

The era of Social Machines in the BPM-SOA and UML-SOA convergence

A era das Máquinas Sociais na convergência BPM-SOA e UML-SOA

La era de las Máquinas Sociales en la convergencia BPM-SOA y UML-SOA

Received: 12/28/2021 | Reviewed: 01/03/2022 | Accept: 01/18/2022 | Published: 01/19/2022

Brunno Wagner Lemos de Souza

ORCID: <https://orcid.org/0000-0002-9071-1863>

Universidade Federal de Pernambuco, Brasil

E-mail: bwls@cin.ufpe.br

Abstract

In this work, we have explored and analyzed fundamental concepts of BPM (Business Process Modeling) methodology, as well as software development utilizing UML (Unified Modeling Language), both of which are based on SOA (Service-Oriented Architecture) architecture, as an objective of to integrate the areas of development of software and management of companies as the advent of Social Machines, to enable a more excellent combination of computer and human elements us processes. This study aims to present a possibility of a verified scenario from a survey of the literature that deals with the integration of BPM-SOA and UML-SOA with Social Machines. Thus a comparison was made between the execution languages of both models and the Social Machines to present a possibility of integration between architectures and models. All this ensures interoperability between different entities, adding value and reducing maintenance costs for companies. With the scientific methods that determine the discovery of the problem, observation, pois, constitutes an investigative procedure to understand the possibility of integration and implementation of BPM-SOA and UML-SOA with Social Machines.

Keywords: Social machines; BPM; UML; SOA; Integration.

Resumo

Neste trabalho, foram explorados e analisados conceitos fundamentais da metodologia de BPM (*Business Process Modeling*), bem como o desenvolvimento de software por meio da UML (*Unified Modeling Language*), ambas atreladas à arquitetura SOA (*Service-Oriented Architecture*), com o objetivo de integrar as áreas de desenvolvimento de software e de gestão de empresas com o advento das Máquinas Sociais, a fim de possibilitar uma maior combinação de elementos computacionais e humanos nos processos. O objetivo deste estudo é apresentar uma possibilidade de um cenário verificado a partir de um levantamento da literatura que trata da integração BPM-SOA e UML-SOA com as Máquinas Sociais. Dessa maneira, foi realizada uma comparação entre as linguagens de execução de ambas modelagens juntamente com as Máquinas Sociais, a fim de apresentar uma possibilidade de integração entre arquiteturas e modelagens. Tudo isso para assegurar a interoperabilidade entre diferentes entidades, agregando valor e reduzindo custos de manutenção para as empresas. Com relação aos métodos científicos que determinam o descobrimento do problema, utilizou-se o método da observação, pois, constitui-se um procedimento investigativo. A finalidade é compreender a possibilidade de integração e implementação de BPM-SOA e UML-SOA com as Máquinas Sociais.

Palavras-chave: Máquinas sociais; BPM; UML; SOA; Integração.

Resumen

En este trabajo se exploraron y analizaron conceptos fundamentales de la metodología BPM (Business Process Modeling), así como el desarrollo de software a través del UML (Unified Modeling Language), ambos vinculados a la SOA (Arquitectura Orientada a Servicios), con el objetivo integrar las áreas de desarrollo de software y gestión empresarial con el advenimiento de las Máquinas Sociales, a fin de posibilitar una mayor combinación de elementos computacionales y humanos en los procesos. El objetivo de este estudio es presentar la posibilidad de un escenario verificado a partir de una encuesta de literatura que trata sobre la integración de BPM-SOA y UML-SOA con Social Machines. De esta forma, se realizó una comparación entre los lenguajes de ejecución de ambos modelos junto con las Máquinas Sociales, con el fin de presentar una posibilidad de integración entre arquitecturas y modelos. Todo ello para asegurar la interoperabilidad entre diferentes entidades, agregando valor y reduciendo los costos de mantenimiento para las empresas. En cuanto a los métodos científicos que determinan el descubrimiento del problema, se utilizó el método de observación, ya que constituye un procedimiento de investigación. El propósito es comprender la posibilidad de integrar e implementar BPM-SOA y UML-SOA con Social Machines.

Palabras clave: Máquinas sociales; BPM; UML; SOA; Integración.

1. Introduction

There are many notations and tools for modeling business processes due to the significant variation in companies' needs. Depending on the goals and resource allocations, the modeler chooses the appropriate notation and tool for each objective.

Thus, the unified modeling language – Unified Modeling Language/UML (Booch et al., 2000), as it is an object-oriented notation for detailing software projects, but which allows the adaptation of diagrams for other purposes, including for modeling business processes, has flexibility that gives it an advantage over other notations.

In a more modern perspective, UML has some disadvantages compared to other notations such as BPMN – Business Process Modeling Notation (Baldam et al., 2008), for example, which is a notation explicitly created for modeling business processes, whose most significant advantage is perhaps its simplicity, which favors understanding, not only of IT specialists but also of laypeople and interested parties in the business. In this aspect, BPMN has an advantage over UML, as it allows the company's processes to be modeled by information flow and not by system.

Service-Oriented Architecture (SOA) adopts existing techniques and notations such as BPMN, UML, BPEL (Business Process Execution Language), WSDL (Web Service Description Language), and WS-BPEL (Web Services Business Process Execution Language). SOA is a paradigm for organizing and using distributed capabilities in different domains and provides a uniform way to offer discovery, interaction, and abilities based on a Web environment. It is interesting for its reusability principles, allowing that the same service is used in other applications, with loose coupling between its components, in addition, it speeds up the development since parts can be exchanged without significant interference in the general functioning of the system, also allowing the use of components already developed by the independent company of languages of schedule. When using SOA in conjunction with agile and traditional methodologies, in addition to available BPM case tools such as BPMN, BPEL, BPMS (Business Process Management System), EAI (Enterprise Application Integration), Workflow, process modeler, and App Servers, it is possible to build a solution that is fast to develop, that has excellent flexibility for maintenance, in addition to having its base in a Web environment.

From this perspective, BPMN facilitates application development practice by creating process models that are easy to understand, both for analysts and developers. The execution of these processes will be done using the BPEL language, which is also based on XML (eXtensible Markup Language) and describes the business process flow.

The BPEL language aims to define business processes using a language based on the interpretation of XML-type files. These processes consist of external entities that interact, such as Web Services. In this way, BPMS, an integrated system that allows the operationalization of strategy flows in a controlled and monitorable environment, enables integration to the solution with SOA, using the orchestration of web services provided by BPEL. A Web Service is an abstract notation that must be implemented by an agent (artifact) that performs communication between different software applications, acting as an interface element in the exchange of messages. This exchange will be done through a file in WSDL format, which is an XML-based language, or through a SOAP protocol (Simple Object Access Protocol), which supports the functionalities required by the Web Services.

When combined, BPM and SOA can enable companies to automate and optimize value chains through adaptable business processes by facilitating IT (Information Technology) capacity to develop and manage flexible systems and applications that integrate complex and heterogeneous technologies. BPM and SOA are, however, two different initiatives. BPM is primarily a management and strategy discipline that supports the idea that we can model a business with its end-to-end processes. On the other hand, SOA is an architectural approach to systems development that develops and delivers reusable business services and separates the system into parts so that different applications can share these services in a flexible and highly interoperable way.

According to Lemos (2009), this shows the ability of a system to communicate transparently with another system, through programs of various functional units, through common languages and protocols.

For a system to be considered interoperable, it must work with open standards or ontologies, which are data models that represent a set of concepts within a domain and the relationships between them.

Regarding the UML, Amsden (2010) states that its need and potential for integration were soon observed, however, some requirements still needed to be fulfilled in the case of service-oriented architecture, as the resources available were not enough to meet the needs of the UML to the demands of companies. As a result, the SoaML (Service-oriented architecture Modeling Language) approach was created, however, this is not the only possible approach. There were other smaller initiatives, however, the results of these were not as broad. Therefore, the interest in SoaML is understandable, considering how widespread and well-known the UML is from today's perspective. In this case, the claim is how they integrate with SOA.

Due to the evolution of the internet, communication mechanisms emerged offering services called Social Machines, which are simply information systems that combine human and computational elements that relate to each other (Lemos de Souza & Meira, 2020).

There is a great need for companies to obtain answers regarding the integration of architecture from developing their processes.

Business processes define how an organization can proceed to achieve its operational goals. In this context, a process driven by SOA is explicitly built to support the definition, execution, monitoring (intra-organizational), and business processes (inter-organizational). The use of service-oriented technologies has also led to the complete integration of security functionality in service-oriented systems processes.

The evolution of process modeling maintains a parallel with object-oriented development and with some contact with SOA. However, this relationship with SOA reveals itself as a path to the future. Thus, one of the challenges is BPM-SOA convergence, at least from a standardization point of view. Although BPMN (BPM notation standard) and BPEL (SOA notation standard) were not originally designed to work together, there is still a tendency to adopt process modeling through process architecture, as it should bring the right level of service granularity that best fits the need for enterprise-grade components and, consequently, the means to design converged solutions that are scalable and adaptable across the enterprise.

2. Methodology

This section presents the methodology and procedures adopted to carry out the bibliographic research, which was used to ensure part of the theoretical basis.

Research is an intuitive, systematic, controlled, and critical process that leads to discovering new facts and realizing the relationships established between the laws that determine these facts' emergence or absence (Prestes, 2011). Still, according to Prestes (2011), scientific knowledge results from a methodical, systematic investigation of reality, transcending the facts and phenomena in themselves and analyzing them to discover their causes and reach the conclusion of general laws. Such knowledge is verified, in practice, through demonstration or experimentation.

This research was carried out to acquire specific and structured knowledge about the integration between BPM-SOA and UML-SOA, resulting from the observation of facts related to these integrations and the registration of indicators presumably relevant for the joint integration with the Social Machines.

This research was structured according to its nature, approach, objectives, form of study, and object. Its core is fundamental and methodological. Its main aim is to generate new knowledge through bibliographic research that is useful for the advancement of science, without foreseen practical application, and deals with the ways of doing science. As for the approach, it is qualitative, as it considers a dynamic relationship between the natural world and the subject, portrayed through documents, taking into account relevant aspects, such as the opinions and comments of the authors. As for its objective, it is exploratory research, since the main aim is the improvement of ideas or discover intuitions, providing more information about the subject

that will be investigated, learning a new possibility of an approach to the topic. The object is bibliographic research, as it tries to solve a problem or acquire knowledge from information from existing materials, assimilating the concepts and exploring the aspects already published, trying to explain the understanding of the existing phenomena and how to create new propositions.

Regarding the scientific methods that determine the discovery of the problem, the observation method was used, as it constitutes an essential investigative procedure. The observation is systematic, structured, and carried out according to previously defined objectives and purposes.

3. Background

The integration process between BPM-SOA and UML-SOA has been adopted for a long time, but not with the emerging theme of Social Machines. This section covers UML, BPM, SOA, and Social Machines.

3.1 Unified Modeling Language

The UML can be defined as a graphical language for visualizing, specifying, building, and documenting complex software system artifacts, according to Booch et al. (2000). Thus, it is not a development method, having a use for the object-oriented programming paradigm. It does not tell us how to make a system, nor what to do first or after. However, through its diagrams, it is possible to represent software systems from different visualization perspectives. This language is composed of many model elements that represent a specific part of the system or a point of view through the creation of diagrams, providing a standardized way to prepare architectural plans of system projects such as business processes and system functions. However, the UML is not restricted to software modeling but is expressive enough to model non-software systems, such as workflow, systems structure and behavior, and hardware design, as stated Booch et al. (2000). In addition, the same authors also state that the UML provides a standard form for the preparation of architectural plans for system projects, with the inclusion of conceptual aspects, among which we can mention the business processes and system functions, in addition to items such as classes written in a particular programming language, database schemas, and reusable software components. Some authors, such as Wilcox and Gurau (2003), defend using the UML for process modeling, highlighting advantages such as simplicity in the notations, high standardization found in published applications; high applicability in real processes; flexible notation to different situations. When designing something, it is convenient to use models that represent what will be developed. These models thus constitute an abstract representation of a reality projected into the future. In this way, the UML offers a standardized way of creating these models, simplifying the complex software design process by using a graphical component and using a limited set of symbols. As a visual language for object-oriented modeling systems, the UML is a modeling language made up of graphic elements that represent the object-oriented paradigm's concepts. Through the graphic elements defined in this language, diagrams illustrating different system perspectives can be built. Therefore, the UML represents an effort to make object-oriented modeling a more straightforward process by unifying the most used methods as one of the best practices. The UML has coherently unified several ideas arising from various concepts, techniques, or theories that such language introduces.

3.2 Business Process Modeling

Processes are the structures by which an organization accomplishes its mission to produce value for its customers. A Business Process is a basis for studying Information Systems. From this perspective, they are previously established activities, which aim to determine how the work will be carried out in a specific organization. In addition, processes have well-defined beginning and endpoints. They must work in line with each other and relate to the entire organizational structure because only in this way will it be possible to achieve the objectives that produce value for its customers. From this perspective, BPM is still one of the business priorities. Building the organization's capacity in this area remains a significant challenge for executives and

professionals (Gartner Group, 2009 as cited in Recker et al., 2009).

Process modeling and optimization are the activities that allow generating information about the process currently taking place in the company or about the future proposal of the new process.

In this perspective, Baldam et al. (2008) say that the execution of processes ensures their implementation, such as technology transfer plans, systems migration, training, equipment, and software adjustment, and monitoring the process in the performance, monitoring, and control of instances.

In business process modeling (BPM), it is interesting to have in mind the objective of a process modeling project even before starting it because you can have several goals, such as documenting what is done, improving what is done, standardizing what has been done, eliminate processes that have been done and that do not generate value and automate processes with workflow systems, etc.

BPM is usually conducted through interviews with those responsible for the processes, by observing the execution of the processes, and by analyzing the documents, systems, and any other instruments used to support the implementation of the processes.

Currently, Business Modeling has been used in organizations that adhere to its users to enjoy manipulating information. In this context, it emerges as an ideal approach to assist organizations in more effective administrative control, facilitating the assessment of the organization's situation and identification of its real needs and serving as a basis for the title of possible systems to support processes. This entire process aims to achieve objectives, that is, a concrete point that one wants to reach, with numerical parameters, dates, and goals, understood as a segmentation of the purpose by the company.

BPMN can be highlighted among the various business process modeling methodologies. This is a notation that emerged to bridge the gap between the systematization of the company and the systems analyst through the implementation processes, bringing together the process specialist and the Information Technology specialist (Baldam et al., 2008). Another goal is to ensure that languages designed for executing business processes can be viewed as business-oriented notation. BPMN intends to unify business processes with different notations and points of view, addressing best practices within the business modeling community. However, organizational structures, functional arrangements, business rules are beyond the reach of BPMN that allows modeling external data. In addition, it is a language used to represent business processes and is expressed through standard symbols organized in a business process diagram.

3.3 Service Oriented Architecture

As software development companies grow, they go through maturing their products, which can be goods or services, using design patterns to reduce costs, reduce time and increase the quality of products or services. From this perspective, Service Oriented Architecture (SOA), according to Microsoft (2007), has become a model used in software development because it is an exciting technology, as it addresses the following aspects: the loosely coupled requirements, these represent a low level of interdependence between the modules of a system, standards-based development, protocol-independent distributed computing, application integration, and legacy systems.

According to (Marzullo, 2009), a well-planned software architecture reduces business risks and presents some advantages such as clarity in management through difficulties; standardization of language and communication between developers, customers, and managers; possibilities of reuse and consequent evolution of the system; determines some factors for suitable analysis such as consistency, quality attributes, compliance with architectural styles (Marzullo, 2009).

Thus, SOA proposes standards and development practices to enable Web services to interact appropriately in an environment as heterogeneous as the internet or any other network. Like any architecture based on components, SOA maintains each one of them and only defines how the communication between them will be done.

For Street (2008), SOA is a software architecture based on service components that can be reused. According to Ortiz (2006), Web Services, is a possible example of SOA-based technology that were created to build Internet applications and use SOA with standardized communication protocols: SOAP – Simple Object Access Protocol (Erl, 2009), WSDL – Web Service Description Language (Erl, 2009) and UDDI (Universal Description, Discovery, and Integration), which describe data through the XML language.

SOA is an architectural model aimed at the strategic planning of the information technology (IT) area, directly aligned with the business objectives. Thus, IBM (2019) argues that SOA is a corporate concept that represents an essential stage in the evolution of application development and integration that allows exposing its functionality in standardized and interrelated services, capable of leveraging legacy functionality in new markets.

According to the Electronic Government Interoperability Standards - ePING (2007), SOA is a proposed architecture for the interoperability of systems through a set of loosely coupled service interfaces. Services do not require technical details of the platform of others. It is an architecture style that promotes the integration between business and IT through services that benefits for the exchange of information to be carried out. Thus, the service is the main component of this architecture.

For Gartner Group (s.d.), SOA is an enterprise architectural approach that allows the creation of interoperable business services that can be easily reused and shared between applications and companies. Thus, according to Fronckowiak (2008), successful implementation depends on a careful approach to business architecture planning.

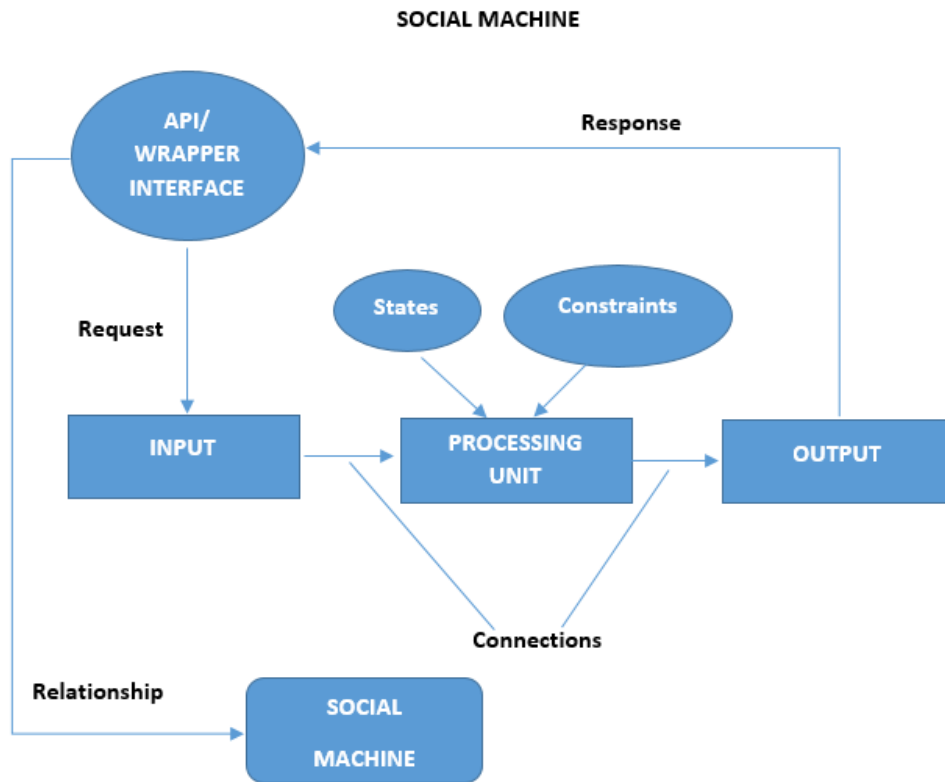
The definition of SOA (Service Oriented Architecture) produced by the Open Group team (2006) states that it is an architecture style that supports service orientation. For Kistasamy et al. (2010), SOA is a business tool that allows companies to architect their processes independently, thus leading other architectural domains to follow the same logic. This is a way of thinking in terms of 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 service composition, message-based communication, and model-driven implementation, which provide rapid development of practical solutions and flexible (Oasis, 2016). The real effect of SOA is the replacement of large, monolithic applications that have small interoperability interfaces, grudgingly provided and not guaranteed with more minor and modular services that have interface descriptions and contracts (Open Group, 2006).

3.4 Social Machines

According to Shadbolt et al. (2013), the generation of current systems will do more than connect people: they will covertly combine our social processes and help us achieve what was considered impossible. This already happens with IoT (Internet of Things), connections of devices with systems providing mechanisms for interaction with people leveraging social computing. This also serves as a reference to the Social Machine, as software platforms, including the Web, promote the creation of ecosystems of applications and services that would directly link people and organizations.

It is already known that the Social Machine can be defined as a programmable structural set that involves an information processing system that consists of a group of provided requirements and services, dynamically available under constraints that are determined by relationships, as shown in Figure 1. These constraints can be specified as rules between the Social Machines involved.

Figure 1. Representation of the relationship between Social Machines in the System approach.



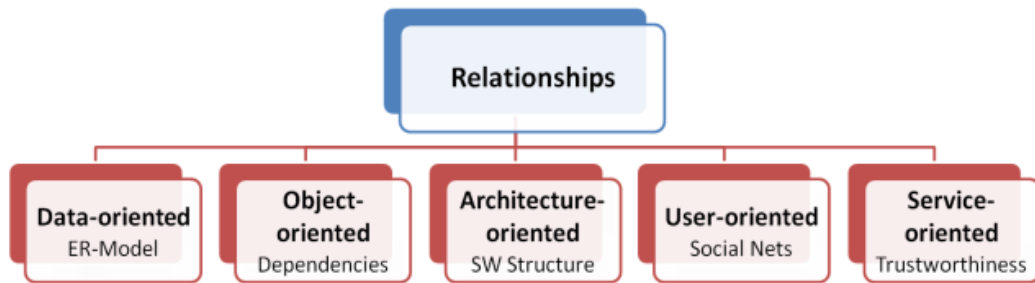
Source: Lemos de Souza e Meira (2020).

Meira et al. (2011) defined a Social Machine stating that it contains relationships, interfaces, requests, responses, state, restrictions, input, processing, and output of data. In the case of Meira et al. (2011), these relationships can occur with people, dealing 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 to services provided by the Social Machine interface. Replies are defined as a hidden reply to other machines via 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) claim that every Social Machine must receive input data, perform processing, and generate output data. The functioning of a Social Machine is to receive requests (inputs) from other SMs (Social Machines), these are processed and return responses (outputs). In addition, there will be rules that define relationships with other SMs under a specific set of restrictions.

The interaction between users and applications with a system can be described as a relationship, expressed as a dynamic set between databases and services. In Social Machines, these interactions are driven by the many possible relationships established between users, applications, and Social Machines, provided through APIs - Application Programming Interface (Shadbolt et al., 2016). When it comes to relationships, it establishes a connection with something (components), and expresses the dependencies and requirements between them, that is, restrictions, and can also be a connection between theory and practice. According to Dicio (2019), a relationship must contain its function, its representation, rules and exceptions of its establishment, its occurrence, and when it can cease to exist. For Silberschatz (2011), a relationship is an association between several entities. Elmasri and Navathe (2005) say that relationships between two or more entities show an association between them and state that “the 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 centerpiece of the Web, which can be seen as a “dynamic set of

relationships” between information collection and services (Burégio et al., 2013). The relationship semantics are now explicit and, in addition to representing static connections and dependencies, it establishes constraints that are influential in how Social Machines dynamically interact (Burégio et al., 2013). According to Burégio et al. (2013), relationship, in general, can be defined as “the way things are connected” and, in this sense, it is often used interchangeably as “connection,” “association,” “link” and “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 how two or more Social Machines are associated or interact with each other” (Burégio et al., 2013). Figure 2 shows the different types of relationships addressed in the computing area. They are data-oriented relationships, object-oriented relationships, architecture-oriented relationships, user-oriented relationships, and service-oriented relationships. The latter will be detailed in this research because the Social Machine uses services and applications involving the internet.

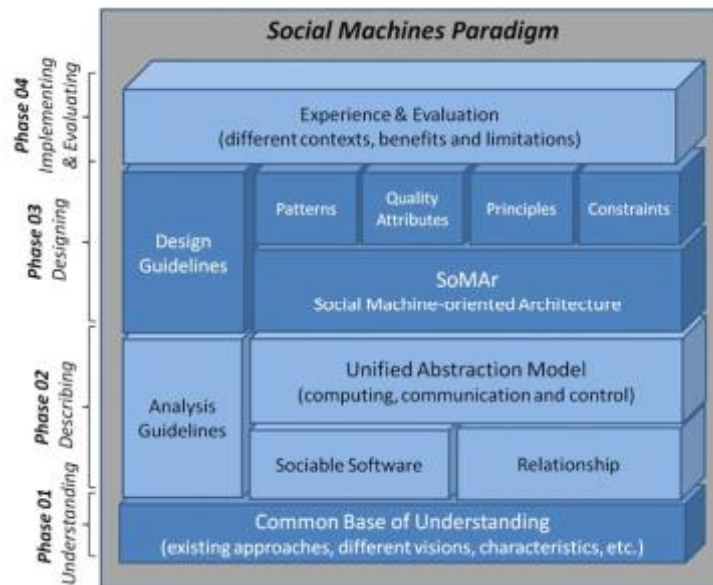
Figure 2. Different relationship views.



Source: Burégio et al. (2013).

In the service-oriented relationship view about service-oriented distributed systems, the relationship underlies the reasoning about reliability, as shown in Figure 2 above. In these systems, trust relationships are used to infer access to reputation, control of services, and resources (Suryanarayana et al., 2004). According to Nascimento et al. (2014), we are experiencing high growth in application development on the Web. The trend is for Social Machines to be thoroughly professional, with applications that connect to other machines, process their data, and, in a way, create these data available for different machines. Within the context of Social Machines, these efforts, in which individuals and machines work together in often complementary roles, represent only a first step, albeit an important one. Figure 3 illustrates a hybrid architectural style for designing Social Machines by combining different principles from current software engineering practice. The idea is that this architectural model can serve as a framework of abstraction oriented to Social Machines, making the constraints, principles, and desired properties more explicit.

Figure 3. SoMAR -Social Machine Oriented Architecture.



Source: Burégio (2014).

The overview of this paradigm of Social Machines is in four phases, according to Burégio (2014): the phase of the common base of understanding, the description phase, the conception phase, and the implementation and evaluation phase.

In phase 1, represented as an common basis of understanding, there are different views, characteristics, existing approaches, representations, etc.

In phase 2, established as a description, the idea of social software and relationship shows a unified abstraction model in terms of computational, communication, and control aspects. In this way, there is a merge of computational and social elements in software, providing a guideline for analysis to address some existing issues related to the field of systems engineering.

In phase 3, defined as conception, there is an architectural model oriented to Social Machines (SoMAR), as stated earlier. It is a hybrid architectural style used to design Social Machines through combinations of different principles, constraints, standards, and quality practices of software engineering that results in a design (project) guideline.

In phase 4, determined as implementation and evaluation, some implementations of some systems are proposed to discuss and evaluate the experiences and lessons learned with applying the Social Machines paradigm in different contexts. Some software paradigms, such as object-oriented, service-oriented, and component-based paradigms, provide us with substantial practices that can successfully support the identification and description of appropriate abstractions within the architecture.

4. Results and Discussion

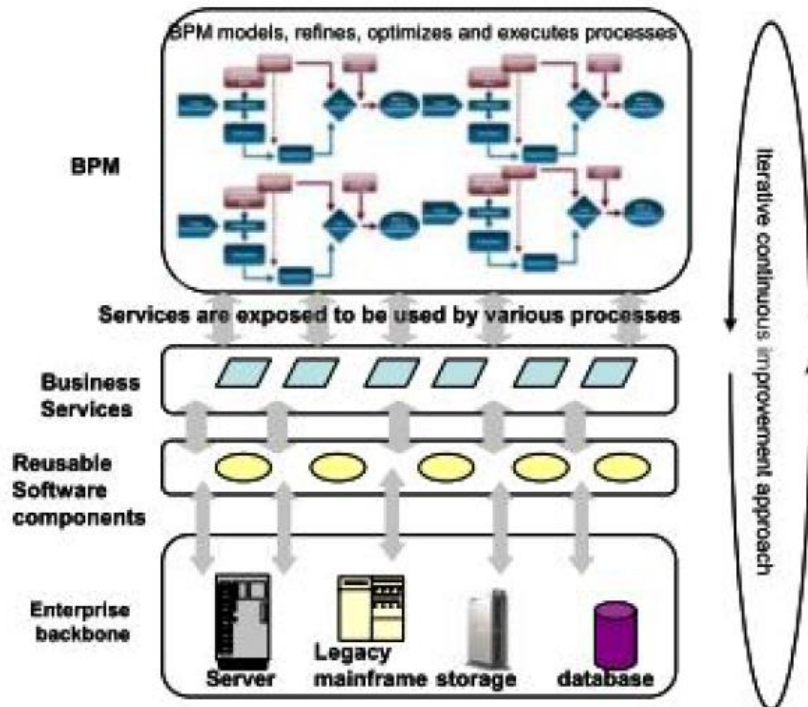
This section deals with part of the work on the convergence between BPM-SOA and UML-SOA, according to Souza (2016), and the possibility of integration with Social Machines, according to Lemos de Souza and Meira (2020).

4.1 BPM-SOA

According to Kamoun (2007), as seen in Figure 4, the BPM-SOA combination is advocated as the best approach for companies, as it brings greater alignment between a company's business processes and IT resources, achieving business agility business through greater flexibility and better reuse, reducing costs and increasing efficiency. In other words, the company has an excellent response to changes in business requirements, facilitating the integration of complex and heterogeneous

technologies.

Figure 4. Unified architecture for BPM-SOA.



Source: Kamoun (2007).

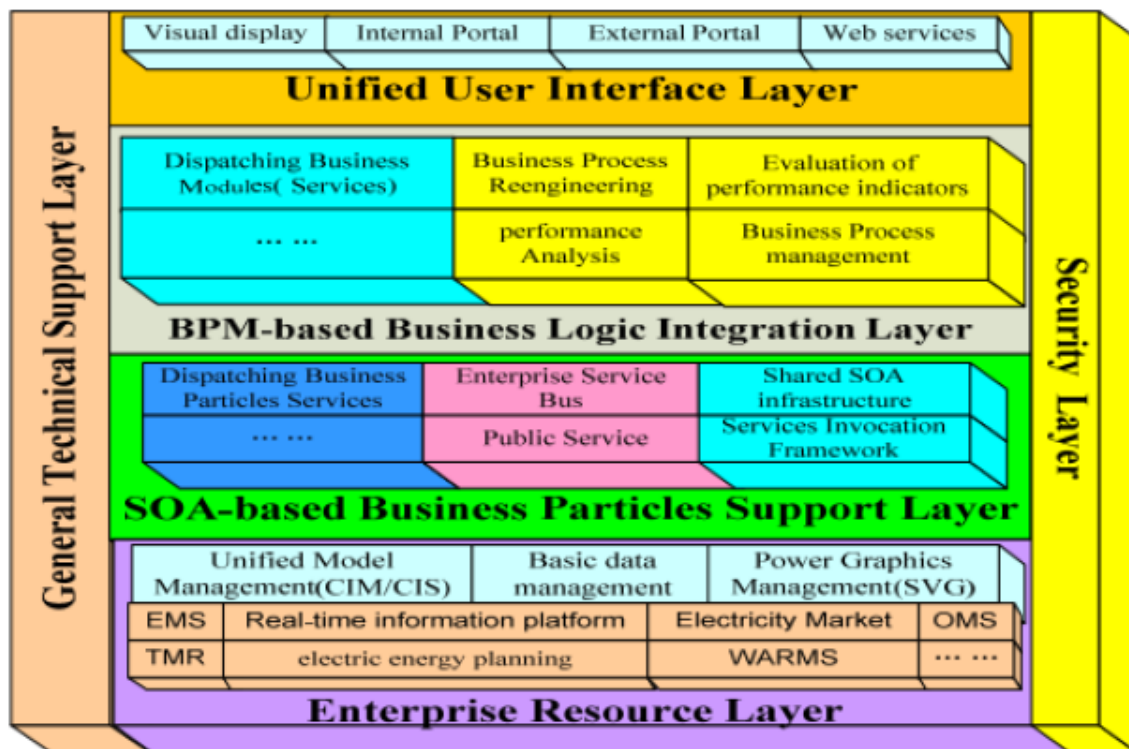
In BPM, the direction is business-oriented, with a top-down approach to processes being process models use reuse, and measurement success is through business metrics and key performance indicators. SOA is IT-oriented, with a bottom-up architectural approach, uses reuse in service implementations, is oriented to business infrastructure, and measurement success is given by architectural metrics, logical consistency, ease of integration, and cost reduction.

The growing combination of BPM and SOA has spawned some protocols and tools, but not all of them are compatible. In SOA: JCA, J2EE, JDBC, XML, WSDL, ESB, UDDI, SOAP, and BPEL, the latter, recognized as the SOA orchestration language (Kamoun, 2007).

In BPM and its technologies: BPML, BPEL4WS, RSS, XPD, BPDM, AJAX, WEB 2.0, UML, and BPMN. The latter is currently the predominant notation standard for graphically representing process models. XPD and BPDM, however, gained widespread adoption as format interchange and serialization, surpassing BPML. These compete with the BPEL standard as if BPEL is not used as an execution language, but only as an import/export interchange language, as is used by many vendors, BPEL has little value over XPD or BPDM, however. There are studies on these languages whether there is an overlap or complementarity about BPEL (Kamoun, 2007).

According to Zhang et al. (2010), BPM and SOA are technologies that have provided desirable solutions for systems integration. One of the predictions defended by these authors is that the future SOA may lie in the integration with the Web and EDA – Event-Driven Architecture (Zhang et al., 2010). From this perspective, as BPM not only encompasses the modeling of business processes, operation, monitoring, analysis, optimization, and restructuring but also plays a supporting role for the parallel operation, collaboration, and distribution of business processes, it is essential to study the following layers, shown in Figure 5.

Figure 5. BPM-SOA Layered Architectures.

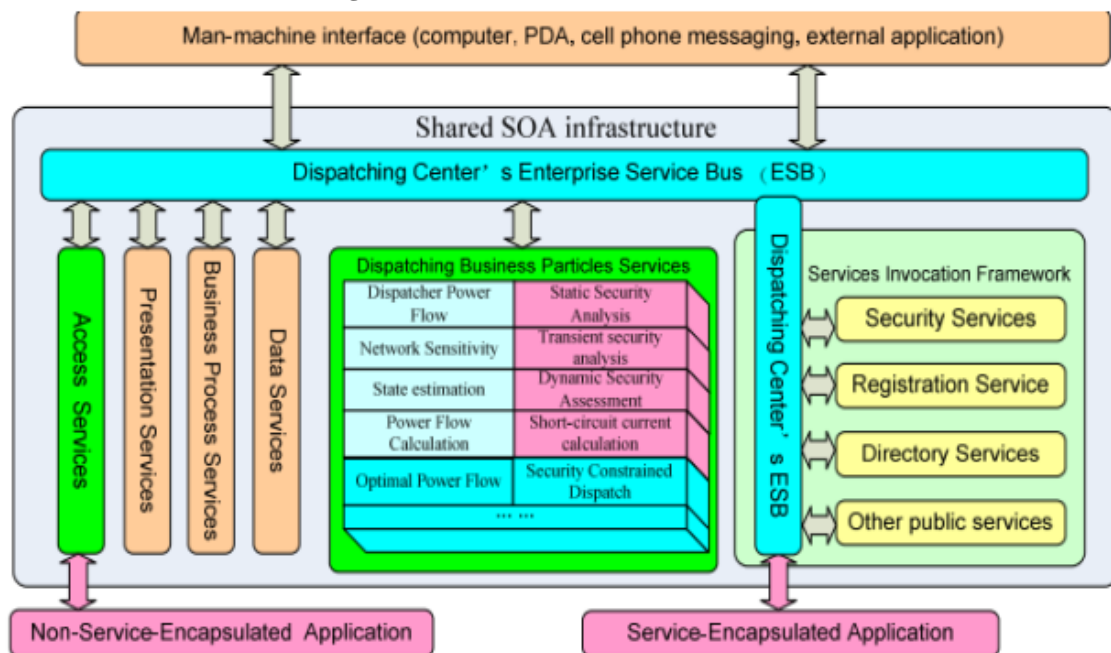


Source: Zhang et al. (2010).

- Enterprise Resource Layer, which includes existing systems, providing the upper layer with existing functions and modules that will be represented as services, in order to reuse and combine them into on-demand business processes;
- SOA-based Business Particles Support Layer, in which functions and modules provided by the previous layer are encapsulated in service interfaces;
- BPM-based Business Logic Integration Layer, a layer in which business requirements and different business processes can be designed and implemented through the composition of different services, in addition to having extra functions, including business process management, analysis data and process reengineering;
- Unified User Interface Layer, in which generated business processes are represented and provided to users using standardized access;
- General Technical Support Layer, this layer includes IT, measurement techniques, automation techniques, component technology, analysis and data mining;
- Security Layer, which represents security related to the technical support needed by all other layers. The BPM-SOA based layered architecture was designed to provide flexibility and scalability where existing functionality is represented as services that expose network interfaces and can be invoked using standardized protocols.

In another analysis pattern, according to the ESB (Enterprise Service Bus) perspective, it links and mediates all communications and interactions between services, providing the transformation of data into XML to allow services with different interfaces to communicate and so that there is the sequencing of services in the business process. Thus, ESB is designed to support multiple protocols, and routing is content-based to ensure communication quality in terms of efficiency, reliability, and security and manage transactions and service invocation, as shown in Figure 6 below.

Figure 6. ESB model based on BPM-SOA.



Source: Zhang et al. (2010).

We verify that:

- Registry Service's primary function is to provide the storage, classification, and search of service information;
- Library Registry Services describe service characteristics as a variety of service attributes, including interface description, business characteristics, and quality of service (security, reliability), in addition to the QoS (Quality of Service) of operations attributes services;
- Business Process Services is a set of standard services used to support the operations of business processes.

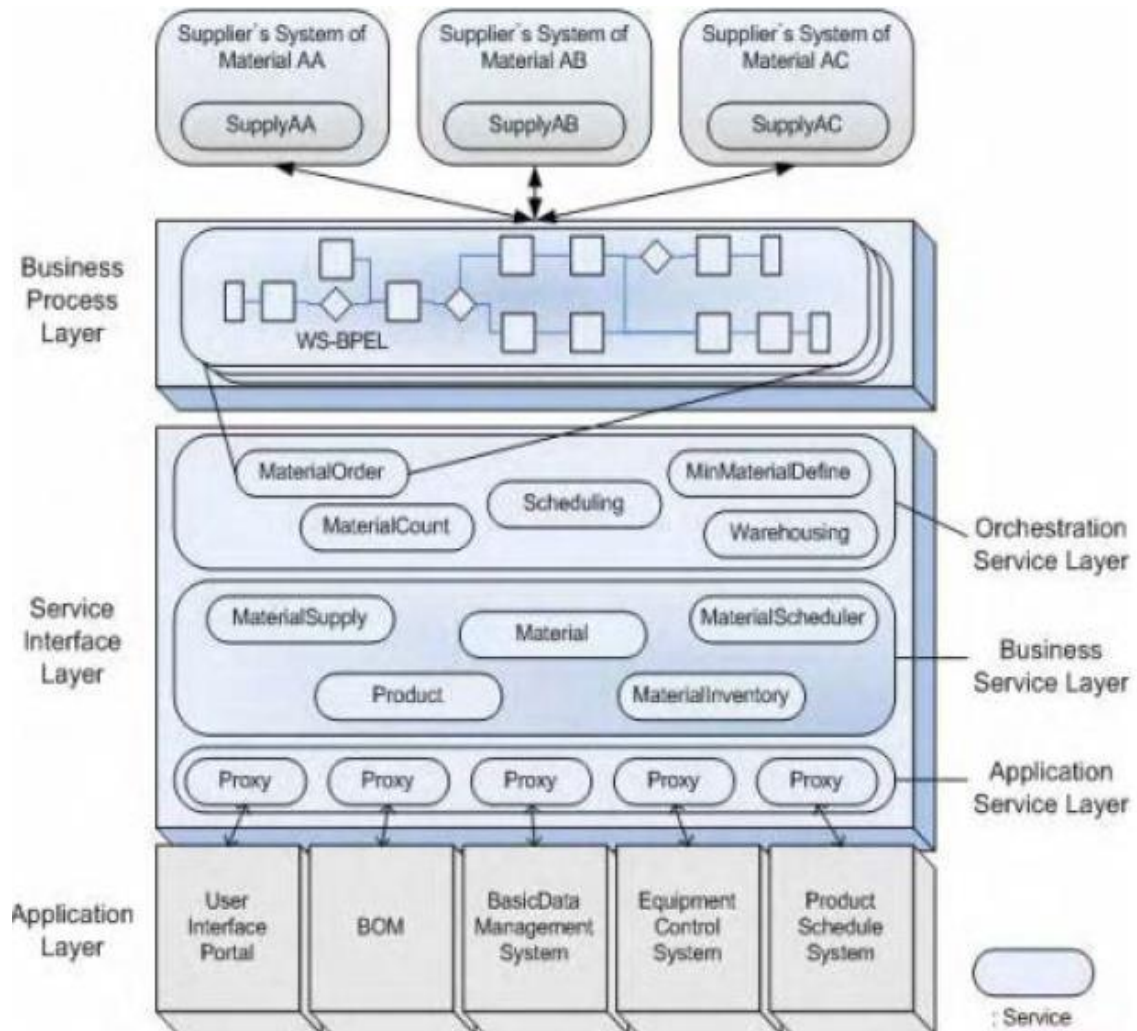
Services can be composed into business processes, and extra modules can be plugged into the architecture to support new operations. In addition, security services are designed to help trust, privacy, protection, access control, and data integrity. Given this, we realize that BPM supports the idea that we can model business processes, which are represented in a way that can be understood. In comparison, SOA is an architectural approach to systems development that uses the reuse and encapsulated business services to share different applications in a loosely coupled and highly interoperable way. Therefore, SOA aims to better align business processes with service protocols and legacy applications and software components.

4.2 UML-SOA

One of the main concerns of any executive in an organization today is integrating information systems to support business strategies. So, when it comes to UML-SOA integration, there will be two main approaches to modeling SOA integration: 1 – Unified Modeling Language (UML) and; 2 – Service-oriented architecture Modeling Language (SoaML). The UML has been widely applied as a general-purpose modeling language used in SOA by creating visual designs through a set of graphical notations. According to An et al. (2007), companies in their business environment try to integrate their IT infrastructure, automating their business processes for possible changes, since SOA is an IT standard, which supports efficient system integration and business processes by automating departments from the company. For a possible UML-SOA integration, the requirements analysis can be done through the UML, explicitly using the use case diagram, because it originates of the modeling through the detailed analysis of the process of each use case, and from there, it is delivered to SOA. Obtaining, operating, and organizing services are abstracted through three layers: application layer, service interface layer, which is subdivided into three

layers: application services layer, business services layer, and orchestration service layer and the business process layer for effective implementation is expressed in Figure 7 below.

Figure 7. Layer Architecture.



Source: An et al. (2007).

- The Application layer is responsible for providing services to applications.
 - The Service Interface layer consists of services and, in this case, divided into three layers:
 - The application service layer can process data in a new environment or in an existing environment as a service to express specialized technology function;
 - The Business Service layer is composed of services that have a business service model directly vital to SOA. In general, it acts as a controller that mixes application services. This layer may be tasked with classifying business services that encapsulate business logic;
 - The Service Orchestration layer is between the process and the SOA component. This layer consists of more than one process service, several business services, and application services according to business rules and logic. Orchestration abstracts rule and business logic practically and improve agility and reuse.
 - The Business Process layer executes the business logic as an internal and external service.
- For Hoisl et al. (2011), in recent years, service-oriented architectures have been increasingly used in business process

management. In this context, a process in SOA is specifically built to support the definition, execution, and monitoring of inter- and intra-organizational processes. The widespread use of service-oriented technologies has also led to calls for the complete integration of security features in the process development of service-oriented systems.

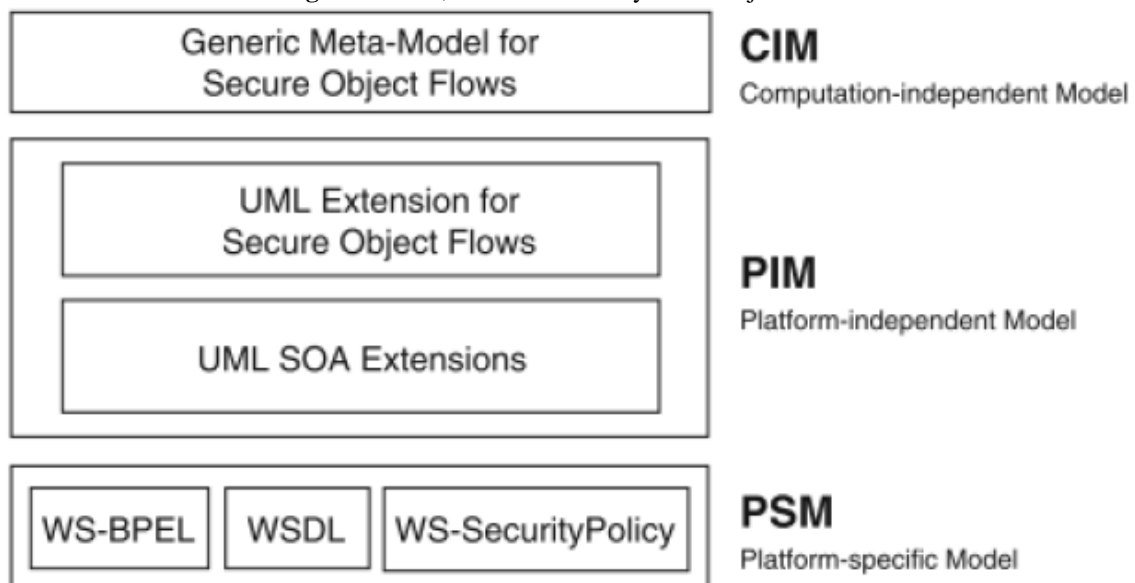
In the modeling level, whether in BPMN or UML, an object flow defined in a business process model is an object passed from one node to another. In the context of Model-Driven Development (MDD), see Figure 8 below, an independent model of computation (CIM) defines a specific domain (or sub-domain) at a generic level. This model is independent of unique modeling and can build a platform-independent model (PIM) of the corresponding domain. Although it is platform-independent and thus implementation-neutral, PIM is typically specified in a particular modeling language (BPMN or UML) and describes the structure of a system and the elements or results produced by a system, or the control of object flow in a system. Finally, a platform-specific model (PSM) describes the realization or implementation of a software system using platform-specific technologies and tools.

In the CIM layer, a generic metamodel for object streams is provided that can be used to extend arbitrary process modeling languages.

A UML extension is provided in the PIM layer that allows modeling object flows via an extended activity diagram, integrated with the SoaML and UML4SOA extension to define process-oriented object flows (Hoisl et al., 2011).

In the PSM layer, WS – BPEL, WSDL, and WS – SecurityPolicy are used, which are specifications of the PIMs. WS–BPEL's role is a standard based on business processes for Web services, WSDL is a language to describe the interface of Web services, and WS – SecurityPolicy expresses security requirements (Hoisl et al., 2011).

Figure 8. CIM, PIM and PSM layers for object flows.



Source: Hoisl et al. (2011).

According to Hoisl et al. (2011), SoaML provides essential modeling primitives for structural views of a service architecture (including participants, collaborations, contracts, and service interfaces, as well as messages). It offers an extension metamodel of the composite structure in UML. This extension allows the composition of consumers and service providers through a set of interactions with entities, referring as participants. UML4SOA is designed to model object flow as an integral part of service orchestration. A SoaML extension is a service-oriented process model through orchestration specifications defined with UML activities.

For Todoran et al. (2011), SoaML is a modeling language for SOA integration, which predominantly acts on distributed applications and has three main parts: a provider, a consumer, and a registry. It is a recent OMG (Object Management Group) specification, first beta version released in April 2009, designed specifically to meet SOA requirements, and consists of a UML profile and metamodel for the specification and design of services in SOA and does not make any changes to existing UML notations, model elements, or semantics, but adds new notation, elements of the model and semantics to support the following new modeling capabilities:

- Identification services and the requirements they are then intended to fulfill, and anticipated dependencies between them;
- The specification of services through their functional capabilities;
- Definition of consumers and service providers through the services they consume and provide;
- The definition of policies for the use and provision of services, linking them to related metamodels, such as the BMM (Business Motivation Model) and BPMN (OMG, 2009).

In first, this seemed to be the most viable solution for SOA Integration Modeling. However, the high degree of generality of the method and the fact that it was initially thought of as a language for working with artifacts of a complex object-oriented software system has led to a need for a specific language, particularly for SOA modeling. This made SoaML emerge as a UML profile and metamodel for modeling and designing services within a service-oriented architecture. From this perspective, while UML comes with different modeling profiles and methodologies to support SOA modeling, SoaML provides a standard and specific way of modeling. Thus, the concept of SoaML revolves around the idea of services, which is the work provided to one another.

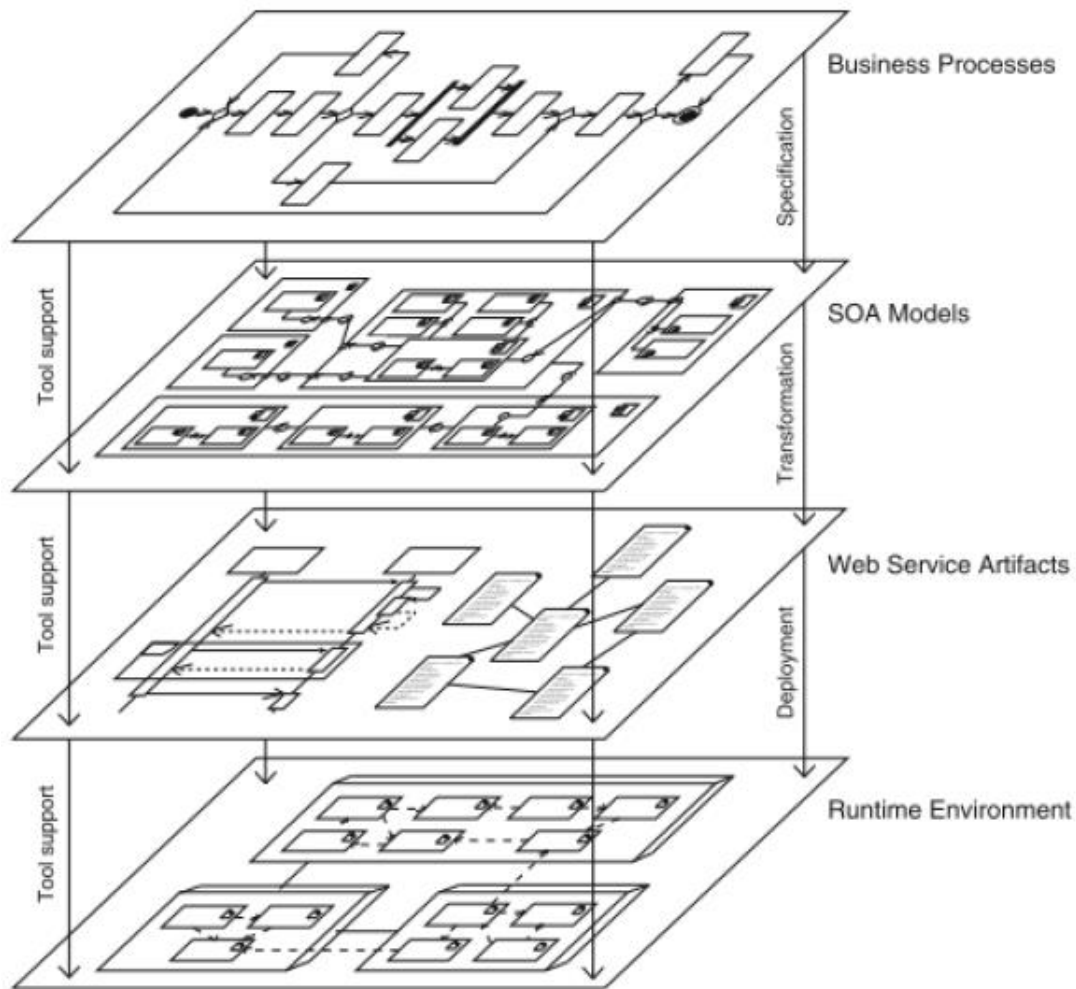
Several points of view must be considered to support the definition of object flow in an SOA context. In this context, the specifications of choreography and orchestration services are of particular importance:

- Choreography specifications for object flows includes modeling support for invoking data (such as input and output parameters) that require integrity and/or confidentiality;
- Orchestration specification for object streams includes modeling support for object stream of process execution data as well as invocation data.

As for workflow models that use BPMN, it is necessary to consider that they only provide model annotations for security properties without processable or formalized semantic restrictions. Therefore, the approach does not support the security specification.

In Figure 9 below, integrated tools approach to modeling and enforcing process-oriented object flow in SOA. In particular, a generic metamodel (CIM) for object flow is provided whose corresponding UML extension defines platform-independent models (PIM), a PIM-to-PSM model transformation, with artifact transformations for the web service as well as the corresponding tool holder.

Figure 9. Built-in support tools for secure object flow.



Source: Hoisl et al. (2011).

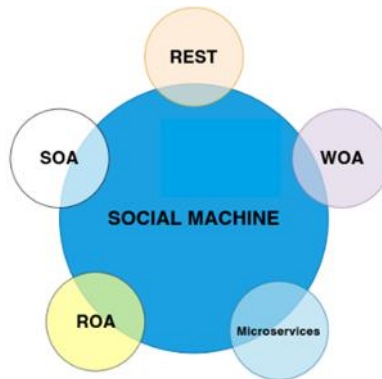
Therefore, we verified that this approach allows the continuous specification and execution of confidentiality and integrity properties with the object flows and business processes executed in a distributed system. Using SoaML/UML4SOA, with modeling support for secure object flow, is possible because most UML tools directly support profiling, and it is relatively easy to integrate with UML profiling in a software tool.

However, these previously discussed contributions only discuss limited modeling options specific to the PIM level. They do not provide generic CIM or PIM models or built-in tool support. Thus, it is presented through MDD techniques the construction of an integrated approach to specification and execution of object flow with process-oriented SOA. In addition, a detailed description of the capabilities for modeling secure object flow in SOA is provided, both at a generic level and the modeling language level. Furthermore, it presents a support tool (including joining model transformations) for specifying and deploying object flows.

4.3 Integration with Social Machines

Knowing that a Social Machine is an information system that uses relationships and restrictions, then, in this research, from the view of the types of relationships, this being are service-oriented, stands out the types of existing architectural styles, and your characteristics.

Figure 10. The Social Machine and service-oriented relationships.



Source: Lemos de Souza e Meira (2020).

Although there are different architectural styles, as shown in Figure 10, in none of them is the issue of trust in decentralized environments explicit, with the architectural style being the combination of distinct characteristics in which architecture is executed or expressed (OPEN GROUP, 2006).

Because of the fact already stated, since a Social Machine architecture can become functional, it must be composed of concrete architecture, in this case, an architecture that represents a service-oriented relationship.

Figure 11 illustrates the association between two Social Machines through a service-oriented relationship.

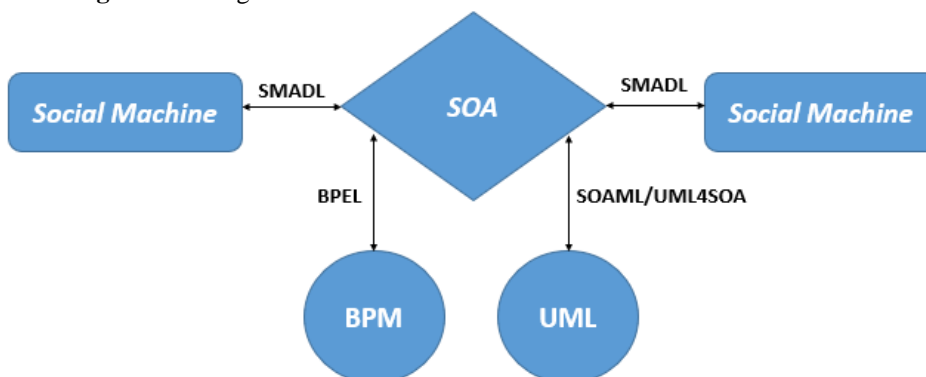
Figure 11. Service-oriented relationship of Social Machines.



Source: Lemos de Souza e Meira (2020).

Figure 12 shows interactions between Social Machines based on service-oriented architecture through SMADL - Social Machines Architecture Description Language (Nascimento et al., 2012), in addition to the convergences between UML-SOA and BPM-SOA (Souza, 2016).

Figure 12. Integration of Social Machines and BPM-SOA and UML-SOA.



Source: Authors.

In this way, BPM, UML, and Social Machines can be converged through unified architectures combined with SMADL-BPEL (Social Machines and BPM) and SMADL-SOAML/UML4SOA (Social Machines and UML). These combinations lead to other ways of functioning Web services together with computational and human elements since many collaborative platforms, social networks, and applications are developed to ease the communication of services. These services produce effects that have value for the people or organizations that are their consumers, as a service architecture generally follows a context of people, processes, and technologies.

5. Final Considerations

It was chosen to analyze BPM because its methodology guarantees a fast development practice of the system, making it flexible and allowing its reuse, using legacy technologies, as companies can achieve greater control of business processes. On the other hand, UML enables standardization in software modeling and, like BPM, with possible integration with SOA. In this way, expanding the combinatorial view to Social Machines is relevant, since an organization combines processes, technologies, and people and the Social Machine itself that emerges from a socio-technical system that portrays a combination of machines (technical computational elements), whether these are processes and technologies and the social (people).

However, for future work, there is a need for further studies on BPM and UML integration with service architectures, whether SOA, mentioned in this research, as well as Microservices¹ (Lemos de Souza & Meira, 2021), along with the architectures of Social Machines that require conceptual standardization and technical knowledge (Lemos de Souza & Meira, 2021). The convergence between modeling tools and service architecture is an area of process management that needs to be better developed. It is necessary to have technical knowledge and skills, including integration issues with Social Machines, interaction languages, orchestration, and choreography² of services.

Acknowledgments

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

References

- Amsden, J. (2010). Modeling with SoaML, the Service-Oriented Architecture Modeling Language: Part 1. Service identification. <https://paulallen.ca/documents/2015/03/amsden-j-modeling-with-soaml-part-1-service-identification-2010.pdf/>.
- An, M. J. et al. (2007). Design of the material control system based on service-oriented architecture, Control, Automation and Systems. ICCAS '07. International Conference on, Seoul, 978-983. 10.1109/ICCAS.2007.4407056.
- Baldam, R. et al. (2008). Gerenciamento de Processos de Negócios. BPM – BUSINESS PROCESS MANAGEMENT. (2a ed.), Érica.
- Booch, G. et al. (2000). UML: Guia do Usuário. Editora Campus.
- Burégio V. et al. (2014). Personal APIs as an Enabler for Designing and Implementing People as Social Machines. International World Wide Web Conference Committee (IW3C2).
- Burégio V. (2014). Social Machines: A Unified Paradigm to Describe, Design and Implement Emerging Social Systems. Tese de Doutorado em Ciência da Computação, Centro de Informática da Universidade Federal de Pernambuco. Repositório da UFPE. <https://repositorio.ufpe.br/handle/123456789/12430>.
- Burégio, V.A. et al. (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.
- Burégio V. et al. (2013). Social Machines: A Unified Paradigm to Describe Social Web-Oriented Systems. *International World Wide Web Conference Committee (IW3C2)*.

¹ Architectural and organizational approach to software development in which the software consists of small independent services that communicate using well-defined APIs (<https://aws.amazon.com/pt/microservices/>).

² Service orchestration coordinates the interaction between different services, and service choreography employs a decentralized approach to service composition, describing the interactions between multiple services (<https://qastack.com.br/programming/4127241/orchestration-vs-choreography>).

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

Elmasri, R. & Navathe, S.B. (2005). *Sistemas de banco de dados 4ª edição*. Revisor técnico Luis Ricardo de Figueiredo. Pearson Addison Wesley.

Eping. (2007). *Padrões de Interoperabilidade de Governo Eletrônico*. Documento de Referência, versão 3. <http://eping.governoeletronico.gov.br>.

Erl, T. (2008). *Principles of Service Design*. Prentice Hall.

Froncowski, J. (2008). *SOA Best Practices and Design Patterns*. http://i.zdnet.com/whitepapers/Oracle_SOA_US_EN_WP_BestPractices.pdf.

Group, G. (s.d). *SOA*. <http://iprocess.com.br/soa>.

Hoisl, B. et al. (2012). Modeling and enforcing secure object flows in process-driven SOAs: an integrated model-driven approach. *Softw Syst Model*, 13(2), 513-548. Springer Science + Business Media. <http://dx.doi.org/10.1007/s10270-012-0263-y>.

IBM (2019). *SOA (Service-Oriented Architecture)*. <https://www.ibm.com/sg-en/cloud/learn/soa>.

Kamoun, F. (2007). A roadmap towards the convergence of business process management and service oriented architecture. Ubiquity, .1-1. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/1241854.1247273>.

Kistasamy, C. et al. (2010). The Relationship between Service Oriented Architecture and Enterprise Architecture. 129 - 137. 10.1109/EDOCW.2010.12.

Lemos, H. D. (2009). Encapsulamento, Polimorfismo, Herança em Java. <http://www.devmedia.com.br/encapsulamento-polimorfismo-heranca-em-java/12991>.

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)* 1, 788-796. 10.5220/0009580507880796.

Lemos de Souza, B. (2020). Grounded Theory of the Evolutionary Behavior of Social Machines. In *Doctoral Consortium on Enterprise Information Systems - (DCEIS 2020) - Final Program and Book of Abstracts*.

Lemos de Souza, B. W. (2020). Ontology of Social Machines. 15th Iberian Conference on Information Systems and Technologies (CISTI), Sevilla, Spain, 2020, 1-4, 10.23919/CISTI49556.2020.9140830.

Lemos de Souza, B. W. & Meira, S. R. L. (2020). Combination of social machines and service-oriented relationships. *International Journal of Development Research*, 10, (11), 41864-41870.

Lemos de Souza, B. W. & Meira, S. R. L. (2020). Diagnosis of Comparative and Future Studies on Social Machines. *International Journal of Science and Research Methodology (IJSRM)* 17 (1): 122-148.

Lemos de Souza, B. W. & Meira, S. (2021). Máquina Social um paradigma emergente ainda desconhecido. *International Conference on Information Technology & Systems (ICITS'21)* e *Revista Ibérica de Sistemas e Tecnologias de Informação (RISTI)*.

Lemos de Souza, B. W. & Meira, S. (2021). Tendências de Relationship-aware entre Máquinas Sociais e Microserviços. *Revista Ibérica de Sistemas e Tecnologias de Informação (RISTI)*.

Marzullo, F. (2009). *SOA na prática*. Novatec.

Meira S. et al. (2011). The Emerging Web of Social Machines. <https://arxiv.org/ftp/arxiv/papers/1010/1010.3045.pdf>.

Microsoft. (2007). *SOA in the Real World*. <https://www.infoq.com/news/2007/07/SOA-in-the-Real-World/> and <http://s3.beckshome.com/20070727-SOA-In-The-Real-World.pdf>.

Nascimento, L. et al. (2012). SMADL: The social machines architecture description language. *CEUR Workshop Proceedings*.

Nascimento, L. et al. (2014). A New Architecture Description Language for Social Machines. In *2014 Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, pages 873 – 874.

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

OMG. (2005). OCL 2.0 Specification. <http://doc.omg.org/formal/06-05-01.pdf>.

OMG. (2005). Unified Modeling Language: diagram interchange. <https://staff.info.unamur.be/ven/CIS/OMG/UML/diagram%20interchange.pdf>.

OMG. (2009). Service oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and Metamodel for Services (UPMS). <http://www.omg.org/spec/SoaML/1.0/Beta2/PDF>.

Ortiz, G. & Hernandez, J. (2006). Toward UML Profiles for Web Services and their Extra-Functional Properties, 2006 IEEE International Conference on Web Services (ICWS'06), Chicago, IL, 2006 pp. 889-892. 10.1109/ICWS.2006.130

Prestes, M. L. M. (2011). *A pesquisa e a construção do conhecimento científico: do planejamento aos textos, da escola à academia (4a ed.)*. Rêspel.

Recker, J. et al. (2009). Business Process Modeling- A Comparative Analysis. *Journal of the Association of Information Systems*, 4, 333-363.

Shadbolt, N. et al. (2016). The Rise of Social Machines The development of a human/digital ecosystem. *IEEE Consumer electronics magazine*.

Shadbolt, N. et al. (2013). Towards a classification framework for social machines. SOCM2013: Workshop on Theory and Practice of Social Machines, WWW2013, 905-911.

Silberschatz, A. et al. (2011). Database System Concepts, (6a ed.), McGraw-Hill.

Souza, B. W. L. (2016). Convergência entre BPM-SOA e UML-SOA: uma análise comparativa de integração e de cenários. Dissertação de mestrado da Universidade Federal de Pernambuco. https://repositorio.ufpe.br/bitstream/123456789/18365/1/Disserta%20Final-Biblioteca_Brunno.pdf.

Suryanarayana, G. et al. (2004). PACE: an architectural style for trust management in decentralized applications, Proceedings. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), pp. 221–230.

Street, J. & Goma, H. (2008). Software Architectural Reuse Issues in Service-Oriented Architectures, Hawaii International Conference on System Sciences, Proceedings of the 41st Annual, Waikoloa, HI, 2008, pp. 316-316. 10.1109/HICSS.2008.406.

Todoran, I. et al. (2011). SOA Integration Modeling: An Evaluation of How SoaML Completes UML Modeling, 2011 IEEE 15th International Enterprise Distributed Object Computing Conference Workshops, Helsinki, 57-66. 10.1109/EDOCW.2011.48

Wilcox, P. A & Gurău, C. (2003). Business modelling with UML: the implementation of CRM systems for online retailing. Journal of Retailing and Consumer Services, 10(3),181-191. [http://dx.doi.org/10.1016/s0969-6989\(03\)00004-3](http://dx.doi.org/10.1016/s0969-6989(03)00004-3).

World wide web consortium (W3C). (2002). Web Service Glossary. Version 20040211. <http://www.w3.org/TR/ws-gloss>.

Zhang, L. et al. (2010). SOA-BPM-Based Architecture for Intelligent Power Dispatching System, 2010 Asia-Pacific Power and Energy Engineering Conference, Chengdu, 1-4. 10.1109/APPEEC.2010.5448783.