Towards User-aware Service Composition

Giacomo Cabri**, Mauro Leoncini, Riccardo Martoglia, Franco Zambonelli

Università di Modena e Reggio Emilia, Italy,
`name.surname@unimore.it`

Abstract. We are more and more witnessing an increasing support of information technology in our everyday life, more specifically a growing number of services exploited through our electronic devices. The research we are conducting in the context of the AMBIT project (Algorithms and Models for Building context-dependent Information delivery Tools) is aimed at supporting the development of services that are able to automatically tailor themselves on the basis of the user profile. The adaptation of the single services is only the first step towards this goal; the next step is to achieve adaptation by *composing* them. In this paper, we explore how to compose services in a user-aware way, finding the composition that better meets the requirements of the users. In particular, user profiles are exploited not only to provide users with customized services, but also to compose them in the most suitable way.

Key words: services, user-awareness, context

1 Introduction

We live in a device-supported world, where electronic devices provide us a lot of services in an ubiquitous way. Currently we have smartphones that enable us to perform requests and to get different kinds of information and services. In a not so far future, we will be surrounded by a multitude of different devices, from smart monitors that will provide us information in an adaptive way situational information to the surrounding public, to smart objects and wearables able to continuously interact with us.

All these interconnected devices will form an infrastructural substrate that could become possibly very useful to help users in performing different kinds of activities. However, the risk exists is that such potentially very large set of provided services will lead to confusion rather than helping users, who could eventually be overwhelmed by information and stimuli.

To overcome this problem, many researchers have proposed to develop applications with user-awareness capabilities [2, 7, 31]. A user-aware application recognizes the context in which the user is performing an activity by means of that application and exploits contextual information to adapt its behaviour.

In the literature we can find different approaches that address specific problems that arise in the development of user-aware applications (e.g., [1, 6, 24, 26]). The limitation of the existing approaches is that they are not global, being bounded to specific application fields or specific aspects of the context. So, in the frame of

---

** Corresponding author: giacomo.cabri@unimore.it the AMBIT (Algorithms and Models for Building context-dependent Information delivery Tools) project[3] we have defined a model of user profile that aims at being more global than existing ones [8, 13]; this model is composed of the following components: *Environment* parameters, which is a set of external conditions surrounding the user while performing activities (e.g., using an application); *Personal parameters*, which contain the essential data about the user's profile (e.g. name, gender, age, nationality), which are usually set by the user during the configuration of the application or the service; *History* parameters, which record past actions of the users, in order to have a more complete picture of the user itself.

This model of context is useful to enable a single application or a single service to adapt itself in order to provide more tailored functionalities.

In this paper we address the *composition* of different services. In fact, more and more often users requests are satisfied by a set of services that are composed to realize a higher-level service. In Section 4.1 readers can find an example. It happened for the web services [27], it is happening for the cloud services [16] and we can imagine that it will be the future for distributed systems [12].

Starting from these consideration, we propose to apply user-awareness to the composition of services, in order to meet the users' requests in a more customized way than the bare adaptation of single services. To this purpose, we rely on the SAPERE middleware infrastructure [9, 32], which enables the dynamic and adaptive composition of services based on flexible nature-inspired rules, and extend its architecture in order to integrate adaptive and semantic composition of services, accounting in a semantic way for the current situation and context of users, accordingly to the models defined in the AMBIT project.

2 Related Work

There is an already impressive body of work that has addressed many research problems in the area of user awareness and context-dependent service delivery. Indeed, the enormous growth of the mobile device market and the need to support the so-called "Web of Things and Services" have provided many motivations. In particular, for some comprehensive surveys the reader is referred to [10, 19, 28, 17].

In this huge active area, our research is primarily directed towards the definition of a general model of user profile and context, and its exploitation in order to provide better tailored service compositions and services. Let us start by recalling that there are many meanings associated to the word "context"; this usually depends on the particular application(s) that researchers have in mind. For instance, the driving environment is the context of interest in automotive applications, in health-care applications context is likely to refer to the different variables that apply to a specific patient. Moreover, in online advertising the context is essentially the page where the commercial is to be displayed, i.e., its contents, the prevailing sentiment, etc.

Differently from these cases, we are interested in a general notion of user profile and context, possibly including all of the above and much more. A handful of research contributions can be found that are relevant to this specific area. In

---

[3] http://www.agentgroup.unimore.it/ambit/

[4], the authors consider different types of context information (i.e., physical, computational, user context) and provide a solution to the problem of modeling and representation based on automated reasoning. The idea is that reasoning makes it possible to attain pieces of information that are suitable for context-aware applications. This highly cited survey also covers some important approaches to context modeling (i.e., object-role based, spatial models, ontology-based).

The paper [14] proposes a SOA-based approach to build automation systems; a description offering context information is provided to the relevant devices. Moreover, a composition engine coordinated appropriate devices/services based on the context, composition plan, and predefined policy rules. The approach proposed by Peko et al. [25] considers that enterprises must adapt to the changes in the context they operate, while always being sustainable in terms of economic, environmental, societal, and cultural concerns. The enterprises' context is modeled in terms of strategy, organization, process, and information. With respect to these works, we aim at exploiting a more general set of information (not only service/device enterprise-oriented); in particular, our approach is designed to address not only specific fields such as building automation or enterprise management, but a wider range of contexts and services.

The survey [5] tackles context modeling and awareness issues within the Context-ADDICT project of the Politecnico di Milano (see http://poseidon.ws.dei.polimi.it/ca/). The authors propose a context management system aside of the so-called "operational system". Whereas the latter is application-dependent, the context management system is not, and it exhibits a hierarchical structure expressed in terms of external parameters that have an internal representation within a context schema. In order to build the bases of our user-aware service composer we will definitely consider this separation of concerns.

A context-based approach for service discovery is proposed in [29]. The focus on services is interesting for our purpose, even if we do not consider the discovery. A formal definition of the context is only provided; we will evaluate it and possibly make it more general.

3 The SAPERE Approach to Service Composition

3.1 The SAPERE Model

SAPERE starts from consideration that the large multitude of ubiquitous services that will soon enrich our lives, will make it suitable to model the ensemble of such services as a sort of distributed pervasive service *ecosystem* [32].

SAPERE conceptually models such pervasive ecosystem as a virtual *spatial environment*[30], laid above the actual network of devices infrastructure. The environment acts as a sort of shared space in which all service components situate, and the environment itself takes care of mediating all interactions. In other words, the spatial environment represents the ground on which services of different species indirectly interact and combine with each other. Such interactions take place in respect of a limited set of basic interaction laws (also called "eco-laws", due to their nature-inspired origins), and typically accounting on the spatial and contextual relationships between services.

For the *service components* populating in the ecosystem, SAPERE adopts a common modeling and a common treatment. Each of them has an associated semantic representation which we call “LSA” (*Live Semantic Annotations*, and describing a list of properties and characteristics for each services), to be injected in the spatial environment as it it were a sort of shared spatial memory. LSA support semantic and context-aware interactions both for service aggregation/composition and for data/knowledge management.

The *eco-laws* define the basic interaction policies among the LSAs of the various services of the ecosystem. The idea is to enforce on a spatial basis, and possibly relying on diffusive mechanisms, dynamic composition of data and services by composing their LSAs and exchanging data via them. Data and services (as represented by their associated LSAs) will be sort of chemical reagents, and interactions and compositions will occur via chemical reactions, relying on semantic pattern-matching between LSAs.

Without going into details about the specific of all the SAPERE eco-laws, we want to emphasize here that the advanced forms of adaptive pattern matching between LSAs that they enforce, can make it possible to dynamically compute, at any time and for every service of the ecosystem, the list of services potentially matching with which other services towards some forms of service composition.

Adaptivity in SAPERE is not in the capability of individual services, but in the overall self-organizing dynamics of the service ecosystem as a whole. In particular, adaptivity will be ensured by the fact that any change in the system (as well as any change in its services or in the context of such services, as reflected by dynamic changes in their LSAs) will reflect in the firing of new eco-laws, thus possibly leading to the establishment of new compositions or aggregations, and/or in the breaking of some existing service compositions.

3.2 The SAPERE Middleware

The execution of SAPERE applications is supported by a middleware infrastructure [9] that implements the SAPERE approach so that mobile devices can host SAPERE nodes.

The repository of LSAs for local services is implemented as a local tuple space [11] on every SAPERE involved in the SAPERE ecosystem. In addition, every node has a local eco-laws engine. SAPERE adopts a connection schema based on neighborhood: each LSA-space is connected other nodes based on spatial proximity. The middleware provides an API to access the local LSA space, to advertise themselves (via writing an LSA), and to support the services’ need of continuously updating their LSAs. the API enables also services to detect local events of change in LSAs or of enactment of eco-laws on available LSAs.

Eco-laws are represented by a set of rules in the SAPERE nodes. For each node, the same set of eco-laws applies to define the dynamics between local LSAs (in the form of bonding, aggregation, and decay) and those between non-locally-situated LSAs (via the spreading eco-law that can propagate LSAs from a node to another to support distributed service interactions and composition).

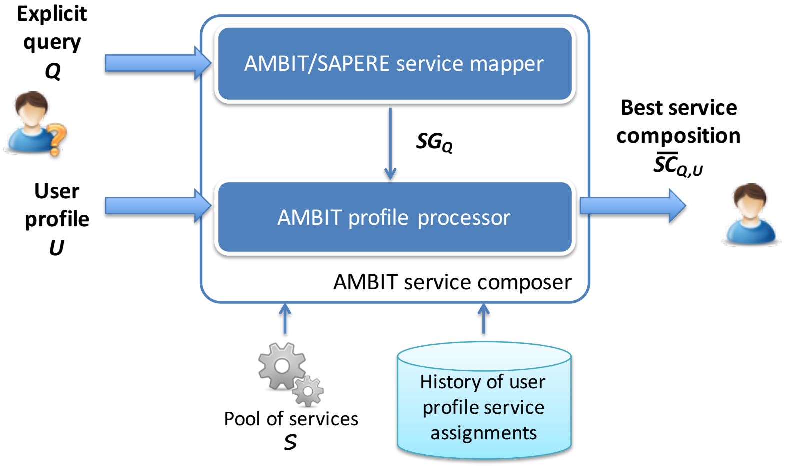

Fig. 1. High-level architecture of the AMBIT service composer

4 Towards Service Composition based on User Profile

In this section we present the AMBIT service composer, relying on the SAPERE middleware, and our approach to user-aware service composition.

4.1 Case Study

As case study, we consider an e-commerce transaction, which can be composed of some services. Typically, in this scenario a customer wants to buy a good and to receive it at home. We can suppose that three kinds of services are involved in this transaction. The first service is the one that support the actual purchase of the good, which we call *shop* service; it provides information such as which the goods are on sale, their price, their availability, the colors, the weight, and so on. The second service enables the payment of the purchased good; we can mention a *credit card* service, which can be provided by different banks. Finally, a *delivery* service takes care of the delivery of the good; usually mail carriers or express courier are exploited to ship the good to the user's home, but other kinds are possible, for instance the free delivery to a physical shop in the user's city.

Nowadays, these services are provided as a whole by online e-commerce site, with few choices, but in the future we envision that many services of each kind can exist, and the users are free to compose them to carry out a high-level transaction; even, a software platform can suggest users which composition best fits their needs, on the base of their preferences.

4.2 The AMBIT Architecture

The overall architecture of the AMBIT service composer includes two main modules: the AMBIT/SAPERE (which builds over SAPERE) and the AMBIT profile processor.

The AMBIT/SAPERE *service mapper* is basically an instance of that SAPERE middleware, embedding eco-laws and capable of digesting the LSAs of the different services of an ecosystem. In reaction to a user request that is translated into a query Q, which from within the SAPERE middleware takes the form of an LSA representing specific desirable features of a service, the eco-laws embedded within the SAPERE middleware react by determining – via pattern-matching – the set of possible service compositions SG_Q matching Q.

The AMBIT *profile processor* has the goal of analyzing the set of possible service compositions SG_Q so as to find the best service composition $\overline{SC}_{Q,U}$ built upon the services available in pool $\mathcal{S}$ and matching at the best the user profile U (i.e., its preferences and context).

4.3 Service Mapper and Service Graphs

The AMBIT/SAPERE service mapper takes in input the query Q and determines a network (***service graph***) SG_Q of suitable service interactions, which represent the composition.

We assume a query Q to be characterized by a set of keywords, i.e., $Q = \{k_i^Q\}_{i=1,\ldots,m}$. Similarly, the context is given by a user profile $U = \{k_i^U\}_{i=1,\ldots,n}$; in this case, the keywords k_i^U can be determined using text analysis techniques, such as the ones described in [22], operating on the environment, user, and history data of the profile. Also, we consider a pool of available services $\mathcal{S}$; each service $S \in \mathcal{S}$ is defined as $S = \{k_i^S\}_{i=1,\ldots,l}$ a set of keywords k_i^S derived from the service description that characterize the service itself.

SG_Q is defined as a connected directed labeled graph $SG_Q = (\overline{\mathcal{S}}, I, w_Q)$ where $\overline{\mathcal{S}} \subseteq \mathcal{S}$ is a set of nodes (services), $I \subseteq S \times S$ is a set of directed arcs (service interactions) and $w_Q : I \rightarrow [0, 1]$ is a function mapping directed arcs to their weights. SG_Q includes a source S_s and sink S_t corresponding to fictitious service nodes. The idea behind the weights $w_Q(S_1, S_2)$ is to quantify the relevance and suitability of a particular service interaction (S_1, S_2) w.r.t. Q. We do this by means of the following formula:

$$w_Q(S_1, S_2) = min(kwsim(S_1, Q), kwsim(S_2, Q)) \tag{1}$$

where $kwsim \in [0, 1]$ is a similarity function that is computed between the keyword set Q and the sets S_1 and S_2, respectively. This can be, for instance, a Jaccard similarity [15] between the involved keyword sets. Note that we adopt a *semantic*, rather than a "syntactic" approach where keywords are matched on the basis of their semantic meaning (e.g. exploiting one or more thesauri such as WordNet [23] and taking synonyms and related terms into account [3, 22]). The minimum in Eq. 1 captures the intuition that, if one of the two services $S1$,$S2$ is not particular relevant to Q, the relevance of the resulting interaction would presumably be equally low.

In Fig. 2 we report an example of service graph related to the case study previously introduced.

4.4 Profile Processor and Best Service Composition

Given the service graph SG_Q computed on the basis of query Q, the goal of the AMBIT profile processor is to find, among all sequences $SC_{Q,U}$ of consecutive

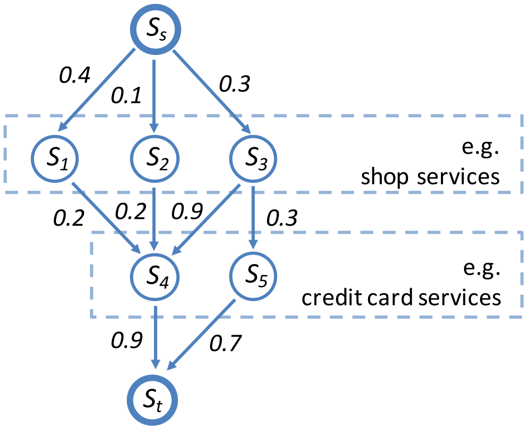

Fig. 2. An example of a service graph

interaction arcs starting from S_s and ending in S_t, the "best" service composition(s) $\overline{SC}_{Q,U}$, also taking the user profile U into account. The obvious question now is how to define the concept of best service composition.

First of all, let's make a step back and elaborate on the concept of service composition as a *sequence* of interactions. The core idea of the profile processor is to store the *history* of service compositions assigned to users in past requests, in order to be able to statistically estimate the ***probability*** $P_U(S_y|S_x)$ that a given service interaction (S_x, S_y) is suitable for a given user profile U. Basically, given a service S_x, we want to find the most likely service S_y that could follow for user U. This can be done by working at the level of the single keywords composing profiles.

For a generic user $U = \{k_1, \ldots, k_n\}$ and any $I \subseteq \{1, \ldots, n\}$, let $count_I(S_x)$ and $count_I(S_x, S_y)$ denote the number of times users characterized (possibly among others) by the set of keywords $\{k_i\}_{i \in I}$ have been successfully serviced by service S_x and service interaction (S_x, S_y), respectively. The probability $P_U(S_y|S_x)$ of successful service interaction (S_x, S_y) for user U can then be estimated using the well-known principle of inclusion-exclusion[4]:

$$P_U(S_y|S_x) \approx \frac{\displaystyle\sum_{e=1}^{n} (-1)^{e-1} \sum_{I \subseteq \{1,\ldots,n\}, |I|=e} count_I(S_x, S_y)}{\displaystyle\sum_{e=1}^{n} (-1)^{e-1} \sum_{I \subseteq \{1,\ldots,n\}, |I|=e} count_I(S_x)} \tag{2}$$

Unfortunately, computing Eq. 2 exactly requires exponential work so in our implementation we will recur to approximation (see, e.g., [18]) or even heuristic algorithms.

---

[4] We adopt the well known Markov chain approximation: in our context, the probability of choosing the next service depends only on the preceding service and not on the whole sequence of services that preceded it.

We are now ready to get back to our final goal and complete the picture. Building on the previous results, first of all we define the new ***weights*** w of the service graph SG_Q, taking into account for each service interaction (S_x, S_y):

(a) the weights w_Q relative to query Q (Eq. 1);
(b) the weights w_U relative to user profile U, defined on the basis of the *probability* $P_U(S_y|S_x)$ (Eq. 2).

$$w(S_x, S_y) = \alpha \cdot w_Q(S_x, S_y) + (1 - \alpha) \cdot w_U(S_x, S_y) \tag{3}$$

where $\alpha \in (0, 1)$ is a tunable parameter that can be freely adjusted in order to change the relative influence of Q and U (default is 0.5). As to $w_U(S_x, S_y)$, a first approximation could be to simply compute it as $w_U(S_x, S_y) = P_U(S_y|S_x)$. However, in the initial computations, history statistics are not sufficient to compute a significant probability, thus we choose to initially base it on a similarity between S_x, S_y and U (similarly as Eq. 1 did for Q):

$$w_U(S_x, S_y) = \beta \cdot min(kwsim(S_x, U), kwsim(S_y, U)) + (1 - \beta) \cdot P_U(S_y|S_x) \tag{4}$$

where β is a time-dependent parameter decreasing from 1 to 0, gradually giving strength to the probability $P_U(S_y|S_x)$.

Finally, we define a ***score*** of a service composition $SC_{Q,U} = \{(S_s, S_x), (S_x, S_y), \ldots, (S_z, S_t)\}$ by composing the weights of its single interactions as defined in Eq. 3:

$$score(SC_{Q,U}) = \varphi(w(S_s, S_x), w(S_x, S_y), \ldots, w(S_z, S_t)) \tag{5}$$

where φ is a composition function that, at comparable weight, privileges the shortest sequences (eg. a composition of a very large number of suitable services is not expected to be equally suitable). A first level of approximation is the product: $score(SC_{Q,U}) = \prod_{a \in SC_{Q,U}} w(a) = w(S_s, S_x) \cdot w(S_x, S_y) \cdot \ldots \cdot w(S_z, S_t)$. In this way, finding the "best" service composition $\overline{SC}_{Q,U}$ becomes a matter of finding the sequence of service interactions maximizing the score in Eq. 5. Since the service graph is a DAG (Directed Acyclic Graph) and the "score" of service composition cannot but decrease when extending a path, this computation can be efficiently performed (in linear time) using a slightly modified version of Dijkstra algorithm.

For instance, for our example in Fig. 2, $\overline{SC}_{Q,U} = \{(S_s, S_3), (S_3, S_4), (S_4, S_t)\}$, $score(\overline{SC}_{Q,U}) = 0.3 \cdot 0.9 \cdot 0.9$.

5 Conclusions

In this paper we have proposed an approach to compose services taking into consideration the user profile, leading to *user-aware service composition.* To this purpose, we rely on the SAPERE infrastructure, which has been enhanced by adding a profile processor that takes the user profile as input and proposes the best composition among potential ones.

In future work we will consider the concept of "semantic path" from complementary research areas [21, 20] for extending our semantic score computation method on a service graph; further, we aim at providing an effective implementation of our approach, and at evaluating it on some case studies.

Acknowledgement This work was supported by the project "Algorithms and Models for Building context-dependent Information delivery Tools" (AMBIT) co-funded by Fondazione Cassa di Risparmio di Modena (SIME 2013.0660).

References

1. Gregory D. Abowd et al. Cyberguide: A mobile context-aware tour guide. *Wireless Networks*, 3:421–433, 1997.
2. Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2:263–277, 2007.
3. Sonia Bergamaschi, Riccardo Martoglia, and Serena Sorrentino. Exploiting semantics for filtering and searching knowledge in a software development context. *Knowledge and Information Systems*, 45(2):295–318, 2015.
4. Claudio Bettini et al. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, 2010. Context Modelling, Reasoning and Management.
5. Cristiana Bolchini et al. Context modeling and context awareness: steps forward in the context-addict project. *Bulletin of the Technical Committee on Data Engineering*, 34:47–54, 2011.
6. Jenna Burrell and Geri K Gay. E-graffiti: evaluating real-world use of a context-aware system. *Interacting with Computers*, 14(4):301 – 312, 2002.
7. Giacomo Cabri et al. Location-dependent Services for Mobile Users. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems And Humans*, 33(6):667–681, 11 2003.
8. Giacomo Cabri, Mauro Leoncini, and Riccardo Martoglia. Ambit: Towards an architecture for the development of context-dependent applications and systems. In *Proc. of the 3rd Int'l Conference on Context-Aware Systems and Applications (ICCASA 14)*, pages 64–68. ICST, Brussels, Belgium, Belgium, 10 2014.
9. Gabriella Castelli et al. Engineering pervasive service ecosystems: the sapere approach. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 10(1):1, 2015.
10. Alfonso Garca de Prado and Guadalupe Ortiz. Context-aware services: A survey on current proposals. In *SERVICE COMPUTATION 2011, The 3rd Int. Conf. on Advanced Service Computing*, pages 104–109, 2011.
11. David Gelernter. Generative communication in linda. *ACM Transactions on Programming Languages and Systems*, 7(1):80–112, January 1985.
12. Stephen Gilmore, Jane Hillston, and Mirco Tribastone. Service composition for collective adaptive systems. In *Software, Services, and Systems*, pages 490–505. Springer, 2015.
13. Daniele Grassi, Anas Bouhtouch, and Giacomo Cabri. Inbooki: Context-aware adaptive e-books. In *Proceedings of the 2nd International Conference on Context-Aware Systems and Applications*, 11 2013.
14. Son N Han, Gyu Myoung Lee, and Noel Crespi. Semantic context-aware service composition for building automation system. *Industrial Informatics, IEEE Transactions on*, 10(1):752–761, 2014.
15. Paul Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

16. Amin Jula, Elankovan Sundararajan, and Zalinda Othman. Cloud computing service composition: A systematic literature review. *Expert Systems with Applications*, 41(8):3809–3824, 2014.
17. JH. Jun, DW. Lee, and KB. Sim. Realization of cooperative strategies and swarm behavior in distributed autonomous robotic systems using artificial immune system. *Procedings of the IEEE International Conference on Systems, Man, and Cybernetics. IEEE SMC '99*, 6:614–619, 1999.
18. Jeff Kahn, Nathan Linial, and Alex Samorodnitsky. Inclusion-exclusion: Exact and approximate. *Combinatorica*, 16(4):465–477, 1996.
19. Georgia M. Kapitsaki et al. Context-aware service engineering: A survey. *J. Syst. Softw.*, 82(8):1285–1297, August 2009.
20. Federica Mandreoli et al. Sri@work: Efficient and effective routing strategies in a PDMS. In *Proc. of WISE Conference*, pages 285–297, 2007.
21. Federica Mandreoli et al. Data-sharing P2P networks with semantic approximation capabilities. *IEEE Internet Computing (IEEE)*, 13(5):60–70, 2009.
22. Riccardo Martoglia. AMBIT: semantic engine foundations for knowledge management in context-dependent applications. In *The 27th International Conference on Software Engineering and Knowledge Engineering, SEKE 2015, Pittsburgh, PA, USA, July 6-8, 2015*, pages 146–151, 2015.
23. A.G. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
24. M.A. Munoz et al. Context-aware mobile communication in hospitals. *Computer*, 36(9):38 – 46, sept. 2003.
25. Gabrielle Peko, Ching-Shen Dong, and David Sundaram. Adaptive sustainable enterprises: A framework, architecture and implementation. In *Context-Aware Systems and Applications*, pages 293–303. Springer, 2014.
26. Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 32–43, New York, NY, USA, 2000. ACM.
27. Jinghai Rao and Xiaomeng Su. A survey of automated web service composition methods. In *Semantic Web Services and Web Process Composition*, pages 43–54. Springer, 2005.
28. Hong-Linh Truong and Schahram Dustdar. A survey on context-aware web service systems. *International Journal of Web Information Systems*, 5:5–31, 2009.
29. Pallapa Venkataram and M Bharath. A method of context-based services discovery in ubiquitous environment. In *Context-Aware Systems and Applications*, pages 260–270. Springer, 2014.
30. Danny Weyns et al. The agent environment in multi-agent systems: A middleware perspective. *Multiagent and Grid Systems*, 5(1):93–108, 2009.
31. Franco Zambonelli, Nicola Bicocchi, and Damiano Fontana. Towards a human-aware operating system. In *9th International Conference on Body Area Networks, BODYNETS 2014, London, Great Britain, September 29 - October 1, 2014*, 2014.
32. Franco Zambonelli et al. Developing pervasive multi-agent systems with nature-inspired coordination. *Pervasive and Mobile Computing*, 17:236–252, 2015.