Chapter 19
Realizing a Social Ecosystem of Web Services

Zakaria Maamar, Youakim Badr, Noura Faci and Quan Z. Sheng

Abstract The success in Web services goes well beyond the building of loosely-coupled, interoperable software components. Nowadays, large-scale collaboration through social media (e.g., social networks) and new generation of service-oriented software have spurred the growth of Web service ecosystems. This chapter discusses how a social ecosystem of Web services can be realized by defining first, the necessary actors that take part in this ecosystem formation and second, the interactions that occur between these actors during this ecosystem management. Such ecosystem permits to track who does what and where and when it is done. Compared to (regular) Web services, Web services in a social ecosystem take different actions that allow them, for instance to establish and maintain networks of contacts with other peers and to form with some peers strong and long lasting collaborative groups. The actors in the ecosystem are referred to as providers of Web services, providers of social networks of Web services, consumers of Web services, and providers of social networks of consumers. They all engage in different types of interactions like making Web services sign up in social networks of Web services, supporting users seek advices from existing members in a social network of consumers, and combining social networks of consumers and of Web services to achieve users' requests. Existing

Z. Maamar (✉)
Zayed University, Dubai, U.A.E
e-mail: zakaria.maamar@zu.ac.ae

Y. Badr
INSA de Lyon, Villeurbanne 69621, France
e-mail: youakim.badr@insa-lyon.fr

N. Faci
Claude Bernard Lyon 1 University, Lyon, France
e-mail: noura.faci@univ-lyon1.fr

Q. Z. Sheng
The University of Adelaide, Adelaide, Australia
e-mail: qsheng@cs.adelaide.edu.au

A. Bouguettaya et al. (eds.), *Advanced Web Services*,
DOI: 10.1007/978-1-4614-7535-4_19,

research initiatives on social Web services as well as open issues in the development of a social ecosystem of (social) Web services are also discussed in this chapter.

19.1 Introduction

The IT community regularly hails Web services for their capacity of implementing loosely-coupled, cross-organization business applications. This is primarily due to the properties that characterize Web services [2]: (*i*) independent as much as possible from specific platforms and computing paradigms; (*ii*) primarily developed for inter-organization situations; and (*iii*) easy to integrate into existing applications so that developing complex adapters for composition needs is not required. Composition is one of Web services' attractive features. It allows to put several Web services together in response to complex users' requests.

In previous work (e.g., [12] and [13]) we designed and developed social Web services in response to certain limitations that undermine (regular) Web services efficient operation. Among these limitations we cite the following: (*i*) Web services know about themselves only, not about their users or peers; (*ii*) Web services cannot reconcile ontologies among each other or with their users; and (*iii*) Web services cannot delegate their invocation requests to other peers. Contrarily social Web services can establish and maintain networks of contacts; count on their (privileged) contacts when needed; form with other peers strong and long lasting collaborative groups; and know with whom to partner so that effort reconciliation due to disparities like semantics is minimized. Web services operation illustrates perfectly how people behave when it comes to *offering services that somebody else may need* and *requiring services that somebody else may offer*. Service offering and requiring permit to connect Web services together (this connection leads into labeling Web services as social), and hence to enrich them with social elements like collaboration and coordination.

Social Web services' operations (e.g., count on their contacts when needed) are made possible because of various details (e.g., collaboration level between peers) that are extracted from the social networks that have these social Web services as members. Networks (e.g., competition, collaboration, and substitution) are developed in order to support social Web services operation. For instance, a social Web service maintains its own network of collaborators, so that it decides if working with certain peers is rewarding based on previous experiences. A social Web service can, also, recommend peers to join its underdeveloped compositions so that additional details are returned to users. Last but not least, a social Web service learns about its competitors, so that it can attempt to improve its non-functional properties.

In this chapter, we identify the necessary actors related to social Web services management in terms of description, announcement, discovery, and connection. We expect that all these actors will form a social digital ecosystem. In this ecosystem the social Web services will be described, discovered, offer services (*a.k.a* functionalities) to users and other peers, tested prior to their use, held accountable for their actions, to cite just a few. A general definition of ecosystem states that "*it is a* *natural system consisting of all plants, animals and microorganisms (biotic factors) in an area functioning together with all the non-living physical (abiotic) factors of the environment*" [3]. Our work on social Web services does not include a complete compilation of all these actors and thus, questions like who are these actors, what are their roles, and how do they interact need to be addressed.

The main contributions of this chapter are manifold: (*i*) a definition of what a social ecosystem of (social) Web services is; (*ii*) a list of all actors contributing to the management of this ecosystem along with their specific roles; (*iii*) a list of existing research initiatives that study social Web services; and (*iv*) a list of open issues that need to be addressed in order to make this ecosystem operational. The rest of this chapter is organized as follows. Section 19.2 discusses the blend of social computing with service computing and provides a literature review of the Web services ecosystems field. Section 19.3 presents an ecosystem of social Web services in terms of architecture, actors in this ecosystem, interactions between these actors, and finally open issues. Conclusions are drawn in Sect. 19.4.

19.2 Background

This section discusses how social computing meets service computing and then, provides an overview of some initiatives on Web services ecosystems.

19.2.1 When Social Computing Meets Service Computing

Current research on blending social computing (Web 2.0) with service computing (Web services) sheds the light on two categories of social networks: those connecting users and those connecting Web services.

On the one hand, social networks of users record users' experiences interacting with Web services over time so that these experiences are captured and shared later with other members of these networks. Assuming that users' feedbacks on these interactions are fair (i.e., unbiased), it becomes possible to advise users on where to look for Web services, how to select Web services, and what to expect out of Web services. A good number of approaches that study Web services-based social networks of users are reported in the literature. Xie et al. propose a composition framework that relies on social based recommendations of semantic Web services [33]. Wu et al. rank Web services using run-time non-functional properties and invocation requests [32]. Ranking takes into account the popularity of a Web service is the social element analyzed by users. Al-Sharawneh and Williams mix semantic Web, social networks, and recommender systems technologies to help users select Web services with respect to their functional and non-functional requirements [1]. Besides the market-leader concept that refers to the best Web service, Al-Sharawneh and Williams use two ontologies that are (*i*) follow-leader ontology to classify users and (*ii*) preference ontology to specify users' preferences. Maaradji et al. propose an event-driven social composer to assist users take actions in response to events such as selecting a given Web service [22]. Finally, Nam Ko et al. discuss the way the social Web (exemplified by well-known networking sites such as Facebook) contributes to create social applications without having to build social networks [25].

On the other hand, social networks of Web services record the situations that Web services come across at run time [18]. These situations known as collaboration, competition, and substitution permit to tell users which Web service peers can or like to collaborate with a Web service, which Web service peers compete in case of selection, and which Web service peers can replace a failing Web service. Different approaches that study social networks of Web services are reported in the literature. In [14] Maamar et al. introduce a method for engineering social Web services. This engineering requires identifying relationships between Web services, mapping these relationships onto social networks, building social networks of social Web services, and setting the social behaviors of social Web services. In [20] the same authors inject social networks' elements into Web services discovery process. Indeed Web services are not "isolated" components that respond to user queries, only. They compete and collaborate permanently during selection and composition, respectively. In [21] Maamar et al. also discuss the different social networks in which Web services can sign up, for instance supervision, competition, substitution, collaboration, and recommendation. The mining of these networks results in identifying social qualities like selfishness, fairness, and unpredictability that Web services exhibit at run time. Finally, in [7] and [15], Maamar et al. propose a set of quality criteria that help Web services assess the pros and cons of signing-up in these networks. This set includes, but is not limited to, privacy, trust, fairness, and traceability. Policies for managing the sign up are also provided in this paper. The adoption and efficiency of these policies are monitored and assessed with respect to the values that these criteria take.

19.2.2 Literature Review

A search of the Web services ecosystems field identifies an exhaustive list of research initiatives [6, 11, 26–28, 31]. In the following we summarize some and discuss how and why they fall short of meeting the intrinsic characteristics of social Web services.

In [11] Li and Chen consider that the overlap between social computing, Internet of things, service computing, and cloud computing disciplines result in a new discipline that is social services computing. This overlap means that the respective constituents in these disciplines interact with each other to form social networks. Social services computing needs to carefully look into service management in terms of classification, clustering, migration, recommendation, composition, discovery, and publication all from a social perspective. The social services ecosystem of Li and Chen consists of service computing infrastructure, social consumers, social providers, and social networks. In this ecosystem services can be shared, partially shared, leased, or sold.

In [26] Riedl et al. propose a framework to analyze service ecosystem capabilities. This ecosystem includes repositories of services that can be re-used, re-combined, and re-purposed to create new, innovative services. The actors populating this ecosystem are: providers, users/customers, brokers that bring service providers and consumers together, mediators that offer translation between different service formats and other routine functions and support brokers in their operation, and specialist intermediaries that offer service delivery components used by others.

In [27] Scheithauer et al. propose a set of necessary properties to describe services in service ecosystems. These latter are electronic marketplaces where services can be traded over the Internet. Two obstacles impede this type of trade: lack of adequate properties for service description and lack, also, of a clear classification for service description notations. Scheithauer et al.'s proposed properties and classifications are as follows: functionality properties namely capability and classification, financial properties namely price, payment, and discount, legal properties namely rights, obligations, and penalty, marketing properties namely certification, expert test rating, and benefit, and finally quality of service properties namely latency, throughput, availability, and reliability.

In [31] Wu and Chang discuss the limitations of the centralized Web services client/service architecture in terms of performance, bottleneck, and scalability and propose DWSASE, standing for Distributed Web Services Architecture for Service Ecosystem, to address these limitations. The components upon which DWSASE is built upon are service peer, domain peer, alliance peer, super peer, domain broker, domain UDDI, global broker, and global UDDI. For instance a service peer is an ordinary service provider and/or service consumer available in an area that does not belong to any domain known as global space. In addition, a super peer initiates the formation of a particular dynamic alliance by sending invitation messages to selected partner peers. The DWSASE components interact according to different protocols that are Web service community protocol, broker protocol, alliance P2P protocol, super-peer protocol, WS business protocol, and domain protocol.

The aforementioned paragraphs offer a glimpse of existing ecosystems for managing Web services. However social Web services' intrinsic features raise other challenges that these ecosystems do not consider for instance, what types of networks social Web services can sign up in, how social Web services get to know about available networks, what billing means can networks adopt for the resource offered to social Web services, how to assist social Web services select the best networks, and how to assess the quality of services that networks offer. The next section proposes a dedicated ecosystem for social Web services.

19.3 Social Web Services Ecosystem

This section begins by proposing a set of components (called actors later) upon which a dedicated social ecosystem of (social) Web services is built. Afterwards it discusses the interactions that occur between these components as well as the existing research initiatives that look into these interactions.

19.3.1 Architecture of the Ecosystem

Figure 19.1 illustrates an architecture for an ecosystem of social Web services. Four different clusters hosting each similar actors populate this ecosystem. These clusters are **P**roviders of Web Services (**P**$_{ws}$), **P**roviders of **S**ocial **N**etworks of social Web services (**PSN**$_{sws}$), **C**onsumers of Web Services (**C**$_{ws}$), and **P**roviders of **S**ocial **N**etworks of consumers (**PSN**$_{c}$). Consumers and providers refer here to both persons and organizations. Web services turn out social when they sign up in at least a **PSN**$_{sws}$'s social network. In the same figure discontinued lines represent cross-cluster interactions that are detailed in Sect. 19.3.3. In this ecosystem there is no central authority in charge of managing the social networks of social Web services or of consumers. Therefore mechanisms that allow to identify who does what are critical and constitutes an open issue to address in the ecosystem. The different networks are completely independent from each other, though bridges connecting social networks of users (i.e., consumers) may exist like discussed in [4].

The actors populating the four clusters are briefly discussed below:

- **P**$_{ws}$ cluster identifies all providers who develop and make Web services available for invocation. The providers rely on regular means like service registries (e.g., UDDI) or **PSN**$_{sws}$ to announce their Web services to potential consumers. Registries are excluded from the architecture since the ecosystem relies on social networks to expose Web services to the external world.
- **C**$_{ws}$ cluster identifies all consumers who invoked Web services and recorded their experiences of using these Web services. Records concern for instance, the quality of response and reliability level (aka QoS). The consumers consult service registries or rely on the **PSN**$_{c}$ to identify the Web services that they will invoke. Registries are, also, excluded from the architecture.
- **PSN**$_{sws}$ cluster hosts different types of social networks of social Web services that independent providers set up. The value added of these networks to users varies depending on the nature of needs to satisfy such as building a new composite Web service, replacing a failing Web service, etc. Collaboration and competition are examples of social networks of social Web services [13].
- **PSN**$_{c}$ cluster hosts different types of social networks connecting consumers together. Facebook, LinkedIn, or any other private social network are examples of social networks of consumers that independent providers set up so that consumers sign up in to report their feedbacks and seek feedbacks on the Web services invoked/to invoke.

The four clusters engage in different interactions that are briefly discussed below. Some interactions are already part of the ecosystem (plain lines in Fig. 19.1) while the rest are recommended for inclusion in the ecosystem (discontinued lines in Fig. 19.1):

- **Interaction 1** (**P**$_{ws}$:**PSN**$_{sws}$) corresponds to the chronology of operations that allows Web services to be members of social networks of social Web services.
- **Interaction 2** (**C**$_{ws}$:**PSN**$_{c}$) corresponds to the chronology of operations that allows consumers to be members of social networks of consumers.

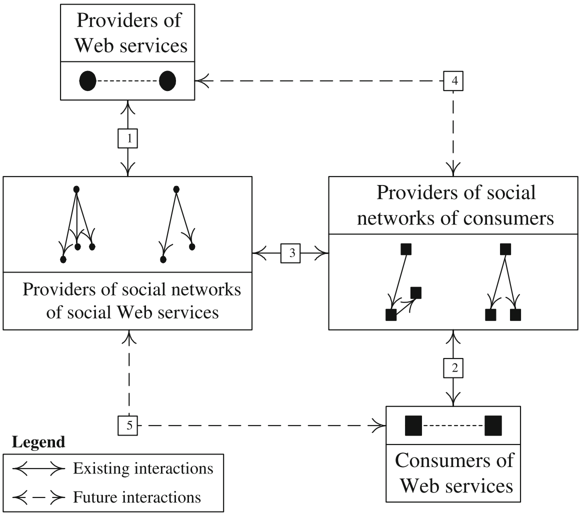

Providers of Web services
1
4
Providers of social networks of social Web services
3
Providers of social networks of consumers
2
5
Consumers of Web services
Legend
Existing interactions
Future interactions

Fig. 19.1 Proposed architecture for a social Web services ecosystem

- **Interaction 3** (**PSN**$_c$:**PSN**$_{sws}$) corresponds to the collaborative actions between the social networks of social Web services and of consumers to help consumers identify the Web services to invoke.
- **Interaction 4** (**P**$_{ws}$:**PSN**$_c$) corresponds to the actions that providers take to expose their Web services to future consumers by relying on social networks of consumers. This interaction is detailed in Sect. 19.3.4.
- **Interaction 5** (**C**$_{ws}$:**PSN**$_{sws}$) corresponds to the actions that consumers take to look for the Web services they need by screening social networks of social Web services. This interaction is detailed in Sect. 19.3.4.

19.3.2 Actors in the Ecosystem

Three types of providers and one type of consumers operate in the ecosystem of social Web services. They perform multiple operations according to their roles, needs, interests, and objectives.

Consumers of Web services correspond to persons or organizations who require Web services to satisfy their requests. Requests vary from basic like currency conversion to complex like travel planning. Consumers can sign up in different **PSN**$_c$ as per interaction 2. It is assumed the existence of mechanisms (e.g., search engines) permitting consumers to locate the relevant **PSN**$_c$. Being a member of **PSN**$_c$s gives consumers the opportunity of sharing their experiences of using Web services with other members as well as seeking these members' recommendations on potential Web services to use. Consumers have to comply with the **PSN**$_c$s' regulations when they sign up, sign off, seek advices, share feedbacks, post comments, etc. This compliance can be based on how users' rights and responsibilities are defined in some online social applications like Facebook and LinkedIn.

Providers of Web services correspond to persons or organizations who offer Web services to other persons and organizations. Web services can be put together to develop composite Web services in response to users' needs complexity. Providers make their Web services join different **PSN**$_{sws}$ as per interaction 1. Like with consumers appropriate mechanisms allow providers to locate the relevant **PSN**$_{sws}$. Web services have to comply with the **PSN**$_{sws}$s' regulations (i.e., policies) when they sign up, sign off, select a certain social network, etc. These regulations are explained in Sect. 19.3.3.

Providers of social networks of social Web services correspond to persons or organizations who offer means permitting to connect Web services together according to specific schemas. Three out of several connection schemas are studied in [13] and summarized below:

- Collaboration schema (Fig. 19.2): by combining their respective functionalities, social Web services have the capacity to work together on complex user requests. Consequently, a social Web service has its own network of collaborators, so that it decides if it likes collaborating with peers based on previous experiences. A social Web service can, also, recommend peers to join underdeveloped compositions.

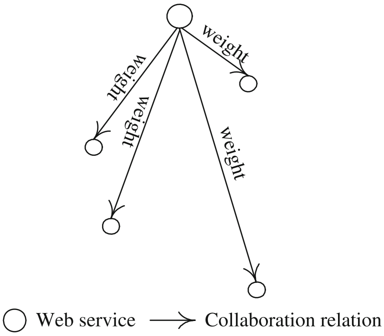


Fig. 19.2 Example of a collaboration social network connecting Web services

- Substitution schema: although social Web services compete against each other, they can still help each other when they fail as long as they offer similar functionalities. Consequently, a social Web service manages its own networks of substitutes, so that it can meet its Service Level Agreements (SLA) when it encounters a potential failure. It can then identify its own best substitutes in response to users' non-functional requirements.
- Competition schema: social Web services compete against each other when they offer similar functionalities. Their non-functional properties differentiate them when users non-functional requirements must be satisfied. Consequently, a social

Web service learns about its own network of competitors, so that it can attempt to improve its non-functional properties with respect to other peers.

Providers of social networks of consumers correspond to persons or organizations who offer means permitting to connect consumers of Web services together according to specific schemas. Recommendation is possibly the most appropriate connection schema between consumers allowing consumers to indicate potential Web services to other peers.

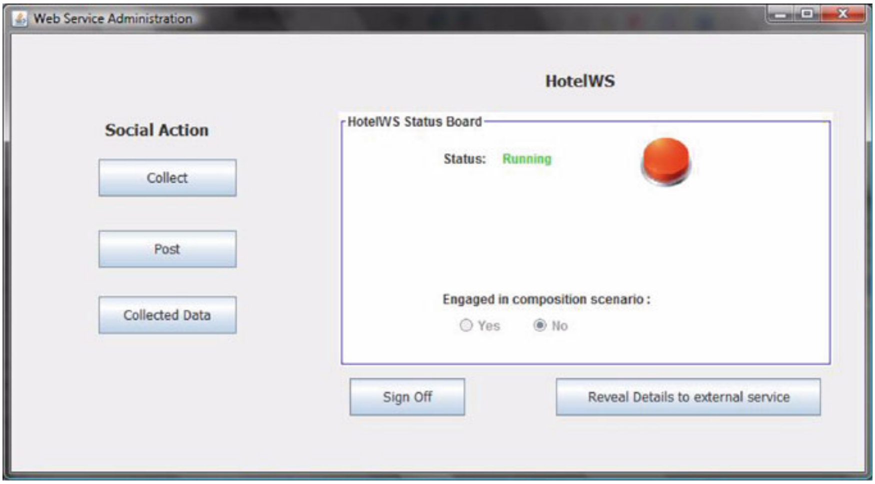
19.3.3 Interactions in the Ecosystem

19.3.3.1 Web Services/Social Networks Interactions (1)

In [16] we study the interactions that take place between Web services and $\mathbf{PSN}_{sws}$ and adopt commitments to guarantee the compliance of the future social Web services (that act on behalf of Web services) with the regulations of these $\mathbf{PSN}_{sws}$ in terms of privacy, content sharing, payment, pricing, etc. Singh et al. seem to be the first who advocate for examining service-oriented architecture principles from a commitment perspective [30]. The traditional service-oriented architecture is built upon low-level abstractions that are inappropriate for capturing the intrinsic characteristics of business services such as autonomy, complexity, and adaptability. Contrarily a commitment-based service-oriented architecture allows to judge the correctness of a service enactment as long as commitments are not violated and to support business compliance without dictating specific operationalization.

When Web services join a $\mathbf{SN}_{sws}$ (led by an authority component (sn_{auth}) and illustrated with Fig. 19.3) the social Web services perform actions whose outcomes might "harm" peers in the same network (e.g., revealing their private details), or even slowdown the operation of the network (e.g., broadcasting irrelevant details). Thus the social Web services are responsible for these actions' outcomes. A Responsibility (*Resp*) is structured as a triple: either an obligation or a permission, actions to perform, and possible conditions that authorize the execution of actions. Below is an example of responsibility.

- $Resp_1$. Collecting any detail (d) in a social network would require indicating the purpose (p) of this collection to this detail's owner (o).
  Representation: `Permission(Collect(`$d, o, valid(p)$`))`.
  `Collect` is the action; d, for instance is a non-functional property like response time and is either public (made available to all members of a social network), protected (made available to the social network's authority component, only), or private (not available); o is the owner of d for instance social Web service; p is the rationale of collecting d; and *valid* is a function that checks p. Two purposes exist: collaboration (*col*) to support the development of composite Web services and substitution (*sub*) to support the execution continuity of Web service-based business processes in case of failure.



Status: Running

Engaged in composition scenario:
○ Yes ● No

Fig. 19.3 *HotelWS* administration module

Representation: `Obligation(Post(`d, *true*`))`.
`Post` is the action and *true* is the veracity of d.
Representation: `Obligation(not-Tamper(`$d, o,$ *collection*(d)`))`.
`not-Tamper` is the action and *collection* is a function that checks if collecting d is approved in compliance with $Resp_1$.

Afterwards the responsibilities are mapped onto commitments. The formalism of Fornara and Colombetti is adopted to structure the commitments [8]:
$C_{Resp_i}(debtor, creditor, content[|condition])$ where sws_i is a social Web service, C_{Resp_i} is a commitment associated with $Resp_i$, and [ ] means optional.

- $C_{Resp_1}(sws_i, sws_j, Collect(d, sws_j)|valid(p_d))$ is a conditional commitment by sws_i to sws_j, that if $valid(p_d)$ holds then $Collect(d, sws_j)$ will be satisfied.

When a social Web service violates commitments for reasons like being malicious or temporary shortage of computation resources this requires continuous monitoring so that corrective actions are taken [24]. Besides commitment violation, it may happen that social Web services carry out actions that are prohibited calling for setting sanctions like decrementing reputation level and revoking some access privileges.

- C_{Resp_1}: violation arises when collection occurs over a non-public detail. And prohibition arises when the purpose of a detail collection is neither composition nor substitution.
  - Violation monitoring requires that sws_j reports to sn_{auth} recurrent, tentative accesses to its non-public details from sws_i. If these tentatives are confirmed using logs for example, this will be a violation to accessing non-public details of sws_j. Sanctions consist of reviewing the trust/reputation levels of sws_i if first

time. Otherwise, eject sws_i from the social network if these levels go below a threshold.

- Prohibition monitoring requires that sn_{auth} checks if sws_j was really used either as a component in an underdeveloped composition or as a substitute in an under-execution composition for the purpose that sws_i mentioned to sn_{auth} so that it collects details on sws_j. If sws_j was not used as expected, this would be a prohibition to collecting details on sws_j. Compensations include informing sws_j of what happened as well as giving it more access privileges like tracking all the peers that request its details.

19.3.3.2 Consumers/Social Networks Interactions (2)

The ecosystem of social Web services treats consumers as not mere end-users but active and trusted co-creators of new composite services. Grouping persons or organizations together into specific social networks may have an important effect on the overall ecosystem. Consumers build trust in their social networks and develop friendships, professional alliances or even cooperate to achieve business-to-business activities. Social networks are, also, important incentive factors for many consumers to organize themselves into communities, assign different roles, and share common interests. This social environment requires basic mechanisms to support consumer-driven activities and enable them to co-operate as well as re-use, combine, and share their Web services. The mechanisms (or services) that facilitate interactions between consumers and social networks of consumers are follows:

- Profile mechanism consists of creating a profile (including private and public data) for each consumer and allowing peers to discover it. Any consumer can belong to one or several social networks. Each network may have at least one or more membership groups (e.g., owners, mediators, and casual members).
- Search mechanism allows members to search social networks based on criteria like names, business domains, and location, and pro-actively recommend interesting Web services to peers to enrich their businesses.
- Contact mechanism provides basic mechanisms to maintain personal contact list, tag members and manages granting access control to profiles maintained by each member.

19.3.3.3 Social Networks Interactions (3)

The interactions between $\mathbf{PSN}_{sws}$ and $\mathbf{PSN}_c$ permit to compose, execute, and monitor Web services while taking into account both consumers' experiences in using Web services and social Web services' connections to other peers. $\mathbf{SN}_c$ and $\mathbf{SN}_{sws}$ interleave during composition, execution, and monitoring requires developing a *social composer*, a *social executor*, and a *social monitor*, respectively (Fig. 19.4, [17]).

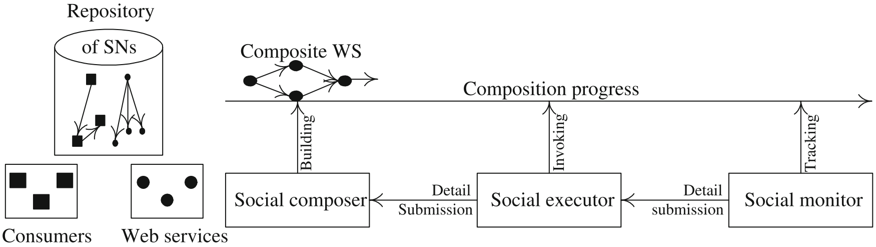


Fig. 19.4 Composer, executor, and monitor social components in action

The social composer relies on $\mathbf{SN}_c$ and $\mathbf{SN}_{sws}$ to advise users on how to build compositions (or composite Web services). These advices concern (*i*) which Web services to include in these compositions [22], (*ii*) which Web services to check in case the contacted ones decline to participate in these compositions [19], and (*iii*) which Web services to select to ensure a better compatibility level of these compositions [29].

The social executor assesses the impact of the social composer's advices (when these advices are considered) on composition execution progress. The social executor feeds the social composer with details so that the social composer updates the necessary social networks. These details include (*i*) how the Web services that were suggested through $\mathbf{SN}_c$ and $\mathbf{SN}_{sws}$ performed and (*ii*) which Web services that were also suggested did not join the compositions.

The social monitor relies on $\mathbf{SN}_{sws}$ to advise users on which Web services to check in case those that are already taking part in their respective ongoing compositions fail. The social monitor feeds the social executor with details so that this latter updates the $\mathbf{SN}_{sws}$ for the benefit of the social composer. These details include (*i*) which Web services failed, (*ii*) which Web services replaced them, (*iii*) how the replacing Web services performed, and (*iv*) how the Web services that are already in compositions reacted to the replacing Web services. Out of these details, the social monitor does more than a simple monitoring but puts forward different solutions for the social composer like assessing Web services performance.

The aforementioned social components are supported by four types social networks: recommendation [22], collaboration, competition, and substitution [13]. The former network ($\mathbf{SN}_c$) is developed to support consumers develop composite Web services. This network suggests Web services according to the current status of the composition process. The Recommendation Confidence (**RC**) as discussed in [22] is defined in Eq. 19.1.

$$\mathbf{RC}(ws_k, ws_l) = \sum_{j=1}^{n} \mathbf{NC}_{v_j}(ws_k, ws_l) \times \mathbf{F}it(v_j, ws_l) \times \mathbf{SP}(v_i, v_j) \tag{19.1}$$

where $\mathbf{NC}_{v_j}(ws_k, ws_l)$ represents how many times a user v_j used Web service ws_l following the use of Web service ws_k in compositions, $\mathbf{F}it(v_j, s_l)$ quantifies the expertise of user v_j in using service ws_l, and $\mathbf{SP}(v_i, v_j)$ defines v_i's social proximity to v_j in the recommendation network.

The collaboration, competition, and substitution social networks ($\mathbf{SN}_{sws}$) are built to support the development of composite Web services. They are established based on the functionality that Web services offer to the external community. Different techniques permit assessing either the similarity or the complementarity of Web services' functionalities, but this is outside this chapter's scope. Interested readers are referred to [5, 23]. For illustration purposes the competition social network $\mathbf{SN}_{sws}$ is analyzed. Since this network involves social Web services that act on behalf of Web services with similar functionalities, they are all in competition against each other and hence, all connected to each other through bidirectional edges. To evaluate the weight of a competition edge, which we refer to as *Competition Level* ($\mathbf{L_C}_{omp}$, Eq. 19.2) between two social Web services (sws_i, sws_j), we use the Functionality Similarity Level ($\mathbf{L_{FS}}$) to compare the functionalities of their respective Web services (ws_i, ws_j) and Non-Functionality Similarity Level ($\mathbf{L_{NFS}}$) to compare the ws_i's and ws_j's non-functional properties (e.g., reliability level and response time). We assume that the non-functional properties of Web services are defined with the same taxonomy.

$$\mathbf{L_C}_{omp}(sws_i, sws_j) = \mathbf{L_{FS}}(ws_i, ws_j) \times (1 - \mathbf{L_{NFS}}(ws_i, ws_j)) \qquad (19.2)$$

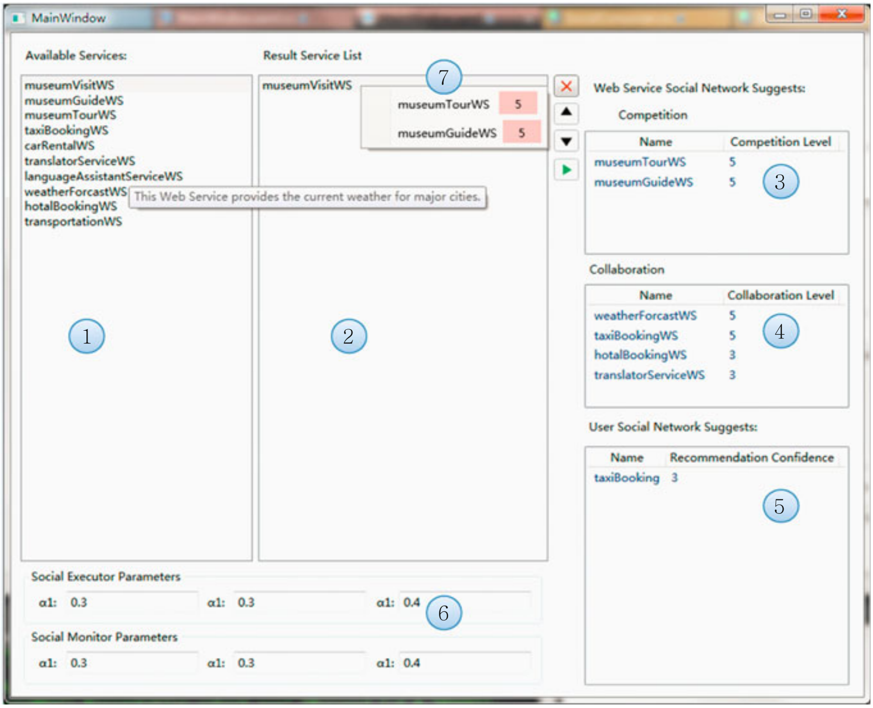
where

- $\mathbf{L_{FS}}(ws_i, ws_j)$ corresponds to the similarity level between the respective functionalities of ws_i and ws_j.
- $\mathbf{L_{NFS}}(ws_i, ws_j) = \omega_1 \times (|\mathbf{P}(ws_{i,1}) - \mathbf{P}(ws_{j,1})|) + \cdots + \omega_n \times (|\mathbf{P}(ws_{i,n}) - \mathbf{P}(ws_{j,n})|)$ with $\mathbf{P}(ws_{i,k})$ is the value of the kth non-functional property of the ith Web service (assumed to be between 0 and 1), ω_k is a weighting factor representing the importance of a non-functional property, and $\sum_{k=1}^{n} \omega_k = 1$.

As per Eq. 19.2 the more the competition level is close to one, the closer sws_i is to sws_j. As a result, ws_i threatens the competitiveness capacity of ws_j. Only one Web service can be selected at a time to complete a task in a composition.

To illustrate how the composer, executor, and monitor components operate so that the interleaving of social networks of consumers and of Web services happens (Fig. 19.5), we suggest the following scenario. A business woman who has a stop over in a city on her way back from a business trip, decides to visit some museums among other sightseeing activities. She logs into a Web site and invokes `museumVisitWS` submitting the museums that she is interested in and her budget. Different cases are listed hereafter to illustrate the role of the composer, executor, and monitor components.

1. Prior to executing `museumVisitWS`, the social composer consults the businessman's social networks of consumers. It finds out that some colleagues at work visited the same city in the past and recommend riding taxis during this time of the year due to heavy rains falling sometimes unexpectedly.



Available Services:

- museumVisitWS
- museumGuideWS
- museumTourWS
- taxiBookingWS
- carRentalWS
- translatorServiceWS
- languageAssistantServiceWS
- weatherForcastWS
- hotalBookingWS
- transportationWS

This Web Service provides the current weather for major cities.

Result Service List

- museumVisitWS

museumTourWS	5
museumGuideWS	5

Web Service Social Network Suggests:

Competition

Name	Competition Level
museumTourWS	5
museumGuideWS	5

Collaboration

Name	Collaboration Level
weatherForcastWS	5
taxiBookingWS	5
hotalBookingWS	3
translatorServiceWS	3

User Social Network Suggests:

Name	Recommendation Confidence
taxiBooking	3

Social Executor Parameters

α1: 0.3 α1: 0.3 α1: 0.4

Social Monitor Parameters

α1: 0.3 α1: 0.3 α1: 0.4

Fig. 19.5 The system frontend

2. To identify a Web service for taxi booking, the social composer consults `museumVisitWS`'s social networks of social Web services to find out that `museumVisitWS` has frequently and successfully collaborated with `taxiBookingWS`, which is subsequently selected to arrange taxi booking. Another Web service called `translatorServiceWS` is also advised by the social composer as reported in the social networks of `museumVisitWS`, but this time the businessman declines the advice since she is familiar with the language spoken in the city.
3. When the selection of `taxiBookingWS` and `museumVisitWS` is complete, the social executor invokes both while keeping an eye on all the Web services that were added to the composition through the social networks of consumers and of social Web services. The objective is to reflect the performance of these Web services on the different networks.

The aforementioned cases offer a glimpse of the advantages of each type of social networks brought to the cycle of Web services composition, execution, and monitoring. It is for sure that some of these cases can be handled by screening registries, but Web services' previous experiences and users' advices are not captured and hence, overlooked during this screening.

19.3.4 Open Issues

19.3.4.1 Interactions (4) and (5)

Providers of and consumers of Web services should be given the opportunity of interacting directly with the providers of consumers and of social Web services, respectively. Providers of Web services could develop and offer new Web services based on consumers' needs and feedbacks on existing Web services. These details can be made available through the social networks of consumers subject to guaranteeing consumers' privacy. The same applies to consumers who could express their requirements and expectations in advance to providers of social networks of social Web services so these latter offer better services like showing the collaborating Web services using graphs, for example. The questions that interaction (4) raise include the following: do providers of Web services have to sign up in social networks of consumers, how is consumers' privacy maintained, how are these providers held accountable for their actions, and how are consumers notified about providers' requests? Interaction (5) raises almost the same set of questions.

19.3.4.2 Payment and Pricing

To create a sustainable social Web services ecosystem, all actors should interact, reuse Web services, and make them available for others. In addition to these basic actions, the ecosystem should provide incentives like financial for providers to offer a large spectrum of Web services for a variety of domains (e.g., business, education, and entertainment). Accessing $\mathbf{SN}_{sws}$ requires mechanisms for electronic payments and online transactions.

Some factors that help encourage or discourage demands of Web services and regulate their usage are pricing strategies and pricing models. This regulation involves to collect and analyze "service" metrics for purposes such as billing and auditing. It requires that "service" consumption be measured and the charging information be communicated between appropriate actors. To obtain viable business models, non-standard pricing mechanisms have to be taken into consideration. Most common pricing models are based on fixed prices. For example, Günther et al. discuss the challenges associated with pricing Web services [10]. They argue that the usage-based pricing model, combined with an option to switch to a flat subscription, is a suitable strategy to penetrate the market of Web services. Bitran et al. advocate that dynamic price models are particularly useful for short selling horizons and demands that are both stochastic and price sensitive [9]. Airline companies and hotels are good examples where dynamic pricing strategies are key drivers for increasing their revenues.

In recent years, a good number payment systems are available for online transactions, such as the traditional credit card but, also, new payment systems such as Google Checkout and Paypal Check-out, which are mainly geared toward selling goods. These systems can be easily adapted to support selling Web services online. Companies that enable financial transactions, a.k.a Payment Service Providers (PSP), are viewed as important actors in the social Web services ecosystem. They do not only allow customers of Web services to transfer funds from their traditional bank accounts into providers' accounts but they establish trust relationships among all actors to collaborate within the social Web services ecosystem.

Interactions among consumers, social networks, Web services and providers, and how they are related to payment mechanisms have the following characteristics.

- Applying payment mechanisms, pricing models and strategies to social Web service ecosystems is particularly interesting since it becomes possible to collect valuable information about Web services, social networks, and actors and process them in real time. As a result, providers can act and react dynamically to changes by adjusting any variable under control, specifically prices.
- Incorporating consumers in the social Web services ecosystem offers them the ability to inquiry prices and keep track of the evolution of the selling process.
- Supporting different payment service providers and pricing models need to be reflected on the infrastructure. Existing service registries needs for example to be extended to support the social Web services ecosystem by including more complex transactions such as negotiations and auctioning.

Emerging potential applications for pricing and payment can be useful for the social Web service ecosystem. Although different in many respects, these applications have to support all actors and deal with the complexity that comes from perishability of Web services and social networks, short selling horizons, and price sensitivity and unpredictable demand of consuming Web services.

19.4 Conclusion

The social Web services ecosystem initiative as a novel approach for fostering development, discovery and, usage of Web services provides a sustainable environment by which all actors share and recommend trustworthy Web services. This chapter discussed the realization of an ecosystem of social Web services. This realization identified the necessary actors upon which this ecosystem is built namely providers of Web services who correspond to persons or organizations offering Web services to other persons and organizations, consumers of Web services who correspond to persons or organizations requiring Web services to satisfy their requests, providers of social networks of social Web services who offer means that permit to connect Web services together according to specific schemas like collaboration and substitution, and last but not least providers of social networks of consumers who offer means permitting to connect consumers of Web services together according to specific schemas like recommendation. Different types of interactions occurred between all these actors such as making Web services sign up in a social network of social Web services, supporting users seek advices from other members in a social network of consumers, and combining social networks of consumers and of Web services to achieve users' requests. Some interactions between these actors are already investigated from different perspectives for instance making Web services sign up in a social network of social Web services requires the compliance of these Web services with this social network's internal regulations to avoid privacy issues. This compliance is being handled through commitments. The rest of interactions that correspond to providers of and consumers of Web services interacting directly with the providers of social networks of social Web services and of consumers are still pending and hence, further investigation is required.

Acknowledgments The authors acknowledge the contributions of Khouloud Boukadi (Sfax University, Tunisia), Salahdine Hachimi (Claude Bernard Lyon 1 University, France), and Lina Yao (The University of Adelaide, Adelaide, Australia) to the social Web services research initiative.

References

1. J. Al-Sharawneh and M.-A. Williams. A Social Network Approach in Semantic Web Services Selection using Follow the Leader Behavior. In *Proceedings of the 13th Enterprise Distributed Object Computing Conference Workshops (EDOCW'2009)*, Auckland, New Zealand, 2009.
2. B. Benatallah, Q. Z. Sheng, and M. Dumas. The Self-Serv Environment for Web Services Composition. *IEEE Internet Computing*, 7(1), January/February 2003.
3. R. W. Christopherson. *Elemental Geosystems*. Prentice Hall, 1997.
4. D. Dasgupta and R. Dasgupta. Social Networks using Web 2.0, Part 2: Social Network as a Service (SNaaS). Technical report, IBM, developerWorks, http://www.ibm.com/developerworks/webservices/library/ws-socialpart2/index.html?ca=drs, 2010.
5. B. Di Martino. Semantic Web Services Discovery based on Structural Ontology Matching. *International Journal of Web and Grid Services*, 5(1), 2009.
6. H. Dong, F. K. Hussain, and E. Chang. A Human-Centered Semantic Service Platform for the Digital Ecosystems Environment. *World Wide Web*, 13(1-2), 2010.
7. N. Faci, Z. Maamar, and P. Ghodous. Which Social Networks Should Web Services Sign-Up In? In *AAAI Spring Symposium on Intelligent Web Services Meet Social Computing (IWEBSS'2012)*, Palo Alto, USA, 2012.
8. N. Fornara and M. Colombetti. Operational Specification of a Commitment-based Agent Communication Language. In *Proceedings of the First International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS'2002)*, Bologna, Italy, 2002.
9. Bitran G. and R. Caldentey. An Overview of Pricing Models for Revenue Management. Technical report, MIT Sloan Working Paper No. 4433-03, December 2002.
10. O. Günther, G. Tamm, and F. Leymann. Pricing Web Services. *International Journal of Business Process Integration and Management*, 2(2), 2007.
11. S. Li and Z. Chen. Social Services Computing: Concepts, Research Challenges, and Directions. In *Proceedings of the 2010 IEEE/ACM International Conference on Green Computing and Communications (GreenCom'2010) & 2010 IEEE/ACM International Conference on Cyber, Physical, and Social Computing (CPSCom'2010)*, Hangzhou, China, 2010.
12. Z. Maamar, F. Faci, L. Krug Wives, P. Bispo dos Santos, Y. Badr, and J. Palazzo Moreira de Oliveira. Using Social Networks for Web Services Discovery. *IEEE Internet Computing*, 15(4), 2011.
13. Z. Maamar, N. Faci, Y. Badr, L. Krug Wives, P. Bispo dos Santos, D. Benslimane, and J. Palazzo Moreira de Oliveira. Towards a Framework for Weaving Social Networks Principles into Web Services Discovery. In *Proceedings of the International Conference on Web Intelligence, Mining, and Seantics (WIMS'2011)*, Sogndal, Norway, 2011.

14. Z. Maamar, N. Faci, L. Krug Wives, H. Yahyaoui, and H. Hacid. Towards a Method for Engineering Social Web Services. In *Proceedings of the IFIP WG8.1 Working Conference on Method Engineering (ME'2011)*, Paris, France, 2011.
15. Z. Maamar, N. Faci, A. Loo, and P. Ghodous. Towards a Quality of Social Network (QoSN) Model in the Context of Social Web Services. In *Proceedings of the 3rd International Conference on Exploring Services Science (IESS'2012)*, Geneva, Switzerland, 2012.
16. Z. Maamar, N. Faci, M. Luck, and S. Hachimi. Specifying and Implementing Social Web Services Operation using Commitments. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC'2012)*, Riva del Garda, Trento, Italy, 2012.
17. Z. Maamar, N. Faci, Q. Z. Sheng, and L. Yao. Towards a User-Centric Social Approach to Web Services Composition, Execution, and Monitoring. Technical report, Zayed University Working Paper, January 2012.
18. Z. Maamar, H. Hacid, and M. N. Huhns. Why Web Services Need Social Networks. *IEEE Internet Computing*, 15(2), March/April 2011.
19. Z. Maamar, S. Kouadri Mostéfaoui, and H. Yahyaoui. Towards an Agent-based and Context-oriented Approach for Web Services Composition. *IEEE Transactions on Knowledge and Data Engineering*, 17(5), May 2005.
20. Z. Maamar, L. Krug Wives, Y. Badr, S. Elnaffar, K. Boukadi, and N. Faci. `LinkedWS`: A Novel Web Services Discovery Model Based on the Metaphor of "Social Networks". *Simulation Modelling Practice and Theory, Elsevier Science Publisher*, 19(10), 2011.
21. Z. Maamar, H. Yahyaoui, E. Lim, and P. Thiran. Social Engineering of Communities of Web Services. In *Proceedings of the 11th Annual International Symposium on Applications and the Internet (SAINT'2011)*, Munich, Germany, 2011.
22. A. Maaradji, H. Hacid, J. Daigremont, N. Crespi. Towards a Social Network Based Approach for Services Composition. In *Proceedings of the 2010 IEEE International Conference on Communications (ICC'2010)*, 2010.
23. L. Min, S. Weiming, H. Qi, and Y. Junwei. A Weighted Ontology-based Semantic Similarity Algorithm for Web Services. *Expert Systems with Applications*, 36(10), December 2009.
24. S. Modgil, N. Faci, F. Rech Meneguzzi, N. Oren, S. Miles, and M. Luck. A Framework for Monitoring Agent-based Normative Systems. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'2009)*, Budapest, Hungary, 2009.
25. M. Nam Ko, G. P. Cheek, M. Shehab, and R. Sandhu. Social-Networks Connect Services. *IEEE Computer*, 43(8), August 2010.
26. C. Riedl, T. Böhmann, J. M. Leimeister, and H. Krcmar. A Framework for Analysing Service Ecosystem Capabilities to Innovate. In *Proceedings of the 17th European Conference on Information Systems (ECIS'2009)*, Verona, Italy, 2009.
27. G. Scheithauer, S. Augustin, and G. Wirtz. Describing Services for Service Ecosystems. In *Proceedings of the ICSOC 2008 Workshops held in conjunction with the 6th International Conference on Service Oriented Computing (ICSOC'2008)*, Sydney, Australia, 2009.
28. J. Schulz-Hofen. Web Service Middleware - An Infrastructure For Near Future Real Life Web Service Ecosystems. In *Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications (SOCA'2007)*, Newport Beach, California, USA, 2007.
29. Q. Z. Sheng, J. Yu, Z. Maamar, W. Jiang, and X. Li. Compatibility Checking of Heterogeneous Web Service Policies Using VDM++. In *Proceedings of the IEEE Workshop on Software and Services Maintenance and Management (SSMM'2009) held in conjunction the 2009 IEEE Congress on Services, Part I (SERVICES I'2009)*, 2009.
30. M. P. Singh, A. K. Chopra, and N. Desai. Commitment-Based Service-Oriented Architecture. *Computer*, 42(11), November 2009.
31. C. Wu and E. Chang. A Conceptual Architecture of Distributed Web Services for Service Ecosystems. In *Proceedings of the 18th International Conference on Computer Applications in Industry and Engineering (CAINE'2005)*, Hawaii, USA, 2005.
32. Q. Wu, A. Iyengar, R. Subramanian, I. Rouvellou, I. Silva-Lepe, and T. Mikalsen. Combining Quality of Service and Social Information for Ranking Services. In *Proceedings of ServiceWave*

2009 Workshops held in conjunction with the 7th International Conference on Service Service-Oriented Computing (ICSOC'2009), Stockholm, Sweden, 2009.

33. X. Xie, B. Du, and Z. Zhang. Semantic Service Composition based on Social Network. In *Proceedings of the 17th International World Wide Web Conference (WWW'2008)*, Beijing, China, 2008.