Bridging the Gap Between the Business and Social Worlds: A Data Artifact-Driven Approach

Zakaria Maamar[1(✉)], Vanilson Burégio[2], Mohamed Sellami[3], Nelson Souto Rosa[4], Zhengshuai Peng[3], Zachariah Subin[3], Nayana Prakash[3], Djamal Benslimane[5], and Roosevelt Silva[6]

[1] Zayed University, Dubai, UAE
zakaria.maamar@zu.ac.ae
[2] Federal Rural University of Pernambuco, Recife, Brazil
[3] ISEP Paris, Paris, France
[4] Federal University of Pernambuco, Recife, Brazil
[5] Claud Bernard Lyon 1 University, Lyon, France
[6] Recife Center for Advanced Studies and Systems, Recife, Brazil

Abstract. The widespread adoption of Web 2.0 applications has forced enterprises to rethink their ways of doing business. To support enterprises in their endeavors, this paper puts forward business-data artifact and social-data artifact to capture, respectively, the intrinsic characteristics of the business world (associated with business process management systems) and social world (associated with Web 2.0 applications), and, also, to make these two worlds work together. While the research community has extensively looked into business-data artifacts, there is a limited knowledge about/interest in social-data artifacts. This paper defines social-data artifact, analyzes the interactions between business- and social-data artifacts, and develops an architecture to support these interactions. For demonstration purposes, an implementation of a socially-flavored faculty-hiring scenario is discussed in the paper. The implementation calls for specialized components known as social machines that support artifact interaction.

Keywords: Business-data artifact · Social-data artifact · Social machine · Web 2.0 application

1 Introduction

Continuous progress of Information and Communication Technologies (ICT) has allowed enterprises to deploy better e-business applications. Among these ICTs, the Web now is a robust platform upon which cross-border business processes are deployed. "*A process is nothing more than the coding of a lesson learnt in the past, transformed into a standard by a group of experts and established as a mandatory flow for those who must effectively carry out the work*" [32].

A. Hameurlain et al. (Eds.): TLDKS XXXV, LNCS 10680, pp. 27–49, 2017.
https://doi.org/10.1007/978-3-662-56121-8_2

An online presence on the Web along with efficient business processes have become mandatory for any enterprise. To ensure that business processes perfectly capture an enterprise's best practices when satisfying and addressing end-users' requirements and concerns, respectively, IT practitioners use multiple techniques such as interviews and observations. Unfortunately, most techniques assume that end-users are familiar with the technical jargons of IT practitioners, which is not always the case. As a consequence, end-users (i.e., non-IT practitioners) do not properly express their requirements and concerns, which has a negative impact on the efficiency of IT practitioners' solutions. To mitigate this impact, business-data artifacts (for short, Business Artifacts (BA) are hailed for their particular role in bridging the gap between IT practitioners and non-IT practitioners [10,13,17]. BA is a concrete, identifiable, self-describing chunk of information that can be used by a business person to actually run a business [31]. BAs have life-cycles consisting of states and vary from one application domain to another.

A total reliance of enterprises on business processes might not be the sole response to today's challenges like globalization and capital scarcity [19]. Along with a smart management of business processes [12], Web 2.0 technologies (e.g., REST and JSON) and Web 2.0 applications (e.g., LinkedIn and Facebook) are, here, to help enterprises tap into social media's opportunities so they reach out to more customers, profile customers, gauge customers' satisfaction levels, etc. Enterprise 2.0 (or social enterprise [15]) is the one that successfully connects the business and social worlds together in a way that the specificities of both the business community (e.g., how to handle a new tax law) and the social community (e.g., how to respond to a customer's tweet) are accommodated. In [25], we introduce the concept of social-data artifact (for short Social Artifacts (SA)[1] that will work hand-in-hand with BAs. On the one hand, the business world includes BA-based processes that enterprises roll out in response to specific events (e.g., customer order) and in compliance with specific regulations (e.g., tax laws). On the other hand, the social world includes SAs that are created in response to events (e.g., social media campaign) and activities (e.g., product recommendation) that happen over Web 2.0 applications. To illustrate BA/SA collaboration, we adopt faculty hiring as a running example. The example consists of two parts: (*i*) business part is concerned with for instance, securing the necessary administrative approvals, defining criteria for shortlisting applicants, and preparing offers to applicants, and (*ii*) social part is concerned with for instance, posting job openings on Web 2.0 applications, answering applicants' questions, and screening applicants' social pages. How to identify the necessary SAs and how to ensure their interactions with BAs are key questions that we address in this paper.

In our previous work [25], we briefly defined SA, sketched an architecture for enterprise 2.0, briefly compared BA and SA, and implemented a proof-of-concept for creating SAs. In this paper, we revise the definition of SA using 2 perspectives referred to as data and operation, provide additional details on

[1] Some authors use the term of Web 2.0 artifact [3].

this architecture in terms of components' roles, define the messages that the business and social worlds exchange in the context of enterprise 2.0, establish correspondences between business and social artifacts, discuss how social artifacts are obtained, and finally, implement the faculty hiring running-example to illustrate these messages and analyze the data exchange. The remainder of this paper is organized as follows. Section 2 presents BAs in brief and then an overview of the use of social software in the context of enterprises. Section 3 details the BA-SA collaboration using a running example, presents the support architecture for this collaboration, and analyzes the interaction messages. A system implementing this example is detailed in Sect. 4. Finally, Sects. 5 and 6 list some future work initiatives and draws some conclusions, respectively.

2 Background

This section provides an overview of BAs and then, discusses the role of social software in supporting enterprise 2.0.

2.1 Business-Data Artifacts in a Nutshell

BA is a concrete, identifiable, self-describing chunk of information that can be used by a business person to actually run a business [31]. Several initiatives on BAs are reported in the literature [1,2,10,17,22,29–31,34,36,37].

Kumaran et al. [22] acknowledge the importance of identifying BAs in domain-specific applications. To this end, they assist IT practitioners identify the necessary BAs through guidelines. Narendra et al. [30] present an approach to model business processes using context-based artifacts and Web services. The authors abstract processes using models that are expressive (i.e., easy to grasp) for non-IT practitioners. These models could be based on BAs. As stated earlier, the technical jargon does not help non-IT practitioners properly express their needs. Maamar et al. [24] examine how to derive BAs from business requirements, so they present an approach for BA discovery and modeling. The approach uses a bottom-up analysis to determine fine-grained data, which are afterwards aggregated into clusters where each cluster being a potential BA. Next, the analysis derives the operations that act upon the discovered data clusters. Finally, the data and operation clusters are grouped into final BAs. Nigam and Caswell [31] define BAs along with their life-cycles in business modeling. In this case, a BA is a formal structure suitable for business people so they can manage, analyze, and control their daily business operations. Last but not least, Popova et al. [34] acknowledge the role of BAs in modeling business processes and propose a set of methods to discover BAs' life-cycles. The methods are implemented as software plug-ins for ProM (www.promtools.org), a generic open-source framework for supporting different process mining techniques.

In line with the aforementioned initiatives, we consider a BA as a chunk of meaningful information related to a certain business operation. Figure 1 illustrates the life-cycles of three BAs: order, customer, and bill. These BAs are used in

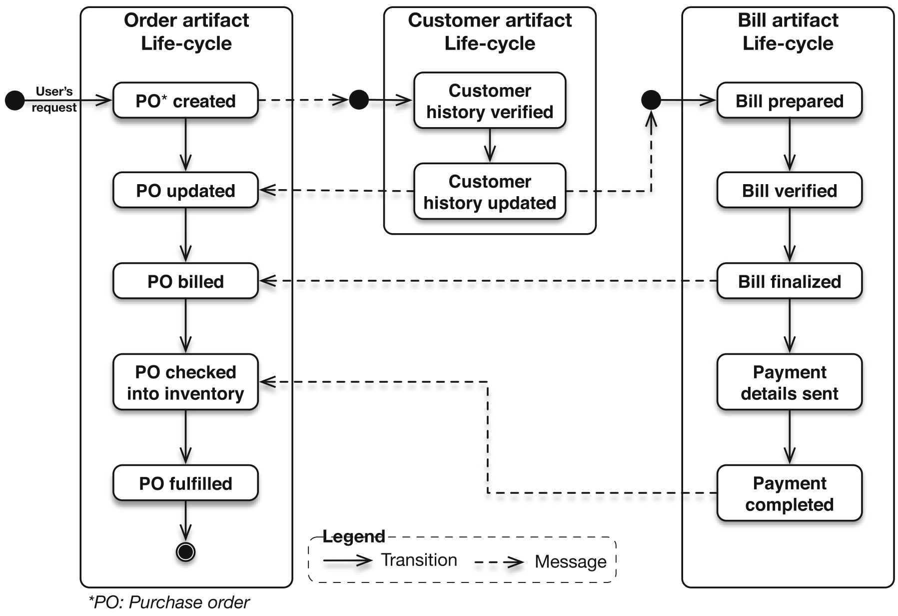

Fig. 1. Representation of a BA-based purchase-order BP (adapted from [30])

the context of a simplified version of the traditional purchase-order scenario [30] where different steps fulfilling a purchase order (e.g., creation/update by the customer and billing) are illustrated. In this figure, rounded rectangles, plain lines, and dashed lines represent artifact states, transitions within the same life-cycle, and messages between states in separate life-cycles, respectively.

2.2 Enterprise 2.0

In addition to regular Web 2.0 applications like Facebook and Instagram, enterprise 2.0 has the opportunities to capitalize on specific software referred to as social software. We discuss social software and then illustrate how it is adopted in two specific areas: business process management and coordination. According to Global Industry Analysts, Inc. "*The global expenditure on Enterprise Web 2.0 is forecast to reach $5.7 billion by 2015, driven by expanding broadband capabilities, decreasing prices, improving performance of networks, and the development of advanced, highly interactive Web 2.0 applications*" [40] and "*... the top 15 Web 2.0 vendors will spend $50 billion in 2015 on servers, networks, and other infrastructure, up from $38 billion in 2014 and $30 billion in 2013*" [39].

According to Warr [38], social software includes a large number of tools used for online communication, e.g., instant messaging, text chat, Internet fora, weblogs, Wikis, social network services, social guides, social bookmarking, social citations, social libraries, and virtual worlds. Erol et al. [14] mention that the roots of social software can be traced back to the 40s and, also, add that impressive results are obtained without a central plan or organization. Social software supports interactions between individuals who do not necessarily know each other and are not organized in any hierarchy. According to Bruno et al. [5], social software has four characteristics: weak ties, social production, egalitarianism, and mutual service provisioning.

A number of projects demonstrate the use of social software in business process management. The AGILe busIness PrOcess (AGILIPO) project embeds social software into business process tools [35]. Because business processes are inherently incomplete, human involvement is usually required. AGILIPO considers that stakeholders play three different roles, namely executor, modeler, and developer. Brambilla et al. [4] propose a specific notation to design social business processes. The notation includes new types of events and tasks like broadcast, post, and invite to activity. Grim-Yefsah et al. [18] acknowledge the existence of informal networks in the work environment and that people rely on these networks to complete their duties. The informal networks perfectly co-exist with regular networks based on formal relations (e.g., supervision). In fact, it is acceptable that the "formal" executor of a task informally seeks assistance from peers in the organization. Busch and Fettke [8] point out that business process management can be improved through social-network analysis in such a way that existing relations among employees are exposed. An accurate assessment of how a work is achieved is to analyze "as-is" processes (what is really happening) *versus* "to-be" processes (what has been defined).

Regarding the role of social software in improving coordination, multiple projects exist as well. Dengler et al. [11] promote social software like Wikis and networks to ensure the success of collaborative process activities. These activities take users' profiles posted on social networks so that appropriate users are identified. Dengler et al.'s approach reuses best practices to facilitate the creation of process models. Masli et al. [27] propose a social coordination mechanism based on the notion of "tentative event" in order to schedule meetings. Meetings are defined with attributes (e.g., date, time, and title) and confirmed upon all attendees' approvals. Coordination means making suggestions for these attributes, either approving or rejecting these suggestions, and giving arguments and counter-arguments before any agreement is reached. Kotlarsky et al. [21] analyze the role of social coordination mechanisms in reducing the knowledge gap between people (e.g., misunderstanding) and creating a social capital (e.g., building trust relationships) for the global team in virtual enterprises. These mechanisms include communication activities, working relationships, and social cognition.

3 Engaging Business- and Social-Data Artifacts in collaboration

In Sect. 1, we mention that enterprise 2.0 is built upon the connection of the business and social worlds together. In this section, we define the constituents of each world in terms of BAs and SAs, revisit the definition of SA presented in [25], and discuss how this connection is shaped through a meet-in-the-middle platform. We illustrate BAs and SAs collaboration with a faculty-hiring scenario.

3.1 Faculty Hiring Running-Example

Knowing that more and more people post their personal profiles (e.g., experience, education, and activities) on social medias[2], many organizations are using these medias as an additional source of information in their hiring endeavors. According to a Forbes post (www.facebook.com/forbes/posts/10153049903042509), 67% of people looking for a job say they use Facebook in their social media search and some 73% of recruiters say they have used some form of social media to hire staff. In addition, "*... The most effective organizations make smart use of employee networks to reduce costs, improve efficiency, and spur innovation*" [9].

Our running example targets universities that could tap into applicants' professional (e.g., LinkedIn and ResearchGate) and social (e.g., Twitter and personal blogs) worlds. Upon receiving applications, a Human Capital (HC) staff screens them for suitability according to the available job postings. One of the common practices, nowadays, is that CVs include URLs to candidates' personal Web pages showing for instance, their academic credentials and professional experiences. Some URLs, also, refer to candidates' social media pages inviting potential universities to check the activities in which they have engaged, such as helping out local communities and supporting special events. The HC staff screens the social pages by pointing his/her browser towards the appropriate URLs. On top of being "painful" and time consuming, this manual screening could overlook some important details if the HC staff has to do it for every candidate.

In this paper, we illustrate how an automatic screening of applications permits to collect details from candidates' professional and social pages. Today's networks like LinkedIn[3] already structure users' profiles in a way that makes their skills, interests, experiences, and educations easy to identify and process. In fact, LinkedIn even provides ways to programmatically access their profiles via online Application Programming Interfaces (API).

3.2 Definition of Social-Data Artifact

Prior to defining SA, we introduce the concept of social action. A social action is an operation that a Web 2.0 application offers to users and other applications (whether Web 2.0 or not) for execution over this Web 2.0 application. Multiple social actions exist like post, invite, tag, and share, and vary from one Web 2.0 application (e.g., post in Facebook) to another (e.g., tweet in Twitter). To keep the paper self-contained, readers can refer to [26] for more details on social actions.

[2] "*By the end of 2013, Facebook was being used by 1.23 billion users worldwide, adding 170 million in just one year*" [20].

[3] LinkedIn's list of attributes that support automatic analysis of members' profiles is available at developer.linkedin.com/docs/fields. Examples of attributes include industry, location, specialties, and positions.

SA is a meaningful piece of information that a Web 2.0 application makes available to users and other applications (whether Web 2.0 or not). We define SA from two perspectives:

1. Operation perspective: it associates, in a straightforward manner, a social action with a SA, e.g., post and tag are directly mapped onto postSA and tagSA, respectively.
2. Data perspective: it may associate the result of executing a social action with a SA. Indeed, not all social actions' results should become SAs. While post on Facebook wall creates eventPostSA, sharing this post, that is already associated with eventPostSA, does not create a new SA. Sharing enriches the existing eventPostSA.

Like a BA, a SA includes a set of descriptive data properties (structured and/or unstructured), a set of states that reflect changes in the SA, and a life-cycle built upon these states.

With respect to the faculty hiring example, jobOpeningBA (description of the job opening) and applicationCandidateBA (application of a candidate) are BAs. And, jobOpeningPostSA and jobOpeningTweetSA, that result from executing post on Facebook and tweet on Twitter, respectively, are SAs. Any comment on Facebook will simply enrich jobOpeningPostSA and will not create any new SAs. Table 1 presents the correspondence between BAs and SAs in the context of enterprise 2.0.

Table 1. BA/SA correspondence in the context of enterprise 2.0

Correspondence type	Description
1:0	A BA is not associated with any SA, e.g., candidateSalaryNegotiationBA treated as private with no corresponding SA
1:1	A BA is associated with exactly one SA, e.g., jobOpeningBA associated with jobOpeningPostSA that exists on the university's private social network
1:n	A BA is associated with many SAs, e.g., candidateInformationBA extracted from user profiles that exist on different Web 2.0 applications like userProfileLinkedInSA and userProfileFacebookSA

3.3 Business-Data Artifact *versus* Social-Data artifact

Our vision of enterprise 2.0, thoroughly discussed in [6], advocates for bridging the gap between the business and social worlds using data artifacts. Business-data artifacts reside in the business world and social-data artifacts reside in the social world. To better understand the synergy between these artifacts/worlds, we propose the following five properties (Table 2):

1. Stakeholder refers to entities (e.g., customers and government bodies) that constitute the enterprise's ecosystem. In the business world, the identity of any stakeholder must be known. In the social world, the identity of a stakeholder might not be known, i.e., anonymity is tolerated.
2. Regulation refers to local and international policies that define and constraint the operation of the enterprise. In the business world, compliance with regulations is a must and needs to be enforced at all times. In the social world, compliance with regulations is somehow relaxed. The (physical) locations of both the enterprise and the Web 2.0 application have an impact on this compliance.
3. Environment refers to the ecosystem in which the enterprise operates. This ecosystem could be political like EU (European Union) or economical like BRIC (Brazil, Russia, India, and China). In the business world, the environment is closed by default (i.e., strict control). This is backed by the definitions of stakeholder and regulation properties in the business world. Any business-data artifact should be mapped onto a specific element that is reported in this environment's legislation. In the social world, the environment is open by default (i.e., loose control); e.g., customers enter and leave the enterprise's Facebook account without any prior notice. This is backed by the definition of stakeholder property in the social world.
4. Content refers to data and messages that the enterprise manipulates and exchanges. In the business world, the content is "well structured" although business-data artifacts handle unstructured data. By well structured, we mean that content management operations are properly defined. In the social world, the content is "loosely structured", i.e., different forms of data (e.g., text, photo, and audio files) and messages can be manipulated and exchanged over Web 2.0 applications.
5. Third-party-application refers to the control that the enterprise has over applications of third parties. In the business world, the enterprise has full control over any third-party application that it uses. For instance, it sets its functional and non-functional requirements over this application. In the social world, the enterprise has limited control over third-party applications (i.e., Web 2.0 application like Facebook). For example, the enterprise might not be able to dictate how its account on a Web 2.0 application operates.

Table 2. Properties associated with data artifacts in enterprise 2.0

Property	Business world	Social world
Stakeholder	Identity must be known	Anonymity is tolerated
Regulation	Compliance is a must	Compliance is relaxed
Environment	Closed	Open
Content	Well structured	Loosely structured
Third-party app	Full control	Limited control

3.4 Architecture

Figure 2 represents the architecture of enterprise 2.0. Each world is the host of specific platforms and/or applications for managing business processes and social activities[4]. These activities are initiated when the business processes are executed (in fact, when BAs take on specific states). Interactions between the business and the social worlds involve BAs and SAs. These interactions either are confined into the borders of a specific world or cross the worlds' borders. The interactions between the different artifacts in Fig. 2 are used for representation purposes, only. To bridge the gap between the business and social worlds, a meet-in-the-middle platform is deployed hosting Social Machines (SMs) [7]. SMs act as proxies over Web 2.0 applications so that BAs act upon SAs, when deemed necessary. Figure 2 also shows interactions between known/unknown stakeholders and business/social worlds. The interaction with the business world is for executing business processes by processing known stakeholders' requests and contacting known stakeholders. Meanwhile, the interaction between stakeholders and social world is for using Web 2.0 applications for creating accounts, signing in, posting notes, etc.

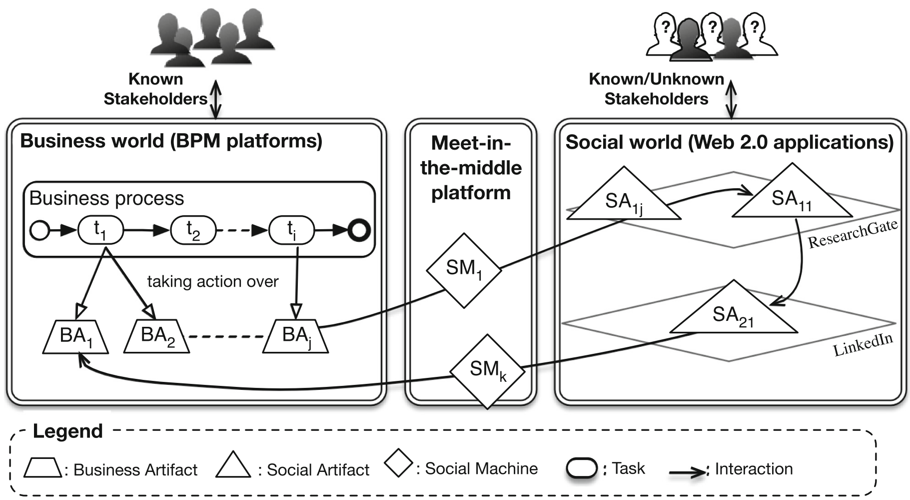

Fig. 2. Architecture of enterprise 2.0

The business world hosts the enterprise's business processes. A business process consists of tasks connected to each other through data dependencies like prerequisite and co-prerequisite. To perform a process, tasks are executed by acting upon the relevant BAs through what is usually known in the database community as Create, Read, Update, and Delete (CRUD) operations. What is

[4] Since processes are usually known for being well-formed, we avoid the term of "social processes" and use "social activities" instead. These latter are generally unstructured and unordered.

important in these operations is how a BA takes on specific states according to its respective life-cycle and how a BA, with the help of specific SMs, interacts with the social world to create new SAs, communicate with existing SAs, etc.

The social world hosts the enterprise's SAs associated with Web 2.0 applications and created as per Sect. 3.2. Some applications are internal to the enterprise (i.e., locally managed) while others are external (e.g., LinkedIn) which needs specific agreements (service level agreement) between the enterprise and these applications' providers. An enterprise can consider different vendors of Web 2.0 applications depending on the type of online social presence that it wishes to have on the Internet. We assume that each Web 2.0 application has a set of proprietary APIs that allow third parties to act on this application in terms of creating new SAs, making SAs take on new states, checking on SAs' states, and so on. Like BAs, SAs interact with others in the same or different applications and respond to BAs' queries like informing them about their current states. It is worth noting that CRUD operations are also executed over SAs.

The meet-in-the-middle platform consists of SMs that offer the necessary support for BA/SA interaction. The diversity of Web 2.0 applications (e.g. APIs, protocols, and data format) makes it quite difficult to "consume" their functionalities in a unified way. To deal with this diversity, SMs offer functionalities, exposed as a unified API, that implement these interactions between BAs and SAs. SMs, also, define a generic model for wrapping Web 2.0 applications such as Twitter, Facebook, Google, and Dropbox. Table 3 includes some SMs' functionalities.

Table 3. Social machines' functionalities

Functionality name	From: To	Description
link	B:S	Connects the business world to a platform hosting SAs in the social world
unlink	B:S	Disconnects the business world from a platform in the social world (coupled with *link*)
search	B:S	Looks for SAs according to a provided search query
list	B:S	Identifies SAs according to specific filters, e.g., all users who declined an invitation to a social event
create	B:S	Requests the establishment of a new SA
delete	B:S	Removes a SA (coupled with *create*)
update	B:S	Requests changes in the data/state of a SA
ping	B:S	Asks for details (e.g., current data values and state) of a SA
ack	S:B	Returns the current data values and state of a SA (coupled with *ping*)
subscribe	B:S	Allows the business world to subscribe to an event of interest (e.g., changing in a SA state) and be notified later
unsubscribe	B:S	Allows the business world to unsubscribe of an event of interest (coupled with *subscribe*)
notify	S:B	Notifies the business world when an event of interest occurs in response to *subscribe*

Most messages associated with these functionalities originate from the business world since it hosts the processes that drive the enterprise operation and it has control over the social activities to perform. Since the business and social worlds are associated with different platforms/applications, operation failure can occur. For the sake of simplicity we assume that all operations succeed. Similarly to enterprise portals (e.g., Alfresco and Liferay) that aim at integrating information, people, and processes across organizational boundaries through a unified framework, the meet-in-the-middle platform acts as a liaison between the business and social worlds.

3.5 Illustration

Let us use the faculty-hiring example to illustrate how a SM's functionality like *link* allows the business and social worlds to interact together. First of all, we assume that the university has accounts (usernames/passwords) registered in certain Web 2.0 applications. These accounts enable the HC staff to execute online activities and to generate some data hosted by these Web 2.0 applications. The HC staff signs in into LinkedIn and posts some job openings on the university's LinkedIn page. This is an example of a "manual" creation of posts. However, to allow an automated business process to create posts on the university's LinkedIn page, we need to understand how LinkedIn allows third-party applications to manipulate its functionalities. In practice, this involves authentication to acquire an access token, to request approvals, and to exchange authorization codes prior to calling LinkedIn's management APIs. To allow this automated authentication, a LinkedIn-SM provides a single functionality called *link* (Table 3) that establishes a "pre-authorized" communication channel between the business and social worlds. *link* is responsible for abstracting the whole authentication process.

Figure 3 shows an UML sequence diagram of this process that begins when LinkedIn-SM requests a token from LinkedIn upon a link request from the business world. Next the HC-staff is redirected to a URL along with some necessary input details like login and consent. LinkedIn handles user authentication and consent. The result is an authorization code that LinkedIn-SM exchanges for an access token, which has a limited lifetime, and a refresh token, which allows to obtain new access tokens beyond the lifetime of an access token. LinkedIn-SM then stores the refresh token for future use and uses the token to access LinkedIn APIs. As expected, this process is only performed in the first access. Technical details on the implementation of *link* are given in Sect. 4.

3.6 Examples of Social-Data Artifacts

As stated in Sect. 3.2, a SA is defined from two perspectives: operation (e.g., postSA) and data (e.g., jobOpeningPostSA). These perspectives correspond to the nature of social actions that users execute along with these actions' possible outcomes. In [26], we describe the categorization of social actions into communication, sharing, and enrichment (Table 4). A social action can simultaneously belong to more than one category but this possibility has been discarded for the sake of simplicity.

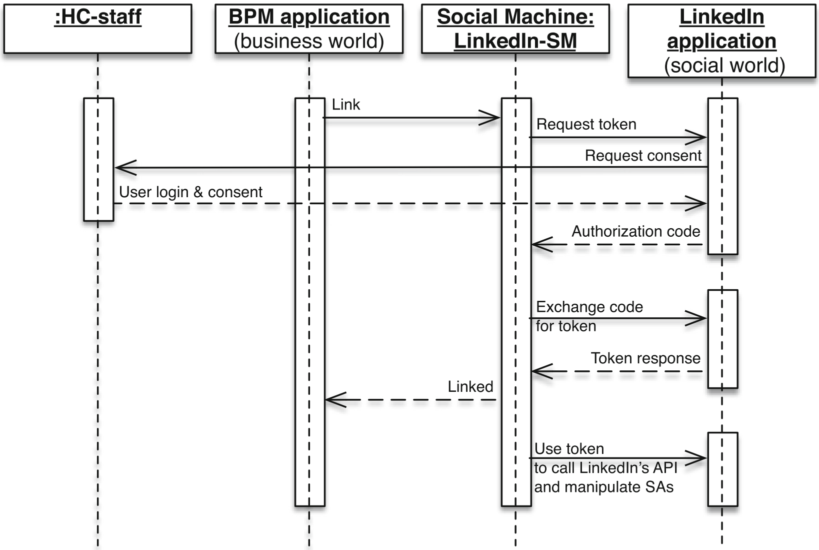

Fig. 3. Authentication using *link* functionality

Table 4. Representative categories of social actions

Category	Description	Examples of social actions
Communication	Includes actions that lead into back-and-forth interactions between users, which should help them engage in joint operations	**Chat** with a user or group of users **Poke** someone **Send** direct messages to a user's inbox **Launch** a social event/campaign
Sharing	Includes actions that allow to create and edit shared content as well as facilitate its consumption; the actions support one-way interaction	**Co-author** a text on a Wiki system **Publish** a post on a Blog Web site **Upload** a video on a public repository **Subscribe** to an RSS Feed
Enrichment	Includes actions that provide additional [meta] data on shared content in terms of providing opinions and/or ranking something	**Comment** a post **Rank** a post, page, video, news **Tag** user's photos, videos, activities **Like/Unlike** a comment or post

In this section, we present some SAs' and BAs' properties and life-cycles. Different BAs and SAs are associated with the faculty-hiring example (Fig. 4). When the university announces a job opening, jobOpeningBA is created so that the opening is tracked from a business perspective e.g., approvals and credentials. jobOpeningBA's life-cycle contains states like created, initiated, and launched. This latter state triggers a post on the university's Facebook account. This post is associated with postSA and jobOpeningPostSA. Values assigned to postSA's properties include (postTrigger: jobOpeningBA) and (postOwner: Mr. Michael from HC) while values assigned to jobOpeningPostSA's properties include postTrigger: postSA and postOwner: Mr. Michael from HC.

When a candidate accesses the university's Facebook account, he/she can apply for the opening (i.e., create candidateApplicationBA) and/or take further social actions like comment. This leads into creating commentSA and enriching jobOpeningPostSA. Figure 4 illustrates the interactions between some BAs and SAs involved in the faculty-hiring running example.

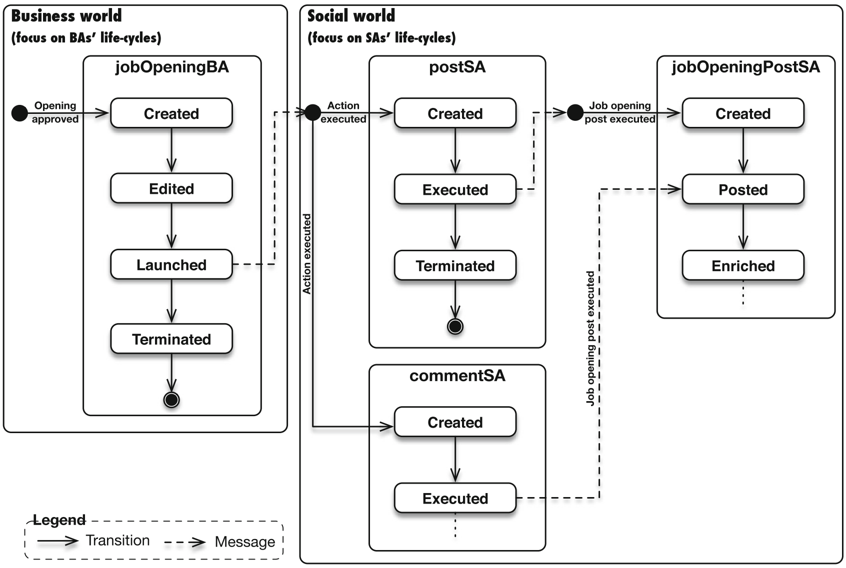

Fig. 4. Artifacts' life-cycles in interaction

4 Faculty-Hiring System Development

We discuss the development of the faculty-hiring system that allows universities to tap into the social world so they identify the right candidates with respect to faculty openings' requirements. We outline the technical architecture of the system as well as the implementation technologies and tools.

4.1 Architecture

Figure 5 illustrates the components of the architecture of the faculty-hiring system. The architecture complies with the design guidelines for enterprise 2.0

presented in [6] by having a meet-in-the-middle platform that bridges the gap between the business and social worlds. The architecture includes a *Data source layer*, a *SA management layer*, a *Business service layer*, and an *Application layer*.

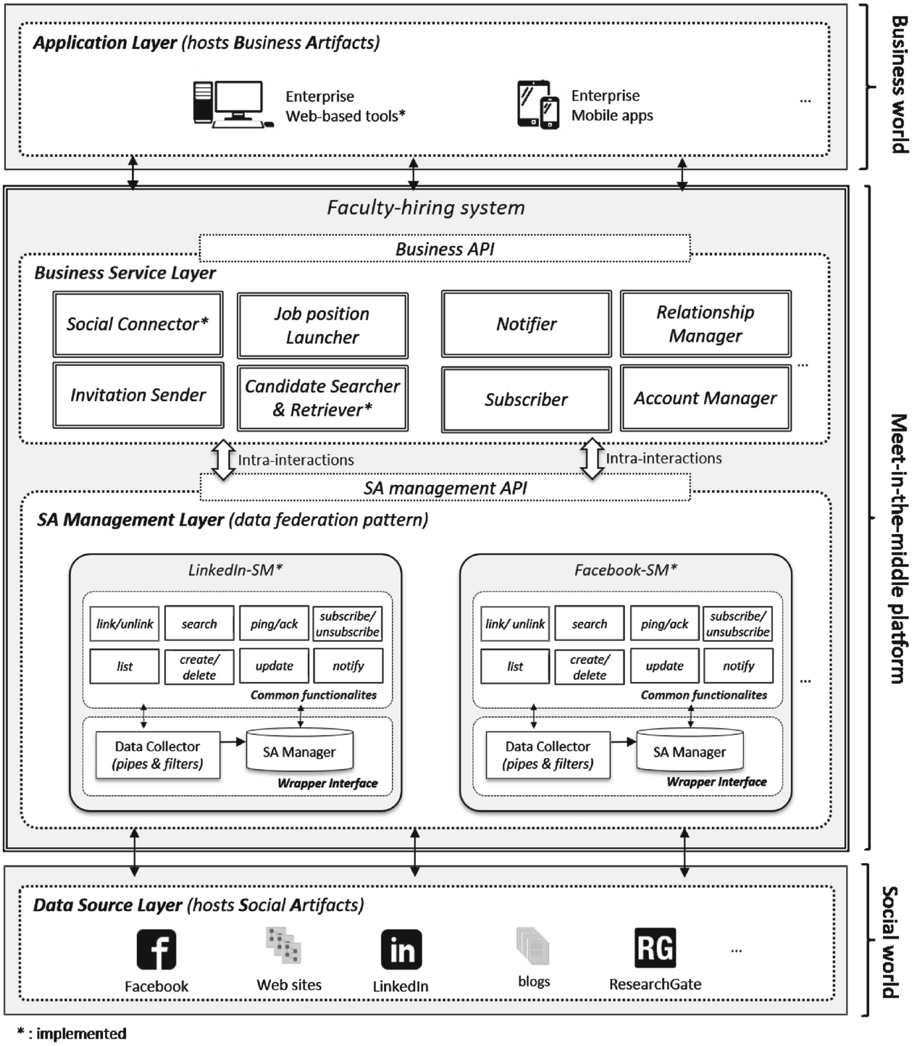

Fig. 5. Faculty-hiring system's architecture

1. *Data source layer* hosts SAs associated with job offers (e.g., jobOpeningPostSA), candidates' profiles (e.g., userProfileLinkedInSA), candidates' comments on openings, etc. SAs are distributed over different Web 2.0 applications such as LinkedIn and Facebook.

2. *SA management layer* hosts specialized social machines like LinkedIn-SM and Facebook-SM for wrapping purposes. Each SM loads, filters, and extracts data of interest from SAs that are afterwards stored in a SA Manager, according to a common SA model. On top of the SA Manager, the functionalities introduced in Table 3 are implemented and provided to the *Business service layer* through SA management APIs.
3. *Business service layer* relies on SA management APIs to define different software components (e.g., *Social connector*, *Job launcher*, and *Notifier*), which implement the business logic of the faculty-hiring system's services. Such services are externalized as REST APIs [16] to be consumed by enterprise business applications.
4. *Application layer* refers to enterprise applications, typically GUIs for the *Business service layer*, that can be developed on top of the faculty-hiring system's services. In Fig. 5, these applications are Web-based and/or mobile-based. In the current version, a Web-based application to use some faculty-hiring system's services is available (youtu.be/Fdvghukdk9w).

4.2 Implementation

Our faculty-hiring system implements the elements marked with asterisk (*) in Fig. 5. The system is implemented in Java 1.8 and deployed on a Tomcat 8.0 server. Facebook and LinkedIn are the Web 2.0 applications included in the *Data source layer* and used for hosting SAs. RESTful style [16] is, also, used to develop both the SA management and Business APIs. Such RESTful APIs are implemented with Jersey (jersey.java.net), which is the JAX-RS (jax-rs-spec.java.ne) reference implementation. The Web-based application consists of a friendly Web interface developed in JSP using Bootstrap (getbootstrap.com) framework. This Web application allows to launch the faculty-hiring system's services in a way to support message exchange between the business and social worlds. Figure 6 shows some screenshots of the developed Web-based application retrieving information on a potential candidate.

On the business world, the system considers jobOpeningBA as the main BA. Meanwhile, on the social world, we highlight some SAs such as jobOpeningPostSA, userProfileLinkedInSA, and userProfileFacebookSA. Figure 7 illustrates the SMs' inner-components in charge of collecting SAs. As shown in this figure, each SM has an internal Data collector that uses pipes and filters as an integration pattern to create the logic for both data conversion and aggregation. Internally, the Data collector is a set of interconnected components that perform tasks such as loading data from the wrapped Web 2.0 application, filtering unnecessary data, and formatting them into the corresponding SA's properties. In practice, as Web 2.0 applications usually provide data in JavaScript Object Notation (JSON, json.org) format, the main conversion taking place within the Data collector is the mapping content from JSON onto SAs and *vice-versa*. The Social Artifact Mapper is the internal component responsible for doing that and making SAs available through the SA manager.

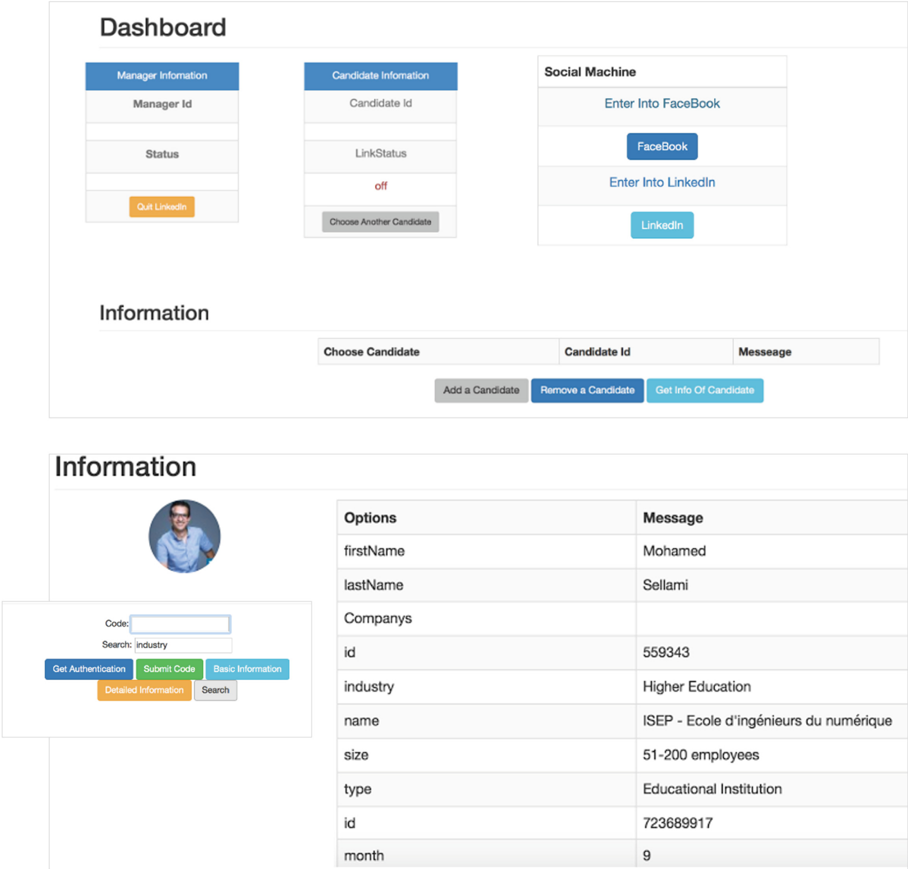

Fig. 6. Screenshots of the Web-based application

As there is a need to integrate heterogeneous SAs from existing Web 2.0 applications, the architecture design of enterprise 2.0 systems encompasses some integration issues. In this sense, it is worth noting that we use the *Data federation pattern* [23] to aggregate and correlate the necessary SAs from multiple Web 2.0 applications. The data federation is realized in the *SA management layer* by a set of parallel *pipes and filters* defined by each individual SM (Fig. 7). *SA Management API* simplifies the manipulation of SAs by abstracting their properties and life-cycles. Furthermore, some generalizations are also implemented in order to extract two or more SAs' shared characteristics and combine them into a unique generalized SA. This generalization process was fundamental to enable a SA common model. For example, candidateProfileSA abstracts both candidateProfileLinkedInSA and candidateProfileFacebookSA.

We comply with REST principles [16] to ensure the proper development of the faculty-hiring system's APIs and allow a seamless access from different

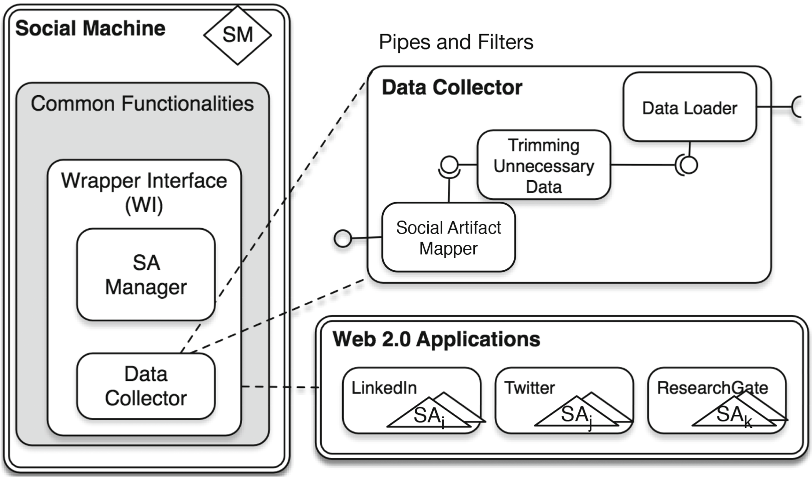

Fig. 7. Internal structure of a SM's data collector **(Z: needs to be fixed)**

platforms including legacy applications. Beyond that, widely known technologies such as JSON for data-interchange format and Open Authorization Protocol (OAuth, oauth.net) for authentication are also used in this development. For instance, *link* functionality is based on OAuth 2.0 that facilitates the establishment of "pre-authorized" and "persistent" communication channels between the business and social worlds, and allows the manipulation of SAs on behalf of the enterprise. JSON is used as a default data format in the interactions between the Web-based application and faculty-hiring system's API. To better understand how the common functionalities already described in Table 3 are implemented as RESTful APIs, Table 5 shows the HTTP method used to execute each of them.

The design and implementation of the faculty-hiring running example points out the degree of integration that can be obtained by using our approach, but also the set of new opportunities that emerge when supporting the connection of the business world and Web 2.0 applications together. One of these opportunities is the adaptation that business processes could be subject ti based on the relevant events that happen in the social world. In the aforementioned SM's functionalities (Table 5), for instance, *notify* could be used to trigger a task of interview scheduling on the business world. This happens when a candidate matches an opening position according to some requirements that could be specified in the *subscribe* functionality.

4.3 Data-Exchange Analysis

In Fig. 5, the meet-in-the-middle platform supports the data exchange between the business and social worlds. From an analysis perspective, we captured the average time and other relevant metrics during this exchange as per Table 6. Our analysis focused on the SM's *retrieve* functionality (Table 5). We assessed the

Table 5. RESTful specification of SM's functionalities

Name	HTTP request	Description
URIs relative to https://{*host*}/{sm-id}		
link	POST /link	Connects the SM to the University's LinkedIn and/or Facebook account
	POST /link/{*candidate-id*}	Connects the SM to the LinkedIn and/or Facebook account of the candidate identified by {*candidate-id*}
unlink	POST /unlink	Disconnects the SM to the University's LinkedIn and/or Facebook account
search	GET /search?*q=query*	Returns candidateProfileSAs related to the search query
list	GET /{*opening-id*}	Gets candidateProfileSAs suitable to the jobOpeningPostSA identified by {*opening-id*}/match
	GET /{*opening-id*}/viewed	Lists candidateProfileSAs that have viewed University's posting on LinkedIn or Facebook about the job opening position identified by {*opening-id*}
	GET /{*opening-id*}/applied	Lists candidateProfileSAs that applied to a job opening position identified by {*opening-id*}
create	POST /{*opening-id*}	Requests the creation of a new jobOpeningPostSA identified by {*opening-id*}
delete	DELETE /{*opening-id*}	Requests the removal of the jobOpeningPostSA identified by {*opening-id*}
update	PUT /{*opening-id*}	Requests changes in the data and/or state of the jobOpeningPostSA identified by {*opening-id*}
retrieve	GET /retrieve/{*id*}	Retrieves basic candidateProfileSA of the candidate identified by {*id*}
	GET /retrieve/{*id*}/detailed	Retrieves detailed candidateProfileSA of the candidate identified by {*id*}
ping	GET /{*opening-id*}	Gets details (e.g., current data values and state) of the jobOpeningPostSA identified by {*opening-id*}
ack	HTTP RESPONSE CODE:200 {*data*}	Returns the current data values and state of a pinged jobOpeningPostSA
subscribe	POST /subscribe/{*topic*}	Subscribes to the *topic* of interest
unsubscribe	POST /unsubscribe/{*topic*}	Unsubscribes to the *topic* of interest
notify	POST {*callbackURL*}	Notifies the business world, via a previously informed callback URL, when an event of interest occurs

response time between our faculty-hiring system and both LinkedIn and Facebook. Retrieving the two types of candidate profiles (i.e., basic and detailed) from Facebook and LinkedIn were simultaneously intercepted. A total of 15,288 requests were tracked over a period of time of 70 min, i.e., 8:40 am–9:50 am (during the regular working hours for an University) with a data calculation every second.

Table 6. Performance analysis of retrieving candidate profiles

	Facebook		LinkedIn	
	Basic profile	Detailed profile	Basic profile	Detailed profile
Number of requests	3809	3824	3828	3827
Average (ms)	273.34	273.63	474.12	480.02
Minimum (ms)	255	255	268	300
Maximum (ms)	1229	1264	4559	4432

From Table 6, we notice that for the same Web 2.0 application, the difference between the average response time for retrieving basic profile and retrieving detailed profile is very small (e.g., Facebook basic profile: 273.34 ms and Facebook detailed profile: 273.63 ms). However, we could realize that LinkedIn presented almost twice the average time assessed with Facebook (e.g., Facebook detailed profile: 273.63 ms; LinkedIn detailed profile: 480.02 ms). Part of this difference is explained by the average amount of data exchanged to get each kind of profile (Table 7).

Table 7. Average amount of data (in Byte) exchanged by each candidate profile

	Facebook	LinkedIn
Basic profile	196 B	284 B
Detailed profile	295 B	500 B

5 Future work

Our future work revolves around 4 initiatives: social query, social-content protection, artifact management, and semantic social machines.

1. In the social-query initiative, we would like to define a language for running queries over social-data artifacts like suggested in [28]. Users could be interested in analyzing the content of some social-data artifacts and identifying patterns between them, for example.

2. In the social-content protection initiative, we would like to address the reluctance of enterprises to embrace Web 2.0 applications. Signs of this reluctance include lack of control over the content posted on these applications and these applications' open nature since members sign up and sign off without prior notice. To motivate the business world's stakeholders embrace Web 2.0 applications, we could think of putting restrictions on social-data artifacts' life-cycles by either enabling or disabling some states.
3. In the artifact-management initiative, we would like to look into ways of applying the five processes of a successful contact center to social-data artifacts [33]. These processes are listening (how to get access to social-data artifacts across various Web 2.0 applications?), funneling (how to find the rights social-data artifacts in response to certain needs?), routing (how to direct relevant social-data artifacts to appropriate stakeholders?), engaging (how to provide stakeholders with the necessary means to track social-data artifacts?), and recording (how to measure the value-added of social-data artifacts to the enterprise operation?).
4. Finally, the semantic social machines initiative would examine the role of ontologies in addressing interaction conflicts that could arise between the business and social worlds. Different terminologies could be used for expressing data and events and hence, consistency would be deemed necessary during interaction.

6 Conclusion

In this paper, we designed and implemented an approach that brings the business and social worlds together. Each world has its priorities and concerns. The approach relies on business-data artifacts, social-data artifacts, and meet-in-the-middle platform. Business-data artifacts capture activities related to business processes, whilst social-data artifacts abstract activities executed over Web 2.0 applications. The meet-in-the-middle platform supports interactions between artifacts through social machines. A running example for faculty hiring was used to illustrate some data artifacts such as jobOpeningBA as a business-data artifact and jobOpeningPostSA as a social-data artifact. Each artifact is a collection of related data (structured and/or unstructured) and has a life-cycle. A comparative analysis between business- and social-data artifacts was also included in this paper using five properties, namely stakeholder, regulation, environment, content, and third-party application.

References

1. Badr, Y., Narendra, N.C., Maamar, Z.: Business artifacts for e-Business interoperability. In: Kajan, E., (ed.) Electronic Business Interoperability: Concepts, Opportunities and Challenges. IGI Global Publishing (2011)

2. Berardi, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Mecella, M.: A foundational vision for e-Services. In: Proceedings of Ubiquitous Mobile Information and Collaboration Systems Workshop (UMICS 2003) held in conjunction with The 15th International Conference On Advanced Information Systems Engineering (CAiSE 2003), Klagenfurt/Velden, Austria (2003)
3. Blinn, N., Lindermann, N., Fäcks, K., Nüttgens, M.: Web 2.0 Artifacts in SME-Networks a qualitative approach towards an integrative conceptualization considering organizational and technical perspectives. In: Proceedings of Software Engineering 2009 (Workshops), Kaiserslautern, Germany (2009)
4. Brambilla, M., Fraternali, P., Vaca, C.: A notation for supporting social business process modeling. In: Proceedings of the Fourth Workshop on Business Process Management and Social Software (BPMS2 2011) held in conjunction with the Seventh International Conference on Business Process Management (BPM 2011), Lucerne, Switzerland (2011)
5. Bruno, G., Dengler, F., Jennings, B., Khalaf, R., Nurcan, S., Prilla, M., Sarini, M., Schmidt, R., Silva, R.: Key challenges for enabling agile BPM with social software. J. Softw. Maintenance Evol. Res. Pract. **23**(10), 297–326 (2011)
6. Burégio, V.A., Maamar, Z., Meira, S.L.: An architecture and guiding framework for the social enterprise. IEEE Internet Comput. **19**(1), 64–68 (2015)
7. Burégio, V.A., Meira, S.L., Rosa, N.S., Garcia, V.C.: Moving towards "Relationship-aware" applications, services: a social machine-oriented approach. In: Proceedings of the EDOC 2013 Workshops held in conjunction with the 17th IEEE International Enterprise Distributed Object Computing Conference (EDOCW 2013), Vancouver, BC, Canada (2013)
8. Busch, P., Fettke, P.: Business process management under the microscope: the potential of social network analysis. In: Proceedings of the 44th Hawaii International Conference on System Sciences (HICSS 2011), Kauai, HI, USA (2011)
9. Cross, R., Gray, P., Cunningham, S., Showers, M., Thomas, R.J.: The collaborative organization: how to make employee networks really work. MITSloan Manag. Rev. **51**(1), 83–90 (2010). Fall
10. Damaggio, E., Hull, R., Vaculín, R.: On the equivalence of incremental and fixpoint semantics for business artifacts with guard-stage-milestone lifecycles. Inform. Syst. **38**(4), 561–584 (2013)
11. Dengler, D., Koschmider, A., Oberweis, A., Zhang, H.: Social software for coordination of collaborative process activities. In: zur Muehlen, M., Su, J. (eds.) BPM 2010. LNBIP, vol 66, pp. 396–407. Springer, Heidelberg (2010). doi:https://doi.org/10.1007/978-3-642-20511-8_37
12. Draheim, D.: Social BPM, chapter Smart Business Process Management. Future Strategies (2012)
13. Dumas, M., Hull, R., Patrizi, F., Editorial, G.: Special issue on data and artifact-centric business processes. Comput. **98**(4), 343–344 (2016)
14. Erol, S., Granitzer, M., Happ, S., Jantunen, S., Jennings, B., Koschmider, A., Nurcan, S., Rossi, D., Schmidt, R., Johannesson, P.: Combining BPM and social software contradiction or chance. J. Softw. Maintenance Evol. Res. Pract. **22**(6–7), 449–476 (2010)
15. Faci, N., Maamar, Z., Kajan, E., Benslimane, D.: Research roadmap for the enterprise 2.0 issues & solutions. Scientific Publications of the State University of Novi Pazar Journal, Ser. A: Appl. Math. Inform. & Mechanics **6**(2), 81–89 (2014)
16. Fielding, R.T.: Architectural styles and the design of network-based software architectures. Ph.D. thesis, University of California, Irvine (2000)

17. Gonzalez, P., Griesmayer, A., Lomuscio, A.: Verifying GSM-based business artifacts. In: Proceedings of the 19th IEEE International Conference on Web Services (ICWS 2012), Honolulu, HI, USA (2012)
18. Grim-Yefsah, M., Rosenthal-Sabroux, C., Thion, V.: Using information of an informal network to evaluate business process robustness. In: Proceedings of the International Conference on Knowledge Management and Information Sharing (KMIS 2011), Paris, France (2011)
19. Houston, J., Hoehler, J.: The Social Enterprise: What it means to be a Social Enterprise, Deloitte Digital (2013). http://deloitteblog.co.za/wp-content/uploads/downloads/2013/04/Enterprise-2.0.pdf
20. Kiss, J.: Facebook's 10th Birthday: From College Dorm to 1.23 Billion Users (2014). http://www.theguardian.com/technology/2014/feb/04/facebook-10-years-mark-zuckerberg
21. Kotlarsky, J., van Fenema, P.C., Willcocks, L.P.: Developing a knowledge-based perspective on coordination: the case of global software projects. Inform. Manag. **45**(2), 96–108 (2008)
22. Kumaran, S., Liu, R., Wu, F.Y.: On the duality of information-centric and activity-centric models of business processes. In: Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE 2008), Montpellier, France (2008)
23. Liu, Y., Liang, X., Xu, L., Staples, M., Zhu, L.: Using architecture integration patterns to compose enterprise mashups. In: Proceedings of the 8th Working IEEE/IFIP Conference on Software Architecture (WICSA 2009) and the 3rd European Conference on Software Architecture (ECSA 2009), Cambridge, UK (2009)
24. Maamar, Z., Badr, Y., Narendra, N.C.: Business artifacts discovery and modeling. In: Proceedings of the 8th International Conference on Service Oriented Computing (ICSOC 2010), San Francisco, California, USA (2010)
25. Maamar, Z., Burégio, V., Sellami, M.: Collaborative enterprise applications based on business and social artifacts. In: Proceedings of the 24th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2015), Larnaca, Cyprus (2015)
26. Maamar, Z., Burégio, V.A., Faci, N., Benslimanem, D., Sheng, Q.Z.: "Controlling" Web 2.0 applications in the workplace. In: Proceedings of the 2015 IEEE 19th International Conference on Enterprise Distributed Object Computing (EDOC 2015), Adelaide, Australia, September 2015
27. Masli, M., Geyer, W., Dugan, C., Brownholtz, B.: The design and usage of tentative events for time-based social coordination in the enterprise. In: International World Wide Web Conference, Hyderabad, India (2011)
28. Montemayor, J., Diehl, C.P., Query, S.: Looking for social signals from online artifacts. Johns Hopkins APL Tech. Digest **30**(1), 41–46 (2011)
29. Nandi, P., Kumaran, S.: Adaptive business objects - a new component model for business integration. In: Proceedings of the International Conference on Enterprise Information Systems (ICEIS 2005), Miami, Florida, USA (2005)
30. Narendra, N.C., Badr, Y., Thiran, P., Maamar, Z.: Towards a unified approach for business process modeling using context-based artifacts and web services. In: Proceedings of the 2009 IEEE International Conference on Services Computing (SCC 2009), Bangalore, India (2009)
31. Nigam, A., Caswell, N.S., Artifacts, B.: An approach to operational specification. IBM Syst. J. **42**(3), 415–432 (2003)

32. OpenKnowledge. Social Business Process Reengineering. Technical report (2012), http://socialbusinessmanifesto.com/social-business-process-reengineering (checked out in May 2016)
33. Oracle. Is Social Media Transforming Your Business? An Oracle White Paper, March 2012. http://www.oracle.com/us/products/applications/social-trans-bus-wp-1560502.pdf. visited June 2013
34. Popova, V., Fahland, D., Dumas, M.: Artifact lifecycle discovery. Int. J. Coop. Inform. Syst. **24**(1), 1–44 (2015)
35. Rito Silva, A., Meziani, R., Magalhães, R., Martinho, D., Aguiar, A., Flores, N.: AGILIPO: embedding social software features into business process tools. In: Proceedings of the Third International Workshop on the Business Process Model and Notation (BPMN 2010), Ulm, Germany (2010)
36. Ryndina, K., Malte Küster, J., Gall, H.: Consistency of business process models and object life cycles. In: Proceedings of Models in Software Engineering Workshops (MoDELS 2006), Genova, Italy (2006)
37. Strosnider, J.K., Nandi, P., Kumaran, S., Ghosh, S., Arsanjani, A.: Model-driven synthesis of SOA solutions. IBM Syst. J. **47**(3), 415–432 (2008)
38. Warr, W.A., Software, S.: Fun and games, or business tools? J. Inform. Sci. **34**(4), 591–604 (2008)
39. www.lightwaveonline.com. Web 2.0 Companies Likely Shape Optical Network Market, March 2015. www.lightwaveonline.com/articles/2015/03/web-20-companies-likely-shape-optical-network-market-says-lightcounting.html (checked out in May 2016)
40. www.strategyr.com. Global Enterprise Web 2.0 to Reach US$5.7 Billion by 2010, June 2015. www.strategyr.com/PressMCP-6116.asp (checked out in May 2016)

http://www.springer.com/978-3-662-56120-1

Transactions on Large-Scale Data- and
Knowledge-Centered Systems XXXV
Hameurlain, A.; Küng, J.; Wagner, R.; Sakr, S.; Razzak, I.;
Riyad, A. (Eds.)
2017, IX, 133 p. 31 illus., Softcover
ISBN: 978-3-662-56120-1