A framework for Enterprise Social Computing: Towards the Realization of Enterprise 2.0

Yousef Baghdadi
Department of Computer Science
Sultan Qaboos University, Muscat, Oman
ybaghdadi@squ.edu.om

Zakaria Maamar
College of Information Technology
Zayed University, Dubai, UAE
zakaria.maamar@zu.ac.ae

Abstract—Newer technologies, termed as Web 2.0 and Social Computing (SC) facilitate online social interactions. However, the broad impact of SC in diverse domains and the complexity of features that span diverse disciplines poses new multi-disciplinary research challenges. Among these challenges, we are interested in a reference model for enterprise interactions that drives an architecture, which in its turn guides a deployment method for enterprise SC Applications (SCAs). Systematically integrating SCAs with Service-Oriented Architecture (SOA) and Web Services (WSs) will move enterprise architecture towards Enterprise 2.0; and propose an interaction reference model and an interaction architecture that guides a systematic realization of Enterprise 2.0.

Keywords—Enterprise Interactions; SNs; Web 2.0; Enterprise Social Computing; Web services; SOA; Enterprise 2.0

I. INTRODUCTION

The nature of the enterprise and the way people work is changing rapidly by the enabling power and competitive advantage of new social and participative technologies. Indeed, newer technologies, termed as Web 2.0, online communities, and SC facilitate collective action and online social interactions [1]. SC, concerned with the study of social behavior and social context using computational systems, has been a popular topic in many different areas [2]. SCAs enable creation, modification, and storage of content generated from interactions, and leverage that content to provide value [3]. According to Gartner, “SC is expected to impact almost every role at every kind of company.” However, the broad impact of SC in diverse domains and the complexity of its features that span diverse disciplines poses new research challenges [4].

Among these challenges, we are interested in a reference model for enterprise interactions that drives architecture, which in its turn guides a deployment method for enterprise SCAs. Indeed, in the one hand, SCAs have more peculiarities in the enterprise context than most SNs for individuals as they differ in many aspects, namely in terms of purposes, characteristics, building blocks (interactions, content, participants, and communities), architectures, and management. In the other hand, they need to build on and integrate with existing enterprise architecture.

We argue that, if systematically set, SCAs associated with Service-Oriented Architecture (SOA) and Web Services (WSs) will move enterprise architecture towards Enterprise 2.0, “a web of interconnected applications and services” as defined by [5]. Enterprise 2.0 is expected to support a web (rather than a hierarchy) of interacting participants, which will change the way enterprises are doing business by fostering creativity and innovation on processes, products, and services through the knowledge that emerges from participant interactions. Indeed, this organization of participants, supported by Enterprise 2.0, aims at: (i) building external relationships, and (ii) improving internal collaboration.

To develop SCAs in a systematic way, we propose a framework for their development and integration. It comprises of reference model for enterprise social interactions that guides a concrete architecture, for SCAs, geared towards the realization of Enterprise 2.0.

The remainder paper is as follows: Section 2 presents a background. Section 3 presents a reference model for enterprise interactions showing the main entities. Section 4 presents a concrete architecture. Section 5 presents realization of the concrete towards Enterprise 2.0. Section 6 relates closer work. Finally, a conclusion section presents further development.

II. ENTERPRISE INTERACTIONS AND SCAs

A. Interactions

Simply put, an interaction is an event that involves two objects (e.g., participants) that act in re-action to one other action.

Interactions are interesting from the perspective of knowledge emerging in terms of information, process, and expertise that are used to improve the processes, products and services provided to customers.

B. Interactions and Social Networks (SNs)

SNs allow individuals to interact by exchanging information for many purposes. Newer technologies termed as Web 2.0 are the commonly used SCAs to interconnect people through Internet. They facilitate collective action and social interactions with rich exchange of multimedia information. However, these kinds of SCAs are for individuals. The enterprise interactions have different context and purposes.

C. The diverse enterprise interactions

Enterprises engage in interactions with peers from different origins, in different locations, and with different profiles and interests. These interactions involve, as shown in
Figure 1, participants such as employees, consumers, suppliers, competitors, stakeholders, and regulating authorities that are implicitly involved in the processes, products and services. We build on [6] to categorize them into:

- **Customer-centric interactions** would disseminate information on processes, products and services and, hence, boost sales.
- **Supplier-centric interactions** would establish the reliability of a supplier, as perceived by other parties (e.g., customers, suppliers) that have dealt with this particular supplier in the past.
- **Partner-centric interactions** would identify the individuals and groups with which it can join forces to tackle complex initiatives.
- **Competitor-centric interactions** would work with competitors and partners of competitors.
- **Employee-centric interactions** expose the enterprise internal expertise and knowledge.

![Diagram of enterprise interactions](Image)

Figure 1. Types of enterprise interactions (I-T: Interaction Type)

Therefore, an enterprise should think about the necessary mechanisms to deploy specialized SCAs with respect to the nature of each type of interactions; and measure their contributions to enterprise development [7]. Indeed, having the correct SCAs in place enables knowledge emerging and contributes to:

- Locate expertise/knowledge: utilize SCAs to build internal communities and improve the visibility and discoverability of expertise and knowledge within the enterprise
- Improve process efficiency: utilize SCAs to enable employees to focus on higher value-added activities and streamline business processes
- Gain competitive advantages: utilize SCAs to use and leverage the wisdom of the crowd (employees, customers, partners, suppliers, and their acquaintance) to: (i) identify creative processes, product services and innovations, (ii) participate in solving business problems
- Support remote flexible working: utilize SCAs to make distances and time transparent enabling internal and external individuals and communities to remotely co-create value
- Reuse the existing knowledge: utilize SCAs to capture hidden knowledge within individuals and communities

The commonly used SCAs (aka Web 2.0), though useful, may not be the right SCAs to fulfill to the enterprise interaction requirements, due to their drawbacks (e.g., lack of editorial control, quality, credibility, content analysis and management) [8].

We do need an abstract framework centered on the enterprise interactions as main entities.

### III. REFERENCE MODEL FOR SCAs

Although, SCAs inherit from technologies such as Web (applications and services), SOA, databases, multimedia, and agent, they have their own specific peculiarities, requirements as shown in Figure 3 (an arrow represents an inheritance, a circle is a single- or multi-valued characteristic), motivations, and goals. Therefore, to build SCAs, we need a systematic approach constrained by a concrete architecture showing the components, the connectors, and the constraints. The concrete architecture is an architecture shared by a class of applications having the same peculiarities. The concrete architecture is itself guided by an abstract framework that plays the role of a reference model. It also accounts for specific requirements, motivations and goals while considering the existing standards specifications, protocols, and related architectures.

A reference model is an abstract framework showing the main entities of an environment, the relationships between the entities, and the constraints (if any). It is used to derive a concrete architecture for SCAs. Our model is inspired from Anderson’s model [9] used in e-learning.

#### A. Main entities: Building blocks

The main entities of the enterprise interaction environment that constitute the entities of our reference model are the building blocks used to develop enterprise SCAs. These are: Participant (Enterprise, or Individual), Content Community, and Interaction (event-actions-reactions).

1. **Participant**, a participant may be the enterprise itself or an individual, i.e., moral or physical.
   - **Enterprise** is the focal point that interacts with all the individuals. It is also responsible for the management of the content.
   - **Individual** may be an employee, a customer, a partner, a supplier, or a competitor who, by her actions/reactions, affects the enterprise processes, products and services. Her actions/reactions are reflected in the enterprise content.

2. **Content** is any type of multimedia information (e.g., texts, photos, graphics, audio, videos) that mainly relates to the enterprise processes, products, and services; and is shared by the participants.

3. **Community** is a group of participants sharing the same interest (e.g., process, product, services).

4. **Interaction** is an event that involves two objects (e.g., individual, enterprise, community, content) that act in re-action to each other action.
It is worth noting that individuals and communities have their own profile.

Figure 2. A reference model for enterprise interactions

B. Relationships between entities

The entities interact with each other according to three types of relationships as shown in Figure 2; we refer to as Content-Centered Interactions (CCI), Participant-Centered Interactions (PCI), and Self-Centered Interactions (SCI). The types of interactions include the Search/Link/Author/Tag/Extend/Signal (SLATES) operations, and Create/Retrieve/Update/Delete (CRUD) operations.

Type 1: Content-Centered Interactions (CCI)

CCI-1: Enterprise-Content. This type of interactions focus on the control and monitoring of content related to the processes, products, services, advertising, and policies. It allows the enterprise to continuously monitor and update the content and activities it creates for the sake of individuals and communities.

CCI-2: Individual-Content. This type of interactions constitute the required interactive content that responds to the individual behavior (e.g., feedbacks, claims, requests, interests, etc.) with respect to what it expects in terms of quality of processes, products and services the enterprise offers.

CCI-3: Community-Content. This type of interactions with different types of participants may lead to an emergent network of participants that is not planned by any of the participants, but which emerges from the individual interactions. The community emergence behavior is a consequence of the content change. In turn, this affects the community interactions. Thus, the content emerges from the community behavior, and vice-versa.

Type 2: Participant-Centered Interactions (UCI)

PCI-1: Enterprise-Individual. Most knowledge emerges from this type of interactions. Enterprise is encouraged to make most of information about its processes, products and services transparent and eligible for individual reactions. Therefore, enterprise should support online, real-time interactions in a large number of varieties and formats that include asynchronous and synchronous communication using all the types of multimedia.

PCI-2: Enterprise-Community. The role of the enterprise is less significant in this kind of interactions. It does not affect in a large scale the communities. However, interactions are initiated at a large scale by the enterprise, where enterprise may shape its context, processes, products and services.

PCI-3: Individual-Community. Though interactions are initiated by the enterprise, some networks (communities) of individuals will emerge from these interactions for specific need, where individuals affect and are affected by any changes. The enterprise could take profit of the emerging communities.

Type 3: Self-Centric Interactions (SCI)

SCI-1: Participant-Participant. This type of interactions allows participant to interact in a peer-2-peer basis to exchange content about their expectations in terms of quality of processes, products and services.

SCI-2: Content-Content. This type of interactions allows the content to automatically update and refresh under certain circumstances. For instance, when the process changes, some services will altered, create, or dropped. Content-content interactions are also necessary to provide a means of facilitating tracking of the use of content by diverse groups of participants.

SCI-3: Community-Community. Diverse communities may emerge from interactions; and these communities will interact with each other for a specific need. It is common to link together different services provided by different communities.

C. Constraints

The enterprise interactions differ from SNs. They are constrained by policies such as the control of the content, participants, and communities. They also concern with the
grant given to the participant and the enabling of actions on the content.

These types of interactions could not be possible neither with traditional applications nor with Web applications and their architectures. These applications do not consider interactions and participants as building blocks, whereas SCAs are both interaction and participant-centric.

Therefore, SCAs need a new concrete architecture that is guided by an abstract framework: the reference model.

IV. CONCRETE ARCHITECTURE FOR SCAs

The developed reference model guides the concrete architecture. Yet the concrete architecture accounts for specific requirements, challenges, motivations and goals, while considering the existing standards specifications, protocols, and related architectures.

A. Challenges of SCAs

SCAs development and deployment has new challenges in addition to those of the traditional and web application development. These are:

- Context: this considers the nature of users and usability with varying levels of comfort with such SCAs. SCAs need to properly work and adopt with existing applications and tools that constitute the context of the users. Achieving this kind of integration and consistency is critical, especially when users have varying degrees of comfort and familiarity with the tools and practices of SCAs.

- Governance and policies: this includes security, privacy, visibility, and custody of content. SCAs need to balance transparency and privacy of the enterprise and the individuals by providing personal and enterprise-level security settings around access and visibility of data.

- Manageability: this includes integration, extensibility, and customization. SCAs add new layers of complexity to the existing content management, security, performance, and user support, which challenges SCAs integration/insulation to (from) the existing systems.

B. Characteristics and requirements of SCAs

SC applications have their own specific characteristics and requirements as shown in Figure 3. Indeed:

- Interactions and collaboration are online and dynamic with continual refinement of the content

- Content co-creation and sharing is bottom-up community rather than top-down, capitalizing on the wisdom of crowds

- Interactions are real time, distributed, and transparent to distances and time

- Emergent, easy to learn and use (e.g., navigable), personal, rewarding: enabling people to represent themselves and how they transparently interact with the network (in a peer-2-peer communication) to discover one another and form their own communities organically, with social incentives (recognition, reputation) for creating acting on the content

- Privacy and easy handling of malicious users

![Figure 3. Peculiarities of SCAs](image)

C. Principles of SCAs

Table 1 summarizes the SCAs principles that are adapted from the 10 principles of Dron in [10].

<table>
<thead>
<tr>
<th>PN</th>
<th>Principle</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Adaptability</td>
<td>Use open standards, and where possible build as open source to adapt and evolve to suit more needs</td>
</tr>
<tr>
<td>P2</td>
<td>Evolvability</td>
<td>SCA whose structure is not fixed, i.e., that can change</td>
</tr>
<tr>
<td>P3</td>
<td>Parcellation</td>
<td>In SCAs, participants, communities and content emerge rather than imposed</td>
</tr>
<tr>
<td>P4</td>
<td>Trust</td>
<td>SCAs provide the necessary trust</td>
</tr>
<tr>
<td>P5</td>
<td>Stigmergy</td>
<td>In SCA, signs are used to guide but not to constrain</td>
</tr>
<tr>
<td>P6</td>
<td>Context</td>
<td>SCAs are only a part of a greater whole</td>
</tr>
<tr>
<td>P7</td>
<td>Constraint</td>
<td>SCAs are aware of the constraints</td>
</tr>
<tr>
<td>P8</td>
<td>Sociability</td>
<td>SCAs enable social presence and communications</td>
</tr>
<tr>
<td>P9</td>
<td>Connectivity</td>
<td>SCAs where nothing should exist in isolation, everything should influence everything else</td>
</tr>
<tr>
<td>P10</td>
<td>Scale</td>
<td>Large SCAs should arise out of the small in an endless iterative cycle of renewal</td>
</tr>
</tbody>
</table>

D. Specialization of SCAs

SCAs are specialized with respect to the nature of each type of interactions.

T1. Customer-centric SCA should state the nature of relationships between an enterprise’s customer and the acquaintances of this customer in the network, such as friendship and kinship relationships. SCA should
provide customers with access help beyond traditional means.

T2. Supplier-centric SCA should state the nature of relationships between an enterprise’s supplier and the acquaintances of this supplier in the network, such as supplier-of-supplier and competition.

T3. Partner-centric SCA should state the nature of relationships between an enterprise’s partner and the acquaintances of this partner in the network, such as add-on and alliance. SCA should enable partners to improve the delivery of the enterprise products/services.

T4. Competitor-centric SCA should state the nature of relationships between an enterprise’s competitor and the acquaintances of this competitor in the network, such as spontaneous-collaboration and competitor-of-competitor.

T5. Employee-centric SCA should state the nature of relationships between the entries and its employee in order to enhance employee engagement and create higher quality deliverables by improving accessibility to the organization combined knowledge and expertise, while enabling employee to manage how and when they consume content.

The aforementioned enterprise interaction model and the characteristics and principles of enterprise SCAs guide their concrete architecture.

E. Concrete architecture defined

The concrete architecture specifies main components, connectors, and operations. Table 2 summarizes how the concrete architecture derives from the reference model.

Components

The components of the architecture are:

1. Participant: a participant may be an individual or the enterprise.
2. Content along with: (i) a content server that manages the content co-created and shared by the participants and (ii) an agent server that controls the automatic updates of the content and connections
3. Intelligent agent

Connectors

The aforementioned components are connected through the following interfaces that are specific to each kind of communities. These are specific SNs.

1. A content server provides some generic features for actions and reactions. These are SLATES and CRUD operations.
2. An integrator to integrate the different applications: this is the right room for SOA.

Actions/Reactions

1. These are SLATES and CRUD operations

<table>
<thead>
<tr>
<th>Reference Model: Concepts</th>
<th>Concrete Architecture: Components and Connectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>Participant</td>
</tr>
<tr>
<td>Content</td>
<td>Content Server</td>
</tr>
<tr>
<td>Intelligence Agents</td>
<td></td>
</tr>
<tr>
<td>Community</td>
<td>SNs + SOA</td>
</tr>
</tbody>
</table>

T2. Supplier-centric SCA

T3. Partner-centric SCA

T4. Competitor-centric SCA

T5. Employee-centric SCA

E. Concrete architecture defined

The concrete architecture specifies main components, connectors, and operations. Table 2 summarizes how the concrete architecture derives from the reference model.

Components

The components of the architecture are:

1. Participant: a participant may be an individual or the enterprise.
2. Content along with: (i) a content server that manages the content co-created and shared by the participants and (ii) an agent server that controls the automatic updates of the content and connections
3. Intelligent agent

Connectors

The aforementioned components are connected through the following interfaces that are specific to each kind of communities. These are specific SNs.

1. A content server provides some generic features for actions and reactions. These are SLATES and CRUD operations.
2. An integrator to integrate the different applications: this is the right room for SOA.

Actions/Reactions

1. These are SLATES and CRUD operations

V. REALIZATION ARCHITECTURE: TOWARDS ENTERPRISE 2.0

The physical architecture shown in Figure 5 is one possible realization of the concrete architecture shown in Figure 4 and specified in Table 2. This implementation of the concrete architecture will lead to Enterprise 2.0. As architecture, it constitutes of components and connectors.

A. Components

The components of the physical architecture reflect the building blocks of the SCAs:

- Existing enterprise applications and specialized SCAs
- Content: structured as well as unstructured
- Participant profiles that constitute a repository of information about the participants
- A content server that manages the multimedia information created and shared between users as well as content from the external.
- A participant profile server that manages profiles
- A participant connections server that stores and manages the interactions between users, including the logs of events and actions
- An agent server that controls the automatic updates of the content, profiles, and connections

![Figure 4. Instantiation of a concrete architecture](image-url)
B. Connectors

The components of the architecture are connected through interfaces. These interfaces are provided by the different servers and may be either unified or specialized. Indeed, a content manager may have an interface that differs from the one provided to other participants to manage their profiles or to create community. The following interfaces are required:

I1. An enterprise unified interface that allows accesses to diverse applications, including SCAs
I2. Enterprise Service Bus (ESB) to enable SOA [12]
I3. Content Common interface presenting common features of the content server
I4. Existing application interfaces (e.g., resources interfaces)

Each of these interfaces should:
• Facilitate participant actions; and make them as easy as possible and encourage input
• Allow a participant to have a direct benefit from his actions
• Allow participant actions to benefit the community
• Allow re-action of other participants to any actions

VI. RELATED WORK

Many of the work relate to SNs geared towards individual interactions, where various types of SCAs provide services to individuals; but little work is concerned with enterprise interactions and enterprise SCAs. Most of the work about interactions is done in education for e-learning [9]. The authors in [6] question whether enterprises could capitalize on their SNs; and provide taxonomy for enterprise interactions. The authors in [4] have developed a model for research issues in Social Computing and sketched out how enterprises enter into SC. In [11], the authors attempted to show how to apply SNs. In [12,13], the authors discuss how to realize SOA with Web services within enterprise architecture.

VII. CONCLUSION

This work has developed comprehensive approach to enterprise SCAs. First, it showed how an enterprise could extend and use existing SNs into SCAs in order to enhance its external and internal interactions. Next, it categorized and characterized enterprise SCAs with their specifics that differ from traditional and web applications, namely in terms of requirements, principles, and building blocks. This has led to reference model that drives a concrete architecture for enterprise SCAs. Finally, it showed how the concrete architecture could be realized to lead to Enterprise 2.0.

This work has opened new issues in developing enterprise SCAs, namely, how to realize Enterprise 2.0. In further development, we could extend this work to a method to build Enterprise 2.0 by integrating more interaction operations using SOA and WSs.

REFERENCES