Service Creation and Composition for Realization On Service-Oriented Architecture

Chi-Lu Yang[1,2], Yeim-Kuan Chang[1], Chih-Ping Chu[1]
[1] Department of Computer Science and Information Engineering, National Cheng Kung University
[2] Innovative DigiTech-Enabled Applications & Service Institute, Institute for Information Industry
[1]Tainan, Taiwan R.O.C.
[2]Kaohsiung, Taiwan R.O.C.
stwin@iii.org.tw, ykchang@mail.ncku.edu.tw, chucp@csie.ncku.edu.tw

Abstract - *Applying ICT to assist daily activities and interests is already a worldwide trend. Through software and network techniques, computers could remotely provide various services. Up to now, Service-Oriented Architecture (SOA) has become one of the most popular techniques to realize these daily services in various areas. Therefore, we would need systematic methods for services creation from a service-centric viewpoint. In this paper, we first illustrated the progress of service realization. The interoperability of service creation and realization along its life cycle were carefully explained. With methodical sequences, domain-specific services can be well created and designed. The home-care services were demonstrated in the case study. These services were verified by a pilot trial in a real environment. The experimental results showed the usability and effectiveness. We believe that a successful experience on service realization is worthy to be shared.*

Keywords: SOA design, service creation, service composition, service verification, digital home-care

1. Introduction

Using Information and Communication Technologies (ICT) to create innovative services models, many enterprises have continuously become famous worldwide, such as YouTube, WRETCH, Google, Amazon, among many others. Their models have successfully provided services to promote business values over the Internet. One of their common features is that they had applied specific information techniques with the network techniques.

Applying ICT to assist daily activities and interests, such as medical care, lifestyle, traffic, education, and entertainment is already a worldwide trend. Through software and network techniques, computers could remotely provide various services. Up to now, Service-Oriented Architecture (SOA) has become one of the most important techniques to realize these daily services in various areas. For example, the SOA-based system was proven to effectively enhance the homecare services [5], [11]. With SOA, various stakeholders could be linked into similar service processes. Moreover, these processes are closely conjoined with their services.

Therefore, we would like to determine how/what services would be realized by SOA techniques. The solutions are inquired from a service-centric viewpoint. In this paper, we illustrate the progress of service realization and share our experiences from a case study at the same time. In Section 2, the SOA principles and challenges are introduced. In Section 3, the interoperability of service creation and realization are proposed along its life cycle. The representations of service elements are even illustrated, since we need large communication with various providers during service creation. A case study on realizing home-care services is carefully explained in Section 4. A pilot trial of the case study is set up for verification. The experiments and their results are discussed in Section 5. The summary of the study is then given in Section 6.

2. Related Work

Service-Oriented Architecture (SOA) is a software architectural style for realizing and constructing business services, which are composed by components as services [1]-[3]. Service-oriented technology could expand information and communication technology (ICT) to provide various services, which frequently require a large amount of data exchange. SOA also separates services into distinct units such as components or modules, which can be deployed over the Internet and can be reused to compose new applications. By SOA, services can be delivered to end-users over the Internet.

The architecture of SOA is clearly layered out. Business services could thus be clearly identified and layered in SOA. Business services are also created and composed by various software components by SOA. The typical layers of SOA are business process layer, service and application layer, and technical layer [4]-[5].

Furthermore, the general architectural principles figure out the ground rules of SOA for development, deployment, and maintenance [4]-[5]. The four ground rules in this study are service modeling, deliverability, compliance to standards and reusability. These are described as follows:

- Service modeling - Service definition and creation, deployment and delivery, monitoring and tracking, service concept, key performance indicators (KPI) definition, and so on.
- Deliverability - A service on SOA should be delivered via the Internet. The charged fee would be accounted for by the service providers.
- Compliance to standards – A large number of messages is frequently exchanged through the SOA platforms. These exchanged messages will extend the SOA capability and result in significant issues for standardization, identification, authorization, security, privacy, and so on.
- Reusability - A segment of the service might be reusable to compose new services. In other words, components or modules in SOA would be reused in various business processes and even mobile services.

In addition, the specific principles for service design and creation are categorized into two types. The first type includes specific design guidelines of SOA for service providers. The second type deals with the interaction between the service providers and consumers. They are described as follows:

- Service abstraction - Services are logically hidden from the outside world, beyond what is described in the service contract.
- Service autonomy - Services have control over the business processes they encapsulated.
- Service encapsulation - Various services in the Internet are consolidated with Web services under the SOA platform.
- Service composition - Collections of units of services can be coordinated and combined to create services.
- Service discoverability - Services are designed to be accessible to the public, they can thus be found and accessed via available discovery mechanisms.
- Service loose coupling - Services maintain a relationship that minimizes dependencies on one another.
- Service optimization - High-quality services are generally considered more than low-quality ones.
- Service contract - Services are attached to the communicable agreements, and are defined in service description documents.

Building SOA is not only a technical challenge, but also a business challenge. In the visions of SOA, relationships between service consumers and providers are not tightly stipulated. Their relations are loose coupling [6]. Thus, consumer services are not forcefully influenced by the changes made by the providers. Moreover, consumer service interacts with the service provider based on the service contract. Thus, negotiating Service Level Agreements (SLA) is even a critical issue. The SLA should even satisfy some general and specific principles.

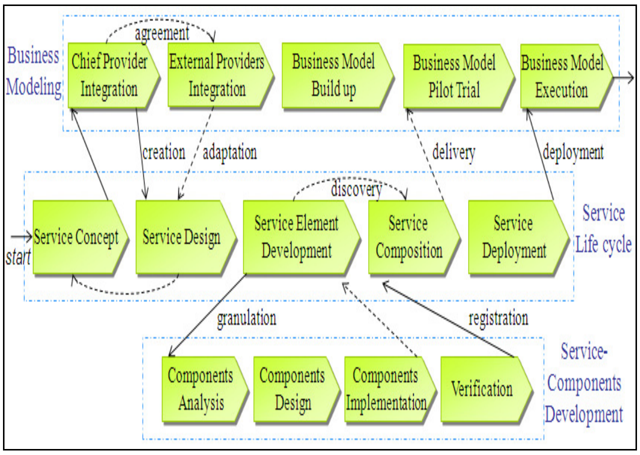

Fig. 1 Service Development Life Cycle

Another constraint is that SOA applications almost have to be used in a distributed environment [7]-[8]. That means end-users and service providers are distributed geographically. Services in SOA are delivered via the Internet. Unfortunately, SOA provides an environment that is convenient for hackers and intruders [9]-[10]. Web service is one of the most important ways to implement SOA. The relevant techniques are Extensible Markup Language (XML) [13], Web Services Description Language (WSDL) [14], Simple Object Access Protocol (SOAP) [15], and Universal Description, Discovery and Integration (UDDI) [16].

3. Service Creation

Service creation is a critical start when we would like to provide services by SOA. A service is composed of various elements, which are provided by different providers. These providers are even geographically distributed. If they are connected with the SOA platform, they will have chance to cooperatively create new services. In order to know how a service is realized, the life cycle of a service will be explored along business modeling and component development in the following sub-section. The representation of service elements will be also introduced.

3.1 Service Development Life Cycle

A service is intensively related with its business model since the service will be realized for serving someone/something in the real world. The life cycle of services will approach business modeling. These phases are defined as service concept, service design, service element development, service composition, and service deployment. The development life cycle of service is shown in Fig. 1. Furthermore, the service elements are closely associated with software lifecycles.

The service concept is initiated by interviewing with domain experts. Its draft will be used for triggering internal integration of chief provider, who is the key provider in the service concept.

The service design, which is constructed by service designer, will show service scenarios step by step. Each step corresponds to a service element. The service element would therefore possibly be reused. The service scenarios are extended based on the service concept and are adapted by external providers. These providers will be included and organized if the scenario requires them to provide specific service elements. The chief and external providers cooperate by following service level agreements (SLA). The agreements should at least include the service elements and charge flow. The regular routines among these providers will be extremely large after the business model is executed in a real environment. Therefore, they need software and internet to manage these routines.

Service elements are granularly developed to be components, which are even encapsulated to web services and are composed to provide services on the Internet. The service flow and charge flow are hidden in these components. A tool engine for service management is also indeed essential.

Since these derived components are analyzed and designed by object-oriented methods, they could characteristically be reused to compose new services. Verification and registration of the components are necessary before they are discovered and composed. The interfaces are the most important during service composition. It's better to standardize interfaces.

Before these services are actually deployed and executed upon the business model, they have to undergo a pilot trial in a real environment. During the pilot trial, some problems will occur. The providers, especially the chief provider, must solve these issues for smoothing future business operation.

3.2 Service Element Representation

A service, which is abstract in the services concept, would be expanded into a service scenario step by step. A service is composed of service elements. The types of service elements include services, roles, proprietors, devices, transactions, and locations. A service here is represented as the following figure.

A role in the services stands for an individual who performs a specific task. The role frequently has professional knowledge and plays an important part in this service. For example, the defined roles in home-care services are doctor, nurse, centre-staff, care-giver, and so on. A role in the services concept is represented by a circle with a person within it, such as the following figure.

A proprietor in the services is responsible for providing general service. This could be combined and replaced into the service through SLA. The collaborative policies are derived from proprietors. For example, the proprietors are the drug deliverer, the hospital, the transporter, and the ambulance, among others. A proprietor is represented by a circle with a service provider such as a car or a building within it. The representation is shown as the following figure.

A device in the services indicates a technical system which could be software, hardware, end-device, system or platform. The technical systems are created through ICT, which are especially focused on software techniques and communication techniques. For example, the defined devices are the homebox, service platform, and bio-signal measurable devices. The homebox, which was developed in our project, is a gateway for collecting and transferring personal bio-signals in the patient's house. Service platform is a software platform for service integration, delivery, and management. An example of the device representation is shown as the below figure.

A transaction is represented as a solid line with a bi-directional arrow. An action(s) on the line is (are) an executable action(s) between a service and a role (or a proprietor). An action would be separated into sequential items in the service scenario. The representation of a transaction is shown as the following figure.

A location is a position where a person or a device is located. Delivering services to a remote side is an important action in modern business models. The services are provided via a communicable network because the service providers and consumers are located in different positions; thus a demarcation of the location is necessary. For example, the defined locations are home, care center, and hospital. An example of location representation is shown as the following figure.

Services are controlled and monitored by a specific role. A dotted line between a service and the role means that the service is uniformly monitored by that role. The progress of a service would be monitored if it is necessary. A service should have features of reliability and efficiency in a real-time system. The services in real life are frequently complex; we would indeed have to manage them using a well-defined service platform, such as service-oriented platform.

4. Case Study

In the home-care services area, patients with chronic diseases need long-term care at home. If they are hospitalized for a long time, a lot of costs are entailed, such as the financial and emotional burden on their families as well as wastage in hospital resources. Therefore, one effective solution is to remotely take care of patients at home. However, that would be a challenge for both the patient's family and the hospital. In this case study, the home-care services are developed by following service life cycle. The service elements introduced in the previous section are also included in the home-care services concept. The services concept in chronic home-care area is shown in Fig. 2.

The home-care services are derived from the services concept. By composing the service elements, service scenarios could be created. Two service scenarios, named as health status monitor and emergency medical treatment, are described in the following sub-sections.

4.1 Health Status Monitor

The patient's health status is regularly monitored by homebox and bio-signal equipment in the house. This service is triggered three times every day. First, homebox sends bio-signals to the SOA-based healthcare platform. When the platform receives the signals, these will be automatically judged by inferable components, which are pre-installed into the platform. If unusual signals are detected, an alert for the patient will be sent to centre-staff's monitor. At the same time, the staff can obtain the patient's conditions through a telephone call, if it's connected. At the same time, the patient's EHR and conditions will be sent to doctor's computer through the service platform for getting the doctor's recommendation. The centre-staff can then quickly take care of the patient and process exact actions for him/her. The service scenario of the health status monitor is shown in Fig. 3.

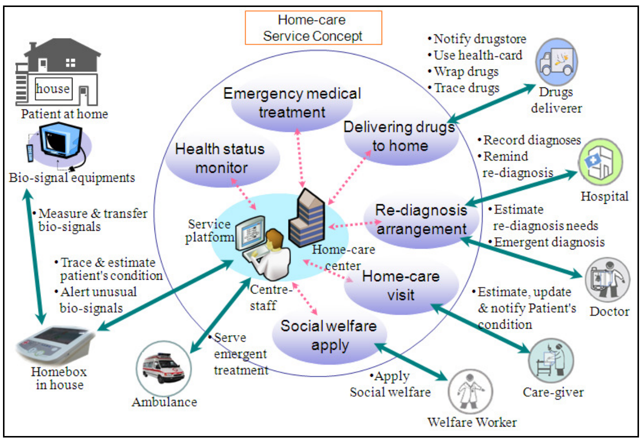

Fig. 2 Home-care Services Concept

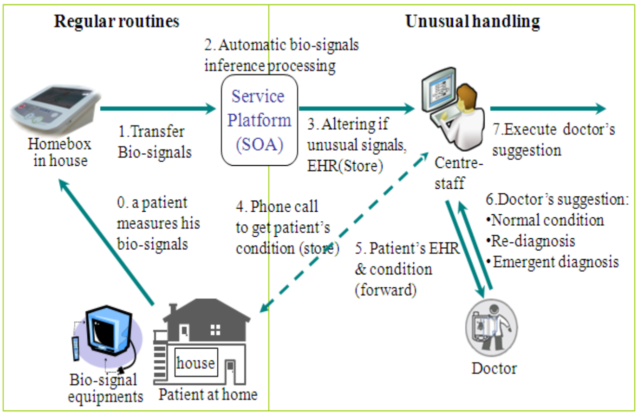

Fig. 3 Health Status Monitor

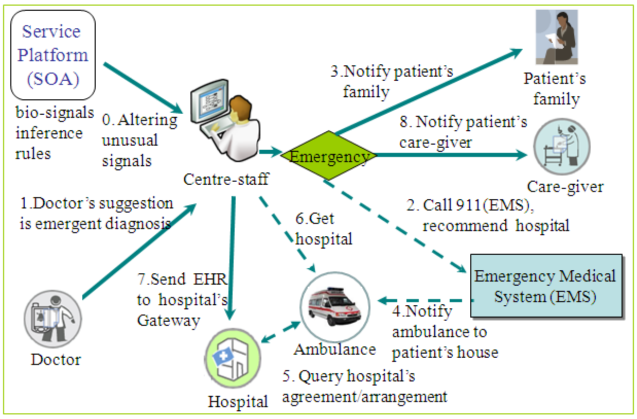

Fig. 4 Emergency Medical Treatment

4.2 Emergency Medical Treatment

The service for patient's emergency care is designed on the service platform. The service will be triggered by the unusual bio-signals, and while the doctor also suggests the patient to diagnose emergently. Thus the centre-staff will immediately call the emergency medical system (EMS). At the same time, the SOA-based healthcare platform is notifying patient's family. While the patient is being taken to the hospital by an ambulance, his/her EHR and the unusual signals are also sent to the hospital (HIS system) through the specific gateway in the service platform. Finally, the patient's emergency will be sent to his/her care-giver PDA. The service scenario of emergency medical treatment is shown in Fig. 4.

The SOA-based healthcare platform is layered out and designed by following SOA principles. Its architecture is in the reference [5]. The service platform can provide executable environments which support standardized messages, various interfaces and flexible connections. Different services techniques and providers could cooperate on this platform. Those messages among providers are passed through a specific message gateway [11]-[12]. The software components are derived from these service elements and are implemented to build up the platform. The components are in described by following Unified Modeling Language (UML) and programmed by C#.NET language.

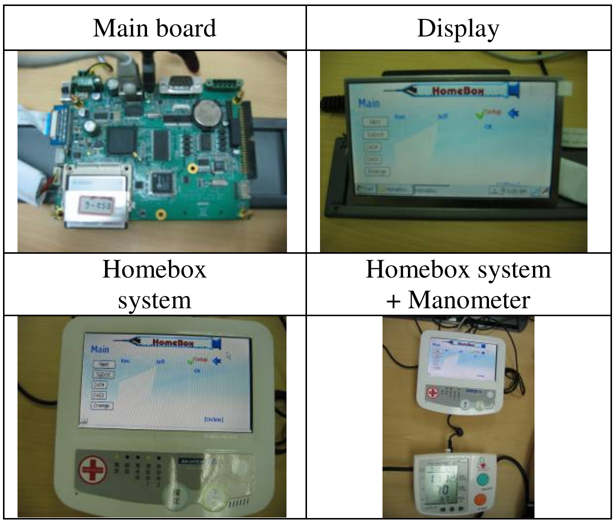

Fig. 5 Homebox system

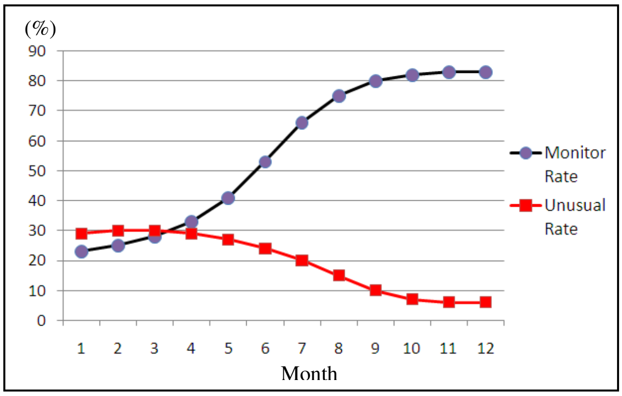

Fig. 6 Rating Services

5. Experiment – A Pilot Trial

5.1 Experiment Environment

The homebox is an embedded system developed by our team in 2006. It is a message gateway used for collecting and transferring patient's bio-signals with his/her conditions to a remote server. At the local site, it could connect bio-signal equipment using RS232 and USB2.0 interfaces. An Ethernet port is also built in the box. Its kernel core is PXA270 with 312MHz CPU and 64MB SDRAM. A 512MB SD card and boot ROM are used to boot the system. Linux kernel version 2.6 is pre-installed as its operation system. A 7" Widescreen LCD with 480x234 Resolution is used for its display. Four user defined hot keys on its surface are used for pressing *Yes*, *No*, *Enter*, and *Display on/off*. The homebox is one of the main devices connected to enable healthcare services at the client site. The homebox is shown in Fig. 5.

With our specific components, the homebox installed in patient's house is reliably on-line for 24 hours a day. The message gateway is even pre-installed to remotely call web services. All components are implemented by C#.NET language. One hundred homeboxes were set up in 100 patient's houses in Kaohsiung City, Taiwan.

The SOA-based healthcare platform is set up in the home-care service center, where the staffs are. A tool for controlling service flows is applied in the service platform. With the tool, centre-staff can be aware of services' progresses. The customized components in the platform are developed by C#.NET language. The users' profiles and bio-signals are permanently stored in MS SQL server 2005. The services scenarios mentioned in Section 4 are experimented in a real environment. One hundred chronic patents were involved in the trial. The pilot trial lasted from September 2007 to August 2008 in Kaohsiung City. The chief medical provider is Chung-Ho Memorial Hospital.

5.2 Experimental Results

The usability of the health status monitor service was rated during the pilot trial. The usage rate was below 30% during the first three months. After eight months, the usage rate increased and reached 80%. This means that more patients got used to the service in the long run. The usage rate was stable up to 80% in the last three months. Patients treat using the service as their daily job. When their bio-signals are forwarded and stored in the back-end server, their health statuses are regularly monitored.

The patients' unusual rate was close to 30% before they joined the pilot trial. However, they cannot be sensibly taken to the hospital when they encounter an unusual condition. The average time of taking them to the hospital is between 10 to 15 hours. However, the medical treatment for them is effective within three hours. Fortunately, our emergent service of medical treatment is enabled when unusual condition is detected. This service could take a patient to the hospital within an average of 35 minutes. Therefore, the variation that a patient will be saved in a critical condition is fairly significant.

The other services were also validated by the pilot trial in this study. The care-givers visited patient's home and took care of them twice a week. They evaluated and recorded the patients' health and life skills. That was why the unusual rate was reduced to below 10%. The rated services are shown in Fig. 6.

6. Conclusion

Up to now, Service-Oriented Architecture (SOA) has become one of the most important techniques to realize those daily services in various areas. Therefore, we should explore systematic methods to realize services from the service-centric viewpoint. In this paper, we carefully explained the interoperability of service creation and realization along its life cycle. Using the methodical sequences, domain-specific services can be well created and designed. The representations of service elements were even used for communicating among providers during service creation. We had validated our methods by the pilot trial in the home-care case study. The experimental results had represented its usability and effectiveness. We believe that a successful experience on service realization is worthy to be shared.

7. Acknowledgements

This research was supported by the 2th Applied Information Services Development and Integration project of the Institute for Information Industry (III) and sponsored by MOEA, Taiwan R.O.C.

Care-experts and pilot patients were supported by the Department of Medical Information, Chung-Ho Memorial Hospital in Kaohsiung, Taiwan R.O.C.

8. References

[1] Thomas Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall PTR, 2004.

[2] Dave Hornford, Definition of SOA, The Open Group, October, 2006.

[3] Choudhary, V., "Software as a Service: implications for investment in software development," in Proceedings of the 40th Annual Hawaii International Conference on System Science (HICSS'07), Los Alamitos, CA, USA, 2007, pp. 209-218.

[4] Yvonne Balzer, "Improve your SOA project plans," IBM Global Services, July 2004.

[5] Chi-Lu Yang, Yeim-Kuan Chang and Chih-Ping Chu, "Modeling Services to Construct Service-Oriented Healthcare Architecture for Digital Home-Care Business," in Proceedings of the 20th International Conference on Software Engineering and Knowledge Engineering (SEKE'08), San Francisco, USA, July, 2008.

[6] Eric Newcomer and Greg Lomow, Understanding SOA with Web Services, Addison Wesley, January, 2005.

[7] Asit Dan, Robert Johnson and Ali Arsanjani, "Information as a service: modeling and realization," in Proceedings of International Workshop on Systems Development in SOA environments (SDSOA'07), IEEE Computer Society, Los Alamitos, CA, USA, May, 2007.

[8] Gennaro Cuomo, "IBM SOA on the Edge," Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, ACM Press, New York, NY, USA, 2005, pp 840-843.

[9] Ned Chapin, "Service Granularity Effects in SOA", in Proceedings of the 20th International Conference on Software Engineering and Knowledge Engineering (SEKE'08), San Francisco, USA, July, 2008.

[10] H. Xu, M. Ayachit and A. Reddyreddy, "Formal Modeling and Analysis of XML Firewall for Service Oriented Systems," International Journal of Security and Networks (IJSN), Vol. 3, No. 3, 2008.

[11] Chi-Lu Yang, Yeim-Kuan Chang and Chih-Ping Chu, "A Gateway Design for Message Passing on SOA Healthcare Platform," in Proceedings of the 4th IEEE International Symposium on Service-oriented System Engineering (SOSE'08), Jhongli, Taiwan, Dec., 2008.

[12] Anthony Nadalin, Chris Kaler, Phillip Hallam-Bakerand and Ronald Monzillo, Web Services Security: SOAP Message Security 1.0, Organization for the Advancement of Structured Information Standards (OASIS), March, 2004.

[13] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau eds. Extensible Markup Language (XML) 1.0 (Fourth Edition), World Wide Web Consortium (W3C) Recommendation, Nov., 2008.

[14] David B., Canyang Kevin L., Roberto C., Jean-Jacques M., Arthur R., Sanjiva W. et al. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. W3C:http://www.w3.org/TR/wsdl20-primer/, 26 June 2007.

[15] Martin G., Marc H., Noah M., Jean-Jacques M., Henrik F., Anish K., Yves L. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), W3C: http://www.w3.org/TR/2007/REC-soap12-part1-20070427/, 27 April 2007.

[16] Tom B., Luc C. and Steve C. et al. UDDI Version 3.0.2. OASIS: http://uddi.org/pubs/uddi-v3.0.2-20041019.pdf, 19 October 2004.