and disseminates information in an organization.

To gain a complete understanding of information systems, it is necessary to understand their dimensions, which are: organization, management, and technology, according to Figure 3, since an information system creates value for the company, an organizational solution and to the challenges inserts by the environment (Laudon & Laudon, 2018).

Figure 3: Dimensions of an Information System. Source: Laudon & Laudon, 2018.

An organization coordinates the work through its hierarchy and business processes. The key elements of an organization are people, structure, business processes, politics, and culture. The role of management is to give meaning to the many situations faced by organizations, make decisions, and formulate action plans to solve problems. Technology is one of the tools that managers use to deal with change, whether these tools, platforms (hardware, software) used by companies (Laudon & Laudon, 2018).

Second Burégio, Meira and Rosa (2013) states that Social Machines represent a promising paradigm for dealing with the complexity that the Internet imposes around society, as well as being a practical way of explaining the connected entities that exist in it. In this way, in Figure 4, the authors Burégio et al. (2013) characterizes the Social Machines paradigm as the result of the convergence of three different visions: social software; people as computer units; and software as sociable entities.

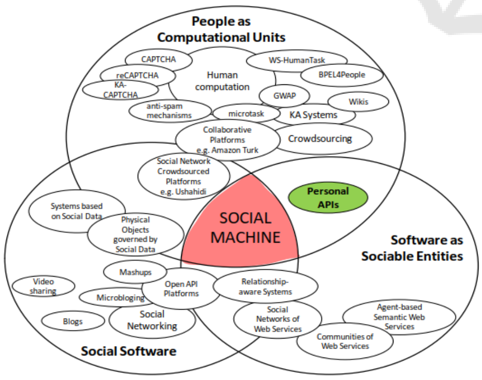

Figure 4: "Social Machines" as a result of the convergence of different views. Source: Burégio et al, 2013.

In the Social Software view, systems are changing the way software is developed (Burégio et al., 2013). This is because companies such as Twitter, Google, Facebook, Instagram, expose their functionality as services in the form of APIs. In the view of people as computing units, the idea is accepted that social computing is based on systems that use human skills in computing to solve problems that are trivial to human and complex to machines (Yuen, Chen and King, 2009). In the software view, as sociable entities, the socialization of software is allowed, mainly in terms of social relationships with other software (Burégio et al., 2013).

2.3 Relationship

Knowing that a Social Machine is an information system that is related through restrictions with other information systems, through its communication interface, in this subsection, the relationship question is discussed, once the systems have been identified, it must and then define how the relationship between them occurs. In general, relationships are named with verbs or expressions that represent how the entities interact or the action that one exerts on the other (Rodrigues, 2014).

For Silberschatz, Korth and Sudarshan (2011), a relationship is an association between several entities. Already Elmasri and Navathe (2005) says that the relationship between two or more entities shows an association between these. Still Elmasri and Navathe (2005) states that "types of relationships often have certain constraints that limit the combinations of entities that can participate in the corresponding set of relationships."

According to Elmasri and Navathe (2005), these restrictions are established according to the reality that is modeled. It was when Meira et al. (2011) defined a Social Machine stating that it contains relations, interfaces, requests, responses, state, constraints, input, processing, and output of data, shown in Figure 2.

The concept of relationships to which Meira et al. (2011) refers is analogous to the concept of the relationship of Silberschatz et al. (2011) and Elmasri and Navathe (2005); these last two authors use the word "entity" as "something" that relates. These relationships can occur with people, in the case of Meira et al. (2011), deals 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 layer of communication through which a Social Machine outsources its services and allows interactions with other Social Machines existing on the internet.

Already requests are defined as a remote procedure call for services provided by the Social Machine interface. Responses are defined as a remote response to other machines, also through the interface. In turn, the state is defined as the current situation of the Social Machine. Constraints are defined as rules to be considered when establishing relationships between different Social Machines.

Finally, Meira et al. (2011) states that every Social Machine must receive input data, perform processing, and generate data for output.

According to Burégio et al. (2013), the relationship, in general, can be defined as "the way things are connected" and, in this sense, is often used as an interchangeable term like "connection," "association," "link," "relationship." The same still affirms that the relationship is an essential element in the model of the Social Machine. For him, a Social Machine can connect with other Social Machines following any well-defined protocol. The concept of relations between SMs is similar to that of relationships between people. We can view them as reliable relationships between different SMs, satisfying the established constraints. In this way, 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 one another (Burégio et al., 2013)."

Therefore, we identified the need for the term Relationship-aware, whose idea was to identify the need for its understanding to connect better the relationship with the model and, consequently, to better satisfy the ownership of sociable software (Burégio et al., 2013).

Initially, the Theory of Relationship Awareness, by Porter (1976), analyzes how relationships are established and maintained so that one has a positive sense of self and of our value as a person; reinforces people's ability to choose behaviors that meet their underlying values while respecting the values of others. In this regard, it is a valuable tool to build trust, empathy, and productive and effective relationships through better communication.

Understanding the term Relationship-aware is a process that depends on the ability of participants in their activities to self-control and organizes with one another. It is fundamental how participants establish these perceptions within a context of social interaction. However, relationship-aware is not just a "feedback process." It consists of two greatnesses: recognition and receptivity. The first refers to how project structures anticipate the fact that participants need to establish or maintain their "social positions," and how that position is recognized and made available as a kind of resource to other participants. The second refers to the quality and speed of reaction to the participant's activity regarding other people and technologies (Burégio et al., 2013).

In this context of relationship-aware, systems that are aware of their relationship to other systems, have restrictive aspects that are considered. In Meira et al. (2011) and Brito, Abadie, Muniz, Marques, Buregio, Vinicius, and Meira (2012), the idea of the Social Machine is as a unifying mental model to understand, describe, and project each connected entity concerning web points as a fundamental element of that model. Turning software into web services means that it can interact with a host of other (and sometimes unknown) applications and services, and possibly establish a myriad of "social" relationships with them. In this sense, a system can be seen as a sociable entity, whose interactions are determined by its "social" relationships as people. Social Machines can interact in various ways, depending on the types of relationships that occur, in some cases, generating new Social Machines, competing or cooperating (Meira et al., 2011).

2.4 Types of Relationship

Knowing that a Social Machine, through the view of Meira et al. (2011) and Burégio et al. (2013), are systems that use relationships with constraints and, at the same time, Burégio et al. (2013) addresses a new type of relationship called Relationship-aware. The latter is not yet shown in Figure 5, which Burégio et al. (2013) exposes the different types of relationships, and in this research, the different constraints in each type of relationship will be signaled.

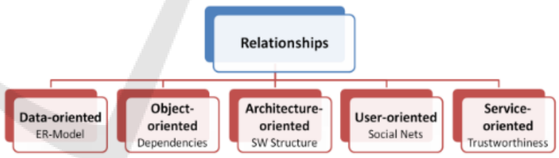

Figure 5: Different views of relationships. Source: Burégio et al, 2013.

In the data-oriented view, the relationship is a tuple of entities (entity-relationship model), and its existing constraints are allowed values (cardinalities) for specific attributes and the functional dependencies of entities in a relationship (Chen, 1976).

In the object-oriented view, relationships represent different dependency forces between object classes. Object orientation and UML (*Unified Modeling Language*) allow relationships to be much more done with classes than just simple statements, and you can apply constraints to class diagrams that describe how objects in a class can be used by restricting objects using the OCL language (Object Constraint Language), simply a language for specifying constraints on objects (Miles & Hamilton, 2006).

In the architecture-oriented view, relationships are introduced as a basis for constraints on the software structure and formal arrangements of its structural elements. Formal relationships can be used to define different topologies of network architectures, components, and associations of data elements. Most of the time, system constraints are the requirements (system description) by the principle of software engineering and software architecture. Each system is, in essence, a new architecture, a new architectural style. The constraints on architectural elements and styles can be used as constraints on an architecture (Perry & Wolf, 1992).

In the user-oriented view, the relationships correspond to connections between users. They form graphs of relationships between people, organizations, states, and other units. User-oriented design (UOD) - that transforms a technology package with the ability to deliver functionality into a "product" that people want to interact with and from which they derive benefits (Veryzer & Mozota, 2005); that is, it is the process that focuses on the needs and desires of users for the development of applications (services or products).

In the service-oriented view, concerning distributed and service-oriented systems, the relationship underlies the reasoning on reliability. In these systems, trust relationships are used to infer access to reputation and control of services and resources (Suryanarayana, Erenkrantz, Hendrickson & Taylor, 2004). Although there are different architectural styles, none of them is the issue of trust in decentralized environments explicit. An architectural style is the combination of distinct characteristics in which the architecture is executed or expressed (Open Group, 2006).

3 METHODOLOGICAL PROCEDURES

This research was done to acquire specific and structured knowledge regarding the concepts and understanding of Social Machines and the Social Micromachine with the Microservices architecture.

The present research is based on the following question: what relations between the social Micro-machine with Microservices architecture and the concept of relationship-aware?

This search for possible relations between the Social Micromachine with Microservices and the term relationship-aware has the ultimate objective of discovering possible effects that may exist of this interaction.

It was made a survey of available research relevant to a particular issue, in this case, are the relationship of Social Machines with the concept of relationship-aware and Microservices, and is related, in principle, to the area of interest in software engineering.

Initially, subjects related to the subject matter were investigated in the databases, and possible authors who have developed similar works were verified. Already in this first moment, a study of bibliographical revision was elaborated, obtaining theoretical references for the research in question.

With this, it is intended to present the understanding of the relationship-aware, adding to a architecture of Microservices and an architecture of a Social Machine, specifically, of a Social Micromachine.

The research was approached from the technical goals proposed below.

M1) Definition on Social Machines;

M2) Definition on Systems;

M3) Definition on Information Systems;

M4) Definition of relationships, their constraints and types of relationships, including relationship-aware;

M5) Definition of software architecture, especially service architecture, scalability and service granularity;

M6) Definition of Social Micromachine.

4 RESULTS

Comparing the generic system itself, presented by O'Brien and Marakas (2010) in Figure 1, and the Social Machines presented by Meira et al. (2011) in Figure 2, we have Figure 6, where the Social Machines interact through their interfaces of communication or a relationship between them.

The relationship is the centerpiece of the web, which can be seen as a "dynamic set of relations" between a collection of information and services (Fowler & Lewis, 2014). There is a need to create new mental models to describe and design this emerging sociable software and its relationship awareness capabilities.

The semantics of the relationship is now explicit and, in addition to representing static connections and dependencies, establishes constraints that are influential in the way Social Machines interact dynamically (Burégio et al., 2013).

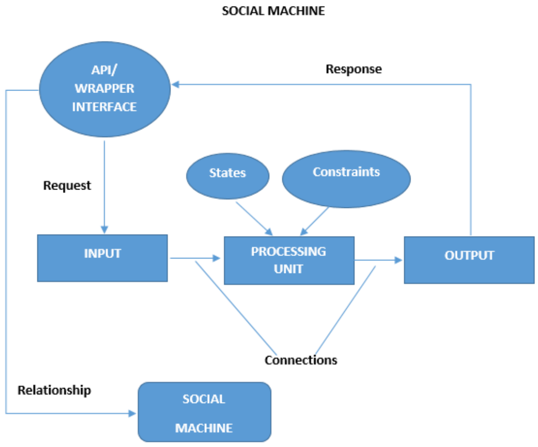

Figure 6: Representation of a Social Machine in the System Approach.

The relationship is the centerpiece of the web, which can be seen as a "dynamic set of relations" between a collection of information and services (Fowler & Lewis, 2014). There is a need to create new mental models to describe and design this emerging sociable software and its relationship awareness capabilities.

The semantics of the relationship is now explicit and, in addition to representing static connections and dependencies, establishes constraints that are influential in the way Social Machines interact dynamically (Burégio et al., 2013).

The central idea of software "relationship-aware" is through a simple analogy in the scope of social relations between people and relationships between software. This analogy is useful to show that almost all software interactions can be explained from the perspective of social relationships.

An easy way to put the same idea in the software context is to map the different sets of interactions of a person to the different views of interaction (services/features) that an application can do. Figure 7 illustrates a sociable application providing different views of interaction (V1 to Vn), whose properties, for example, set of services, rate of limitation, (performance) is determined according to the relationship between them and their client applications (Burégio et al., 2013).

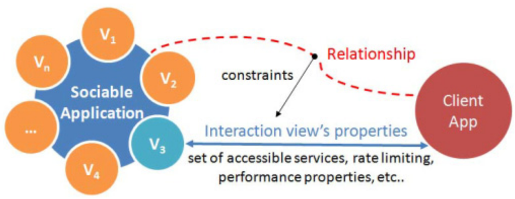

Figure 7: *Relationship-aware* application: relationship determining interaction views. Source: Burégio et al, 2013.

It is important to note that, in this context, two different visions of interaction mean two sets of different services or the same set of services provided under different constraints.

In practice, different types of “relationship-aware” software have been written, as the spread of the web as a software development platform coupled with increased interactivity and connectivity of applications and services has changed the understanding of the nature of computing in such a way that many computational processes, nowadays, based on the web, are autonomous and concurrent, including the architecture of software is highly interconnected in a way that are considered interactions, relationships and their restrictions in the behavior of the software.

Thus the notion of Social Machines is extended from its initial model, as well as its main elements to create a common abstraction that has the potential to describe any existing application or service through a unifying building block that makes use of computing concepts (in the form of information processing system), communication (relations and interactions) and control (restrictions), in order to possibly specify interaction services.

In comparing the dimensions of an information system presented by Laudon and Laudon (2018) and the converging visions, in Social Machines presented by Burégio et al. (2013), it is verified that social software is equivalent to technology, people as computational units are equivalent to management and software, as sociable entities, is equivalent to an organization. The result of the convergence of Laudon and Laudon (2018) are information systems, while the convergence result of Burégio et al. (2013) is the Social Machine. In this way, the Social Machine is an information system that relates to other systems, containing significant and restrictive elements in the relationship.

The Social Machine as a component of social service from the point of view of relationship-aware has been defined as: "a connectable and programmable building block that involves an communication interface (wrapper interface), an information processing system and defines a set of services required and provided, dynamically available under constraints, which are determined, among other things, by their relationships with others. "From this definition, a Social Machine can be seen as a Social Service Component, that is, a block (component) software architecture that provides a set of services that may vary according to their "social" relationships with others. building blocks interact to compose new systems, as shown in Figure 8 (Burégio et al., 2013).

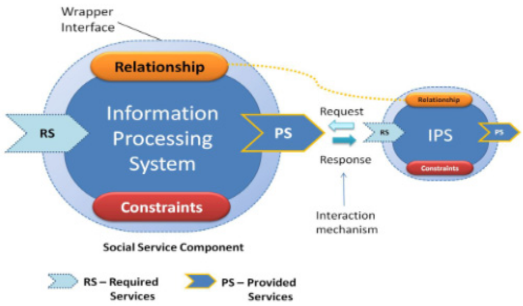

Figure 8: Social Machine as a Social Service ComponentSource: Burégio et al, 2013

Emphasizing that a social service component is built on three main concepts: computing, communication, and control, it is crucial to understand the role that each plays to understand Social Machines.

In Figure 8, discussed by Burégio et al. (2013), and in Figure 2, addressed by Meira et al. (2011), it is shown that the information processing system is equivalent to the processing unit and the state of the machine, as well as the existing services (required and provided) are equivalent to the input and output elements, respectively. However, the great difference of this new definition of the Social Machine is in the form of communication, which is based on constraints on services, and that these are determined by their relationships.

In practice, a relationship between two Social Machines can be obtained by the prior establishment of the persistent relationship between them. Other types of relationships can also be considered. However, regardless of types, the main idea to be highlighted here is the notion of the relationship as the 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 relations of trust between different Social Machines, satisfying the established restrictions (Burégio et al., 2013).

The Wrapper Interface abstracts any layer of communication through which a Social Machine outsources its services to allow interactions with other Social Machines. This communication interface can also be responsible for composing the views of interaction of the Social Machine according to the constraints and relations existing with other Social Machines (Burégio et al., 2013). In this way, the emergence of a Social Micromachine is attractive, from the point of view that a Social Machine can be seen as a Social Service Component, that is, a Social Micromachine being a block software (component) architecture that provides a set of specific services that may vary according to their "social" relationships through their constraints. Thus, the Social Machine would be a large-scale software architecture that uses fine granularity in services, while the social micro-machine would be a small-scale software architecture that uses gross granularity in services. The lower the service, the more the benefits are maximized, and the disadvantages over the architecture are minimized, that is, as services decrease, the benefits around interdependence increase. In dealing with this complexity, one can strive for ever smaller services. With smaller services, one can only scale the services that need to scale, which allows executing other parts of the system in smaller and less active hardware, where each service runs on several separate machines, that is, using features of Microservices.

Figure 9 represents the Social Micromachine as an equivalent system of the Social Machine since it is seen as a component of social service. The Social Micromachine is represented by the specific services that may occur. These can be complex, but the level of service granularity is low, and the service is more specific, having all the constituent elements (relationship, service architecture, constraints, computational unit, social entities, social software) of a Social Machine that works with complex and less specific services, containing the level of service granularity high.

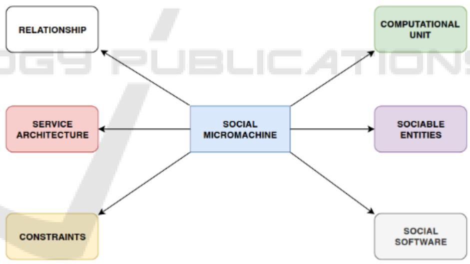

Figure 9: The Social Micromachine.

A service architecture serves, in general terms, to identify all the connections between business and IT from a context of people, processes, and technology (Open Group, 2006). Still Open Group (2006) states that service is a logical representation of a repeatable business activity that has a specified, self-sufficient result that can be composed of other services, thus being a "black box" for service consumers.

The purpose of a service is to represent what the business does and put a limit on all but predominantly where the business can agree, and it is in the business representation that the creation of a service architecture should be focused because, in this context, technology becomes a secondary element (Kistasamy, Van der Merwe, & De La Harpe, 2010).

Figure 10 represents the Social Micromachine as a subset of the Social Machine that relates to the types of service architectures described in this research.

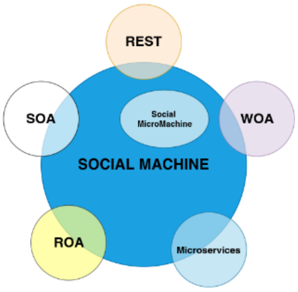

Figure 10: The Social Micromachine as subset of the Social Machine.

Taking into account all the characteristics of the service architectures mentioned, the Microservices describe a particular way of designing the implementation of independent services, which are more flexible, scalable and maintenance simpler, since it uses an approach for the development of a single application as a set of small services, each running in its process and communicating with lightweight mechanisms, often an HTTP resource API. They are small, autonomous services that work together and use the same approach for independent services and that can develop better ways of having machines talking to other machines.

In view of the above, referring to Figure 5, which discusses the different types of vision in relation to the relationship, we have in Figure 11, its magnification, linking the term relationship-aware to the Social Machine and the Social Micromachine, inheriting characteristics of the Social Machine itself, as well as Microservices, inheriting features of the services architecture and the set of relationships with their respective constraints.

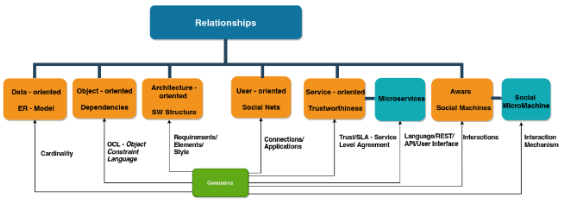

Figure 11: Expansion of the different views of relationships and their constraints.

5 FINAL CONSIDERATIONS

According to what was presented on the characterization of the Social Machine, aspects about the constituent elements were expanded and discussed so that Social Machines could be implemented and that they had restrictions through the types of relationship. Knowing that a Social Machine is an information system that uses relationships and constraints, then, from the view of the types of relationships, this view was broadened according to existing relationships, including relationship-aware, being presented the restrictions in each relationship and the Social Micromachine, which will use specific services, inheriting characteristics of the Social Machine itself, in the same way that Microservices inherit characteristics of the service architecture.

ACKNOWLEDGEMENTS

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

REFERENCES

Brito, K.; Abadie, L.; Muniz, P.; Marques, L.; Buregio, V.; Garcia, V. and Meira, S. “Implementing Web Applications as Social Machines Composition : a Case Study,” The 24th International Conference on Software Engineering and Knowledge Engineering, vol. (SEKE'2012), pp. 311–314, 2012.

Burégio, V.; Meira, S. and Rosa, N. Social Machines: a unified paradigm to describe social web-oriented systems, 2013.

Burégio, V; Meira, S; Rosa, N and Garcia, V. "Moving towards "*Relationship-aware*" Applications and Services: A Social Machine-Oriented Approach," *2013 17th IEEE International Enterprise Distributed Object Computing Conference Workshops*, Vancouver, BC, 2013, pp. 43-52.

Chen, P. “The entity-relationship model---toward a unified view of data,” ACM Transactions on Database Systems, vol. 1, no. 1, pp. 9–36, Mar. 1976.

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

Elmasri, R.; Navathe, S. Sistemas de banco de dados 4º edição. revisor técnico Luis Ricardo de Figueiredo. -- São Paulo : Pearson Addison Wesley, 2005.

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

Hendler, J. and Berners-Lee, T. From the Semantic Web to Social Machines: A research challenge for AI on the World Wide Web, 2009.

Laudon, K. and Laudon, J. Management Information Systems, 15 Ed., 2018, Pearson Education, Inc., New York, NY.

Kistasamy, C.; Alta, V. and Harpe, A. (2010). The Relationship between Service Oriented Architecture and Enterprise Architecture. 129 - 137.10.1109/ EDOCW.2010.12.

Meira, S.; Buregio, V.; Nascimento, L.; Figueiredo, E.; Neto, M.; Encarnação, B. and Garcia, V. “The Emerging Web of Social Machines”, 2011.

Miles, R. and Hamilton, K. Learning UML 2.0. O'Reilly Media, 2006, p.290

O'Brien J.A. and Marakas, G.M. Introduction to Information Systems, Fifteenth ed, McGraw-Hill_Irwin (2010).

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

Perry, D. and Wolf, A. “Foundations for the study of software architecture,” ACM SIGSOFT Software Engineering Notes, vol. 17, no. 4, pp. 40–52, Oct. 1992.

Pinto, R. O que é granularidade de serviços? Disponível em: <https://pt.stackoverflow.com/questions/19050/o-que-%C3%A9-granularidade-de-servi%C3%A7os>.

Porter, E. On the Development of Relationship Awareness Theory. 1976. Disponível em: <http://www.peakperformancepm.com.au/uploads/32766/ufiles/Relationship_ Theory_Overview.pdf>.

Rodrigues, J. Modelo Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER). 2014. Disponível em:<https://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>.

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

Sommerville, I. Engenharia de Software / Ian Sommerville ; tradução Ivan Bosnic e Kalinka G. de O. Gonçalves; revisão técnica Kechi Hirama. — 9. ed. — São Paulo: Pearson Prentice Hall, 2011.

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

Veryzer, R. and Mozota, B. The Impact of User□Oriented Design on New Product Development: An Examination of Fundamental Relationships, 2005.

Yuen, M; Chen, L. and King, I. “A Survey of Human Computation Systems,” 2009 International Conference on Computational Science and Engineering, pp. 723–728, 2009.