

Available online at <http://www.mecs-press.net/ijwmt>

## Compose Real-world Service With Context

Guoping Zhang<sup>a,\*1</sup>, Jiazheng Tian<sup>a,\*2</sup>

<sup>a</sup> School of Computer & Communication Engineering, China University of Petroleum, Dong Ying, China

<sup>b</sup> Second affiliation, Address, City and Postcode, Country

---

### Abstract

Advances in the areas of smart embedded devices, computing and networking are leading to the emergence of the Internet of Things (IoT) composed of millions of heterogeneous devices. These devices will be interconnected, providing and consuming information available on the network and cooperate. As the improvement of the performance of industrial processes, the service-oriented approach can be adopted for smart embedded devices. Each device can offer its functionality as services. In such infrastructures, traditional web service compositions can not adapt to the dynamic environment for the Internet of Things. In this paper, we propose an approach to compose services with context. It can not only adapt to dynamic environment for the Internet of Things, but also satisfy the user's preferences. We first build context ontology to describe the scenario for user. According to context information, the service composition process is presented.

**Index Terms:** Internet of Things; Real-World Service; Service Composition; Context Ontology

© 2012 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science

---

### 1. Introduction

In recent years, smart embedded devices have developed rapidly. They are becoming cheaper, smaller and more powerful. According to the Internet of Things vision [1], the most devices will have communication and computation capabilities, which they can interact and cooperate with their surrounding environment.

Along with the development of SOA (Service-Oriented Architecture), service technical gradually become mainstream technologies for information integration of in the Internet environment. As these devices need to interoperate, the integration of devices into the business IT-landscape through SOA [2][3] is a promising approach to connect physical objects and to make them available to IT-systems. Each device should offer its functionality as standard services, while in parallel it can discover and invoke other services on-demand from other devices.

In the future Internet, real-world devices will be able to offer their functionality by SOAP-based web service or RESTful APIs [4], enabling other components to interact with them dynamically.

As the real-world devices are related directly to the physical world, the service that they provided is often referred as real-world service.

\* Corresponding author:

E-mail address: <sup>\*1</sup> [zhanggp@upc.edu.cn](mailto:zhanggp@upc.edu.cn); <sup>\*2</sup> [tjzh519@126.com](mailto:tjzh519@126.com)

Compared with the traditional enterprise services, there are some differences. Firstly, real-world services provide real-time data about the physical world, not virtual entities. Secondly, real-world services are deployed on resource constrained devices, with limited computing, communication and storage capabilities. Additionally, real-world services are found in highly dynamic environments where devices and their underlying services constantly degrade, vanish, and possibly reappear. This implies the need for automated, immediate discovery of devices and services as well as their dynamic management, and a reasonable service composition method based on specific application environment to meet user's request.

Our work aims at taking context information to guide composition so that we can get the best satisfying composed service. In Section 2, we discuss related work. In Section 3, we build context ontology. In Section 4, we propose a context-based web service composition approach. Finally, Section 5 gives conclusion of our work.

## 2. Related Work

Currently, service composition has become a hot topic in the industry and academic field. Researchers have proposed several service composition approaches. According to the technical they rely on, we can divide them into two categories. One category defines the business process to support service composition, its composition path is predefined, so it is difficult to meet user's needs. The other based on semantics or domain related to satisfy user's personalized demand. But these two categories did not consider the environment information in which services, making service composition lack of adaptability. Therefore, the researchers introduced context concept, and proposed service composition based on context information [5][9], but did not give specific composition method. As for the real-world services that are directly linked to the physical world, service composition should adapt to the dynamic changes of devices or resources, and service selection should depend on environment information in which services and resources that services need. In addition, we should consider users demand on non-functional properties of services.

Our work is different from traditional compositions, we take into account context information to filter inappropriate services and find the best solution for the given scenario.

## 3. Context Ontology

Context is information that qualifies the physical world, and can help in both reducing the number of services returned to the user, as well as in finding the most appropriate services for the current environment. The definition of context provided by Dey & Abowd is widely accepted, "Context is any information that can be used to characterize the situation of an entity [6]." An entity is a person, place, computational entity, or object which is considered relevant for determining the behavior of an application.

When dealing with context information, it is always a challenge to describe contextual facts and interrelationships in a precise and traceable manner. Ontology is a formal and explicit specification of a conceptualization. It provides a vocabulary for representing domain knowledge and for describing specific situations in a domain. Therefore, context can be defined as the semantic representation of real-world knowledge in a machine understandable format through an ontology-based approach. In addition, we choose the Web Ontology Language (OWL) to model context ontology for expressiveness and extensibility.

Comprehensive research institutions have made the context of the body design [7][8], combined with the characteristics of the IoT environment, we propose a hierarchical ontology model for context. An overview of the proposed context ontology is given in Fig.1.

Hierarchical structure design can reduce the range of context information and enable developers to reuse the upper domain ontology. More importantly, the use of hierarchical design method can reduce the burden that the lower specific sub-domain handles context information, especially for the IoT.

Generic top-level ontology captures all common knowledge for the IoT. It provides basic public concepts collections of the different domains. It is shared by the ontology from the different domains. The low-level ontology defines concepts and properties for each sub-domain.

To fit our requirements, the top-level ontology is divided into three categories around which we built our ontology. They are described as follows.

Computing context describes the hardware and software of a specific device. This includes network connectivity, available memory and bandwidth, but also information about nearby resources such as printers, displays, and workstations.

Physical Environment is mainly composed of time, location information and environmental conditions, such as temperature and lighting.

User context mainly refers to the expectation of the developer in terms of how should the discovered service perform, such as price, Response Time, availability and reliable.

After building context ontology, we can describe scenario that a person driving on the highway in a windy condition.

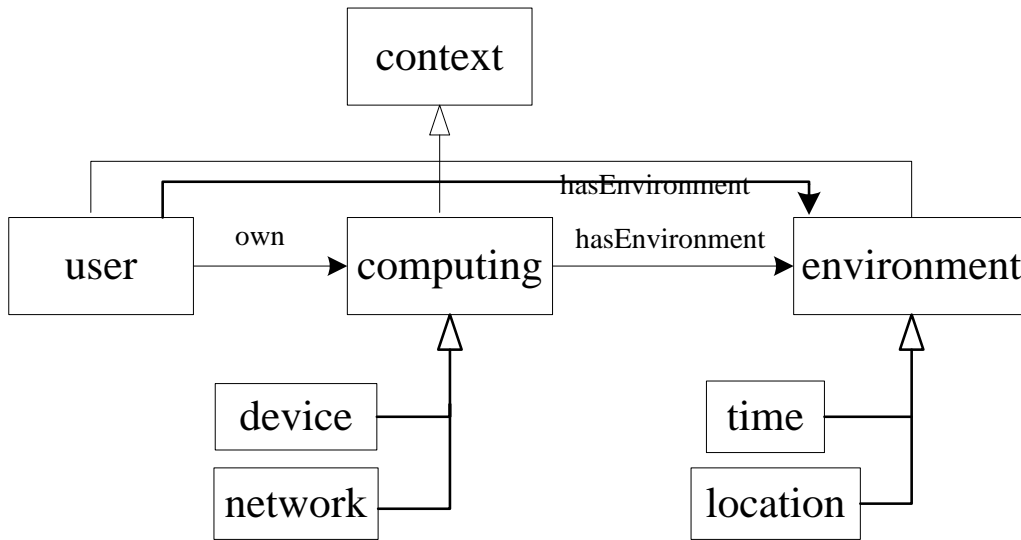


Fig. 1. Context ontology overview

#### 4. Web Services Composition

Our goal is to use context information to optimize service composition so as to improve the quality of the composed service. To evaluate service quality, like traditional web service composition, we need to consider non-functional attributes of services, such as the cost of services and the response time of services. As to the real-world services provided by smart devices, we should also consider various quality factors (e.g. Network Speed, Memory Capacity) to maximize user's satisfaction.

We propose a context-based partial weight approach to let context information exert an impact on service composition. This is different from traditional compositions, in which the composed services must obey constraints defined by context, such as network bandwidth.

According to the context of ontology model we defined, we divide composition process into two sub-processes. First, we use device context to filter the appropriate services. Second, we select the best service that satisfies user's needs according to user's expectation to the quality of service.

#### 4.1. The Context Matching Between Service And Device

Suppose candidate service set  $S$  is composed of  $n$  services, denoted by  $S = \{s_1, s_2, \dots, s_n\}$ , their context denoted  $sc_i$ ,  $i=1,2,\dots,n$ . There are  $m$  devices in the network, denoted  $D = \{d_1, d_2, \dots, d_m\}$ , their corresponding context denoted  $dc_i$ ,  $i=1, 2, \dots, m$ .

The impact that device make on the service showed mainly in two aspects, one is the device's internal resources (such as CPU, memory, etc.) should meet the required that the service instance runs; The other, if the service runs on different devices, the devices resources (such as network bandwidth) will limit communication between services and also affect the choice of composite services.

We first consider the situation that service  $s_i$  and its output and input exist on only one device  $d_s$ . When the attribute of service context is the same as attribute of device context, the attribute value of service context should not greater than the device's. The relationship can be expressed as follows.

$$sc_i.value_j \leq dc_s.value_j, j \in N \quad (1)$$

If there are  $n$  services running on one device, for the attribute  $k$ , the attribute total value of  $n$  services should not greater than the attribute value of the device, such as the memory of the device. The relationship can be expressed as follows.

$$\sum_{i=1}^n sc_i.value_k \leq dc_s.value_k, k \in N. \quad (2)$$

#### 4.2. The Context Matching Between Service And User

In user context, we defined user preferences to non-functional properties of service, including the preference attributes and attribute values. Among them, the preferred service attributes describe that some non-functional properties customers care about, such as the price of service and response time of service. Users need to consider the function and operation environment of the task, and select the quality factor. The preference of non-functional attributes, denoted by  $w$  ( $0 < w < 1$ ), our method to the weight setting of the quality factor is based on rank-order centroid method [10] [11].

The weights of quality factors are determined according to the priority of quality dimensions and quality factors. In rank-order centroid method, the weight  $w_j$  of the  $j$ th ordered factor among  $n$  quality factors can be expressed as follows.

$$w_j = \frac{1}{n} \sum_{k=j}^n \frac{1}{k} (j = 1, 2, \dots, n) \quad (3)$$

We rank quality factors according to user's preference to the non-functional attributes of services. In order to select the best service that meet the user needs, we should consider several non-functional attributes of service that user prefers. Therefore, it can be seen as multiple attributes decision making (MADM) problems. We adopt simple additive weight (SAW) method, the principle is the assessment value of each candidate plan multiplied by the corresponding weight, the sum of these products called the candidate's score. According to scores, compare the merits of the candidate plan, and then choose the best solution.

After filtering inappropriate service by device context matching, we get a new candidate service set, denoted by  $Se = \{s_1, s_2, \dots, s_m\}$ , suppose user focus on  $r$  attributes of service  $s_i (1 \leq i \leq m)$ , the value of attribute denoted repetitively,  $sc_i.value_1, sc_i.value_2, \dots, sc_i.value_r$ . To descript facilitatly, we denoted  $v_{i1}, v_{i2}, \dots, v_{ir}$ , then the matrix that composed by the context of service in  $Se$  can be expressed as follows,

$$SCM = \begin{pmatrix} v_{11} & \cdots & v_{1r} \\ \vdots & \ddots & \vdots \\ v_{m1} & \cdots & v_{mr} \end{pmatrix}. \quad (4)$$

$v_{ij}$  expresses the  $j$ th attribute value of the service  $s_i$ . Because of attribute value has different ranges and unit, we use SAW method [12], SAW divide the criteria into cost and effectiveness criteria. If the lower the price of the service is, the more able to meet the needs of users, likes this property using the cost criteria method to

standardized, that is  $x_{ij} = 1 - \frac{v_{ij}}{v_j^*}$ , here  $v_j^* = \max_i(v_{ij})$ . However the higher the service reliability is,

the better meet the needs of users, this property using effectiveness criteria standardized, that is  $x_{ij} = \frac{v_{ij}}{v_j^*}$ . After standardized, the matrix  $SCM'$ , can be expressed as follows,

$$SCM' = \begin{pmatrix} x_{11} & \cdots & x_{1r} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mr} \end{pmatrix}, \quad (5)$$

According to the SAW principle,  $Score = SCM' \times W^T$ , after computing,  $Score = (\sum_{j=1}^r x_{1j}w_j, \sum_{j=1}^r x_{2j}w_j, \dots, \sum_{j=1}^r x_{mj}w_j)$ ,  $\max_i(\sum_{j=1}^r x_{ij}w_j, 1 \leq i \leq m)$  means the best service that meets users needs.

In this manner, we use context information on service composition to form an optimization problem. Upon a given scenario, we can find the composed service with highest score while satisfying constrain defined in the context.

## 5. Conclusion

In a future highly populated by networked embedded devices, composing real-world services that can be dynamically included in enterprise applications will be a challenging task, because the real-world services are different from traditional services. In this paper, we first analyzed the characteristic of real-world services, and also show how context is important for real-world services. Then we presented a basic, generic ontology for the description of context information. Finally, we have presented here an approach that optimizes service composition for developers. In the service composition process, we use context information to filter inappropriate services, then select the best service that user satisfy according user's preference to quality of service.

Until now we have completed the definition of context ontology and service composition process. In the future, we will analyse the complexity of this problem, and we will design an algorithm for almost optimal solution. And in the end we will realize our service composition approach based on Socrates Integration Architecture [2]. Finally, we will deploy the approach in real-world trials in order to better understand the uses and limitations.

## References

- [1] E. Fleisch and F. Mattern, *Das Internet der Dinge: Ubiquitous Computing und RFID in der Praxis: Visionen, Technologien, Anwendungen, Handlungsanleitungen*. Springer-Verlag, 2005
- [2] Patrik Spiess, Stamatis Karnouskos, et al, "SOA-based Integration of the Internet of Things in EnterpriseServices," 2009 IEEE International Conference on Web Services, pp. 968–975, doi: 10.1109/ICWS.2009.98
- [3] Luciana Moreira Sa de Souza, Patrik Spiess, Dominique Guinard, Moritz Kohler, Stamatis Karnouskos, and Domnic Savio, "SOCRADES: A web Service Based Shop Floor Integration Infrastructure," C.Floerkemeier et al. (Eds.): IOT2008, LNCS4952, pp.50–67
- [4] D. Guinard and V. Trifa, "Towards the Web of Things: Web Mashups for Embedded Devices," Proc. Workshop Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM'09), Apr. 2009.
- [5] Dominique Guinard<sup>1</sup>, Vlad Trifa<sup>1</sup>, Patrik Spiess, Bettina Dober<sup>1</sup>, and Stamatis Karnouskos. Discovery and On-Demand Provisioning of Real-World Web Services, 2009 IEEE International Conference on Web Services, DOI 10.1109/ICWS.2009.94, 583-590
- [6] A.K. Dey, G.D. Abowd, "Towards a Better Understanding of Context and Context-Awareness", in Proc. of Workshop on The What, Who, Where, When, and How of Context-Awareness, Conference on Human Factors in Computing Systems, The Hague, The Netherlands, 2000.
- [7] Tam Van Nguyen, Wontaek Lim, Huy Anh Nguyen, Deokjai Choi and Chilwoo Lee, Context Ontology Implementation for Smart Home. ICUT 2007.
- [8] Davy Preuveneers, Jan Van den Bergh, Dennis Wagelaar, Andy Georges, Towards an extensible context ontology for Ambient Intelligence, 2004, pp.148-159.
- [9] Sen Luo, Bin Xu, Kewu Sun, Compose Real Web Services with Context, 2010 IEEE International Conference on Web Services, ICWS.2010.63, 630-631.
- [10] F. H. Barron and B. E. Barrett, "Decision quality using ranked attribute weights", *Management Science*, vol.42, no.11, November 1996, pp.1515–1523.
- [11] Heejung Chang, Kangsun Lee, Quality-Driven Web Service Composition for Ubiquitous Computing Environment, 2009 International Conference on New Trends in Information and Service Science. DOI 10.1109/NISS.2009.117, pp. 156-161.
- [12] Lei Dang, Xiaoyong Hui, Mingshu Li. An Approach to Dynamic Service Composition Based on Context Negotiation. *Journal of Computer Research and Development*. ISSN 1000-1239/CN, 11-1777/TP45(11). PP. 1902-1910. 2008.(in Chinese)