Towards User-Aware Service Composition

Giacomo Cabri(✉), Mauro Leoncini, Riccardo Martoglia, and Franco Zambonelli

Università di Modena e Reggio Emilia, Modena, Italy
{giacomo.cabri,mauro.leoncini,riccardo.martoglia, franco.zambonelli}@unimore.it

Abstract. Our everyday life is more and more supported by the information technology in general and by specific services provided by means of our electronic devices. The AMBIT project (Algorithms and Models for Building context-dependent Information delivery Tools) aims at providing a support to develop services that are automatically tailored based on the user profile. However, while the adaptation of the single services is the first step, the next step is to achieve adaptation in the *composition* of different services. In this paper, we explore how services can be composed in a user-aware way, in order to decide the composition that better meets users' requirements. That is, we exploit the user profile not only to provide her with customized services, but also to compose them in a suitable way.

Keywords: 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]).

P.C. Vinh et al. (Eds.): ICTCC 2016, LNICST 168, pp. 11–21, 2016.
DOI: 10.1007/978-3-319-46909-6_2

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 the AMBIT (Algorithms and Models for Building context-dependent Information delivery Tools) project[1] 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 Sect. 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

Many challenging research problems in the area of user-awareness or more in general of context-awareness and context-dependent service delivery have been addressed in an already impressive body of work. These efforts are largely motivated by the astounding growth of the mobile device market and the need to support the evolution of the traditional Web into the so-called "Web of Things and Services". For some comprehensive surveys the reader is referred to [10,17,19,28] and the many references contained therein.

In the context of this huge active area, our research interests are primarily directed towards a general model of user profile and context, and its exploitation in order to provide better customized service compositions and services. First of all, it must be pointed out that the meaning associated to the word "context" is not unique and usually depends on the particular application(s) that researchers have in mind. For instance, in automotive applications context clearly refers to

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

driving environment, while in health-care applications context is likely to refer to environmental, physiological and behavioral variables that apply to a specific patient. As another example, in online advertising the context is essentially the page (say, the page of an on-line newspaper) where the commercial is to be displayed, namely, its contents, the prevailing sentiment, etc.

Instead, we are interested in a general notion of user profile and context, that can possibly include all of the above and much more. There are a handful of research contributions that are relevant to this narrow area. Bettini et al. [4] take different types of context information into account (physical, computational, user context) and approach the problem of modeling and representation from within the perspective of automated reasoning. The rationale is that modeling real-life situations requires the ability to process basic context facts and reasoning makes it possible to attain pieces of information that are appropriate for use by context-aware applications. This highly cited survey is also important since it covers some prominent approaches to context modeling (object-role based, spatial models, ontology-based).

Han et al. [14] proposed a SOA-based approach to build automation systems with devices that are provided by a description that contains context information. This information is managed by a composition engine to coordinate appropriate devices/services based on the context, composition plan, and predefined policy rules. Another approach that is interesting because it takes into consideration different aspects is the one proposed by Peko et al. [25]. In fact, they start from the consideration that enterprises must be adaptable to the changes in the context they operate, but at the same time they must be sustainable in terms of economic, environmental, societal, and cultural concerns. Their approach considers the enterprises' context in terms of strategy, organization, process, and information. With respect to these works, our approach aims at exploiting a more general set of information (not only service/device enterprise-oriented) and is designed to address not only specific fields such as building automation or enterprise management, but a wider range of services.

The survey [5] covers various research works related to context modeling and awareness within the Context-ADDICT project of the Politecnico di Milano (see http://poseidon.ws.dei.polimi.it/ca/). They propose to design a context management system to be placed aside what they call the "operational system". While the latter is application dependent, the context management system is not, and exhibits a hierarchical structure in terms of observable (i.e., external) parameters that have a symbolic internal representation within a context schema. We will consider this separation of concerns also for building the bases of our user-aware service composer.

In [29] the authors propose a context-based approach for service discovery. The focus on services makes this work interesting for our purpose, even if we do not focus on the discovery. We will evaluate their formal definition of the context and possibly make it more general.

An approach that is interesting because it takes into consideration different aspects is the one proposed by Peko et al. [25]. In fact, they start from the consideration that enterprises must be adaptable to the changes in the context they operate, but at the same time they must be sustainable in terms of economic, environmental, societal, and cultural concerns. They approach consider the enterprises' context in terms of strategy, organization, process, and information. This global approach can influence our work and we could extend it to be more general.

Venkataram and Bharath [29] propose a context-based approach for service discovery. The focus on services makes this work interesting for our purpose, even if we do not focus on the discovery. We will evaluate their formal definition of the context 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] which reifies the SAPERE environment in terms of a lightweight software support, enabling a SAPERE node to be installed in tablets and smartphones.

Each SAPERE node wishing to participate to the SAPERE ecosystem should host a local tuple space [11], to act as a local repository of LSAs for local services, and a local eco-laws engine. The LSA-space of each node is connected with a limited set of neighbor nodes based on spatial proximity relations. From the viewpoint of individual service, the middleware provides an API to access the local LSA space, to advertise themselves (via the injection of an LSA), and to support the services' need of continuously updating their LSAs. In addition, such API enables services to detect local events (as the modifications of some LSAs) or the enactment of some eco-laws on available LSAs.

Eco-laws are realized as a set of rules embedded in SAPERE nodes. For each node, the same set of eco-laws applies to rule 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).

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

A typical example of composition is an e-commerce transaction. Let us consider a user that aims at buying a good and requires the home delivery. She can exploit at least three services of different kind. The first one is the *shop* service, which enables the user to know which the goods on sale are, along with a set of further information such as the price, the availability, the size, and so on. The second service is related to the payment of the chosen good, for instance, a *credit card transaction* service. Finally, a *delivery agency* service is in charge of the actual delivery of the good to the users' home. Currently, these three services are provided by online e-commerce Web site as a whole, but in the future we envision that many services of each kind exist, and they can be composed to provide the higher-level service. In the latter case, the user preferences can influence the composition.

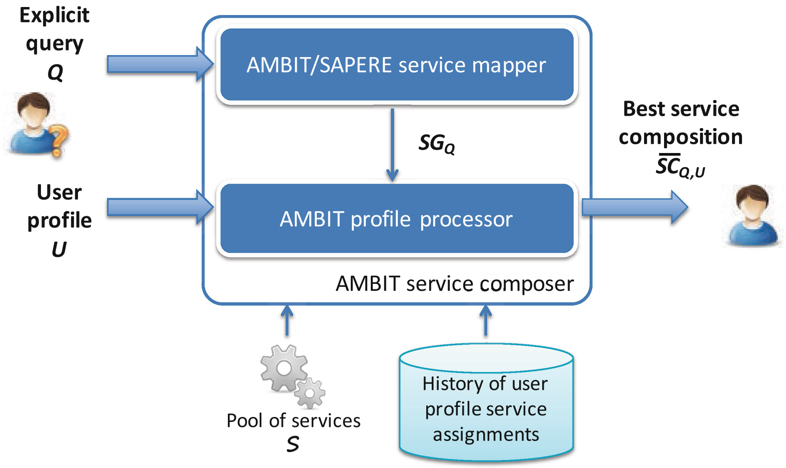

Fig. 1. High-level architecture of the AMBIT service composer

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 (Fig. 1).

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:

$$w_Q(S_1, S_2) = min(kwsim(S_1, Q), kwsim(S_2, Q)) \quad (1)$$

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.

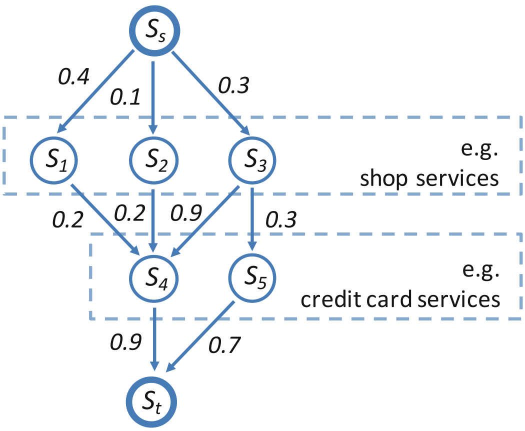

Fig. 2. An example of a service graph

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 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[2]:

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

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.

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):

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

[2] 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.

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:

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

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 [20, 21] 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. Abowd, G.D., et al.: Cyberguide: a mobile context-aware tour guide. Wirel. Netw. **3**, 421–433 (1997)
2. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. Int. J. Ad Hoc Ubiquit. Comput. **2**, 263–277 (2007)

3. Bergamaschi, S., Martoglia, R., Sorrentino, S.: Exploiting semantics for filtering and searching knowledge in a software development context. Knowl. Inf. Syst. **45**(2), 295–318 (2015)
4. Bettini, C., et al.: A survey of context modelling, reasoning techniques. Pervasive Mob. Comput. **6**(2), 161–180 (2010). Context Modelling, Reasoning and Management
5. Bolchini, C., et al.: Context modeling and context awareness: steps forward in the context-addict project. Bull. Techn. Committee Data Eng. **34**, 47–54 (2011)
6. Burrell, J., Gay, G.K.: E-graffiti: evaluating real-world use of a context-aware system. Interact. Comput. **14**(4), 301–312 (2002)
7. Cabri, G., et al.: Location-dependent services for mobile users. IEEE Trans. Syst. Man, Cybern.-Part A: Syst. Hum. **33**(6), 667–681 (2003)
8. Cabri, G., Leoncini, M., Martoglia, R.: AMBIT: towards an architecture for the development of context-dependent applications and systems. In: Proceedings of the 3rd International Conference on Context-Aware Systems and Applications (ICCASA 2014), pp. 64–68. ICST, Brussels, October 2014
9. Castelli, G., et al.: Engineering pervasive service ecosystems: the sapere approach. ACM Transactions on Autonomous and Adaptive Systems (TAAS) **10**(1), 1 (2015)
10. de Prado, A.G., Ortiz, G.: Context-aware services: a survey on current proposals. In: The 3rd International Confernece on Advanced Service Computing, Service Computation 2011, pp. 104–109 (2011)
11. Gelernter, D.: Generative communication in linda. ACM Trans. Program. Lang. Syst. **7**(1), 80–112 (1985)
12. Gilmore, S., Hillston, J., Tribastone, M.: Service composition for collective adaptive systems. In: Nicola, R., Hennicker, R. (eds.) Software, Services, and Systems. LNCS, vol. 8950, pp. 490–505. Springer, Heidelberg (2015). doi:10.1007/978-3-319-15545-6_28
13. Grassi, D., Bouhtouch, A., Cabri, G.: Inbooki: context-aware adaptive e-books. In: Proceedings of the 2nd International Conference on Context-Aware Systems and Applications, November 2013
14. Han, S.N., Lee, G.M., Crespi, N.: Semantic context-aware service composition for building automation system. IEEE Trans. Ind. Inf. **10**(1), 752–761 (2014)
15. Jaccard, P.: 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. Jula, A., Sundararajan, E., Othman, Z.: Cloud computing service composition: a systematic literature review. Expert Syst. Appl. **41**(8), 3809–3824 (2014)
17. Jun, J.H., Lee, D.W., Sim, K.B.: Realization of cooperative strategies and swarm behavior indistributed autonomous robotic systems using artificial immune system. In: Procedings of the IEEE International Conference on Systems, Man, and Cybernetics, IEEE SMC 1999, vol. 6, pp. 614–619 (1999)
18. Kahn, J., Linial, N., Samorodnitsky, A.: Inclusion-exclusion: exact and approximate. Combinatorica **16**(4), 465–477 (1996)
19. Kapitsaki, G.M., et al.: Context-aware service engineering: a survey. J. Syst. Softw. **82**(8), 1285–1297 (2009)
20. Mandreoli, F.: Sri@work.: Efficient and effective routing strategies in a PDMS. In: Proceedings of WISE Conference, pp. 285–297 (2007)
21. Mandreoli, F., et al.: Data-sharing P2P networks with semantic approximation capabilities. IEEE Internet Comput. (IEEE) **13**(5), 60–70 (2009)

22. Martoglia, R.: 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, 6–8 July 2015, pp. 146–151 (2015)
23. Miller, A.G.: Wordnet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
24. Munoz, M.A., et al.: Context-aware mobile communication in hospitals. Computer **36**(9), 38–46 (2003)
25. Peko, G., Dong, C.-S., Sundaram, D.: Adaptive sustainable enterprises: a framework, architecture and implementation. In: Vinh, P.C., Alagar, V., Vassev, E., Khare, A. (eds.) ICCASA 2013. LNICST, vol. 128, pp. 293–303. Springer, Heidelberg (2014). doi:10.1007/978-3-319-05939-6_29
26. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The cricket location-support system. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, MobiCom 2000, pp. 32–43. ACM, New York (2000)
27. Rao, J., Su, X.: A survey of automated web service composition methods. In: Cardoso, J., Sheth, A. (eds.) SWSWPC 2004. LNCS, vol. 3387, pp. 43–54. Springer, Heidelberg (2005). doi:10.1007/978-3-540-30581-1_5
28. Truong, H.-L., Dustdar, S.: A survey on context-aware web service systems. Int. J. Web Inf. Syst. **5**, 5–31 (2009)
29. Venkataram, P., Bharath, M.: A method of context-based services discovery in ubiquitous environment. In: Vinh, P.C., Alagar, V., Vassev, E., Khare, A. (eds.) ICCASA 2013. lnicst, vol. 128, pp. 260–270. Springer, Heidelberg (2014). doi:10.1007/978-3-319-05939-6_26
30. Weyns, D., et al.: The agent environment in multi-agent systems: A middleware perspective. Multiagent Grid Syst. **5**(1), 93–108 (2009)
31. Zambonelli, F., Bicocchi, N., Fontana, D.: Towards a human-aware operating system. In: 9th International Conference on Body Area Networks, BODYNETS 2014, London, Great Britain, 29 September–1 October, 2014
32. Zambonelli, F., et al.: Developing pervasive multi-agent systems with nature-inspired coordination. Perv. Mob. Comput. **17**, 236–252 (2015)

http://www.springer.com/978-3-319-46908-9

Nature of Computation and Communication
Second International Conference, ICTCC 2016, Rach Gia, Vietnam, March 17-18, 2016, Revised Selected Papers
Vinh, P.C.; Barolli, L. (Eds.)
2016, XIII, 398 p. 175 illus., Softcover
ISBN: 978-3-319-46908-9