

Design and Implementation of a Web-Service-Based Public-Oriented Personalized Health Care Platform

PengWei Wang, ZhiJun Ding, ChangJun Jiang, and MengChu Zhou, *Fellow, IEEE*

Abstract—The use of information technology and management systems for the betterment of health care is more and more important and popular. However, existing efforts mainly focus on informatization of hospitals or medical institutions within the organizations, and few are directly oriented to the patients, their families, and other ordinary people. The strong demand for various medical and public health care services from customers calls for the creation of powerful individual-oriented personalized health care service systems. Service computing and related technologies can greatly help one in fulfilling this task. In this paper, we present PHISP: a Public-oriented Health care Information Service Platform, which is based on such technologies. It can support numerous health care tasks, provide individuals with many intelligent and personalized services, and support basic remote health care and guardianship. In order to realize the personalized customization and active recommendation of intelligent services for individuals, several key techniques for service composition are integrated, which can support branch and parallel control structures in the process models of composite services and are highlighted in this paper.

Index Terms—Health care service systems, service composition, Web services.

I. INTRODUCTION

IN THE past 50 years, the world's population was growing rapidly. Meanwhile many developed countries are also facing the trend of population aging. These make the shortage of health care and medical resources more and more obvious and severe. The contradiction between per-capita level of medical resources and the growing demand from people for health care

is increasing. Thus, how can we best utilize the limited medical resources to provide more efficient health care services for people and to construct new medical service systems becomes a serious and urgent problem.

The problems faced in the field of medical and health care such as the aging of the population, the increase of chronic patients, the rising medical expenses, and the needs to improve the quality of medical services are common around the world today. In order to solve them, information technology (IT) must be deployed to this field. In recent years, medical information systems play an increasingly important role in supporting doctors and nurses, enhancing the quality of medical services, reducing the medical expenses, and improving the care of chronic patients. Therefore, medical informatization has drawn more and more attention in various countries.

In the early 1960s, the U.S. began the study of hospital information system (HIS). In 1985, according to a survey of data processing of all of the hospitals in the U.S., 80% of them used computers to manage the finance charge, and 25% of them had relatively integrated HIS. By the year 2004, 20% of the hospitals in the U.S. had completed electronic medical record (EMR) system transformation and picture archiving and communication system (PACS) experimentation. The National Committee on Vital and Health Statistics of the U.S. submitted a strategic report “*Information for Health: A Strategy for Building the National Health Information Infrastructure (NHII)*” [37] to the congress in 2001, which introduced the planning blueprint of establishment of the national health information infrastructure from three aspects: personal health, medical institution, and public health. On January 20, 2004, President George W. Bush proposed to establish electronic health records (EHRs) for all of the U.S. citizens within 10 years. Then, in the next year, in order to implement the plan, the National Health Information Network of the U.S. selected four of the world's leading IT companies as total integrators to develop the prototype of national health information network architecture in four pilot regions.

In 1998, the Blair government put forward a goal to make British nationals enjoy the world's best medical services. The important means to achieve it is to realize the sharing of patients' information among general practitioners and hospitals in real time. The National Health Service (NHS) released a development report entitled “*Information for Health: an information strategy for the modern NHS 1998–2005*” [38] in the same year, which proposed the development goal of establishing lifelong EMR for each resident and ensuring that all of the doctors could access these records. In 2005, the Department of Health created a specialized agency named “NHS Connecting for Health,”

Manuscript received October 29, 2011; revised March 25, 2012 and June 19, 2012; accepted June 25, 2012. Date of publication February 1, 2013; date of current version June 12, 2013. This work was supported in part by the National Natural Science Foundation of China under Grants 61173042 and 90818023, by the National Basic Research Program of China (973 Program) under Grant 2010CB328101, by the Shanghai Science & Technology Research Plan of China under Grant 10DJ1400300, by the Program for New Century Excellent Talents in University under Grant NCET-10-0598, and by the “Shu Guang” Project supported by Shanghai Municipal Education Commission and Shanghai Education Development Foundation. A preliminary conference version of this paper appeared in Proceedings of the IEEE International Conference on Web Services (ICWS 2011). This paper was recommended by Associate Editor M. K. Tiwari.

P. Wang, Z. Ding, and C. Jiang are with the Department of Computer Science and Technology, Tongji University, Shanghai 201804, China, and also with the Key Laboratory of Embedded System and Service Computing, Ministry of Education, Shanghai 201804, China (e-mail: pwei.wang@gmail.com; zhijun_ding@hotmail.com; cjjiang@tongji.edu.cn).

M. Zhou is with The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 201804, China and also with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSMCA.2012.2210412

responsible for implementing “the National Programme for IT” derived from 1998. The agency’s goal is to realize EMR, online service for choosing medical institutions and making appointment, electronic prescriptions, health network infrastructure, PACS, and so on. It aims to help patients obtain better, faster, and safer medical care by using IT. To achieve this goal and to build a network infrastructure of health information, the agency signed a 10-year contract, valuing more than 6 billion pounds, with many transnational health informatization corporations in 2005. This was the biggest order of civil IT in the world, and the project would benefit 50 million people in Britain, 28 000 medical institutions, and more than 1 million medical workers. The project divided Britain into five regions and built EMR systems to connect various hospitals and clinics by four contractors, respectively [58], [59]. Now, the British national health information network has made periodical achievement and becomes a representative in national health information construction in Europe.

Canada is also building a national EHR system, and the government plans to provide EHR services for every Canadian. The Canada Health Infoway, founded in 2001, has been exploring the role of information and communication technologies in the construction of Canadian health infrastructure and HIS. In 2002, Infoway announced to invest hundreds of millions of dollars to promote medical institutions and other end users to accept medical ITs, and build the national EHR system, drug information system, laboratory information system, PACS, public health information system, telemedicine system, and unified identification system for users and medical institutions. It planned to establish EHR for 50% of the population in Canada by 2009 and to cover the entire population by 2016 [66].

In Australia, medical informatization in both national and provincial levels has gained great progress. The National E-Health Technology Architecture was founded at the national level to formulate policies, regulations, and standards in the field of medical informatization [67]. Through the establishment of enterprise-level patient-centered clinical information systems in the major hospitals, the South Australia government provides medical professionals with the access to patient medical history information. This changes the preservation, transmission, and access means of information in healthcare systems and even the traditional medical service model. This project covers eight major public hospitals in Adelaide, which provides services for 75% of the statewide 1.5 million population. Other states are doing similar work of regional health informatization.

Similar work of regional health informatization is being carried out in some European countries, Singapore, Japan, and other countries. For instance, the Red System in Denmark manages 76 hospitals and clinics. The Grenoble Integrated HIS of No. 8 health center in France makes integrated management to three major hospitals and three medical colleges. With the development of IT and the execution of the national project named “Jin Wei,” China has also made great progress in the field of medical informatization. According to a recent survey by the Ministry of Health, 84% of provincial-level hospitals, 37% of municipal-level hospitals, and 34% of county-level hospitals in China have established HIS. From the content of hospital informatization, 85% of HIS are financial accounting-

centric management information systems, 10% of the hospitals are trying to establish CIS with information-based doctor’s clinic offices as its core, and 5% of them are exploring PACS.

It is clear that previous work, including each country’s policy support and projects, research, and product development, mainly focused on the following: 1) informatization for hospitals or medical institutions within the organization and regional medical or health information infrastructure, including HIS, radiology information system, EMR, and PACS, and 2) standardization, such as Health Level seven (HL7), Digital Imaging and Communications in Medicine (DICOM), and Integrating the Healthcare Enterprise (IHE). However, there are few health care service systems for the patients, their families, and general users. According to the problems and development requirements in the field of medical and health care, the model of medical and health care services should be gradually transformed to prevention-oriented comprehensive medical model combining hospitals, communities, families, and individuals from the current hospital centralized treatment-based model. The new model needs to provide individuals with personal health information management services and to realize personal health risk assessment and health counseling, dynamic personal health monitoring and real-time early warning, health-knowledge-based customization and active recommendation of health care services, and so on. In other words, it can provide individuals with intelligent health information management and health care services, thereby enabling them to enjoy medical prevention and health care services in their daily lives.

In addition, with the rapid development of ITs, the computing model is shifting to user-centric service computing. A new computing model is bound to change the original business model. Therefore, the appearance of service computing will also change the original business model of medical and health care informatization for the general public. User needs are often diverse, uncertain, and personalized in the field of medical and health care. Moreover, for end users, what they are really concerned of are the health care services that they can enjoy. However, most of the existing medical information systems are based on simple application architectures and not on the modern and smart service-oriented architecture (SOA), and various functions are not packaged and used in the form of “services.” Thus, the existing systems have difficulties in meeting the diverse and personalized needs of users and in adapting to the dynamically changing complex application environments. Consequently, they fail to realize the personalized customization and active recommendation of health care services for users.

Therefore, there is a strong demand for medical and health care service systems for the public under the new computing model. They can provide remote health consultation, remote real-time monitoring, remote diagnosis, personal health record (PHR), network-based health care education, and other personalized services for the public through personalized medical information management and services configuration and integration. Thus, they can help us realize the integration of health care prevention, medical services, and health management.

In this paper, we present a platform called Public-oriented Health care Information Service Platform (PHISP) under the new computing model, which can support numerous health care

tasks and provide individuals with many intelligent and personalized health care services. In particular, for such diseases as cerebral apoplexy, coronary heart disease, and diabetes, some decision-making libraries and remote medical and care systems are established, respectively. To overcome the shortcomings of the existing medical information systems that are based on simple application architectures and to take full advantages of the newly emerging service-oriented technologies, we design the platform based on the modern SOA and adopt Web services and related composition technologies as the means of implementation. We describe its architecture and main components in detail and discuss some important issues about the platform. In the platform, most of the functional modules are packaged in the form of “services,” some of which are described briefly. Then, three types of user/domain preferences are presented, and a simple programming language style expression is proposed to express them. For these preferences that are not considered in the previous work on automatic service composition, we present new solutions to introduce branch structures into the process models of composite services accordingly. Moreover, a new strategy for identifying possible parallel cases in service composition is also presented.

An overview of our platform architecture is reported in Section II. Section III provides basic concepts and definitions. Section IV describes several key techniques integrated in the platform for service composition and an illustrating scenario is followed. The implementation status of the platform is reported in Section V. Finally, Section VI concludes this paper.

II. RELATED WORK

Creating a public-oriented health care system has triggered considerable research interest in both academia and industry in recent years, and the rapid advancement of information and communication technologies brings more opportunities for innovations in the health care field. Some systems are now available to provide various solutions, and there are also some research projects that aim at providing Web-based personalized health care support. Thus, in this section, we provide an overview of some of these systems and projects, and related service composition work.

A. Health Care Systems and Projects

Google Health [62] is a personal health information centralization service by Google introduced in 2008, which can help people organize, track, monitor, and act on their health information. This service allows the users to volunteer their health records into the Google Health system either manually or by logging into their accounts at partnered health service providers, thereby merging potentially separate health records into one centralized Google Health profile. Volunteered information can include health conditions, medications, allergies, and laboratory result. Once entered, Google Health uses the information to provide the user with a merged health record, information on conditions, and possible interactions between drugs, conditions, and allergies [63]. In summary, Google Health is a PHR service and only available to users in the U.S.

Unfortunately, it was discontinued as a service after January 1, 2012.

HealthVault is a Web-based platform developed by Microsoft to put people in control of their health and fitness information, which started in October 2007. It helps the users collect, store, and share health information with family members and participating healthcare providers, and it provides people with a choice of third-party applications and devices to help them manage their fitness, diet, and health [64]. It is also a PHR system. It is offered free of charge initially for consumers in the U.S. and was launched in the U.K. in June 2010.

Dossia [60] is a PHR service offered by some of the largest employers in the U.S., with the intention to offer a PHR to their employees. Along with Microsoft's HealthVault and Google Health, Dossia is one of the largest PHR deployments in the world. Unlike the other two large PHR efforts, it is based on Open Source software. Dossia differs from traditional tethered PHR services, by providing user access to health information regardless of health plan, employer, or physician. Users also have the ability to download their full record, in an electronic form, at any time [61].

World Medical Card [69] is a product and registered trademark belonging to World Medical Center, a Norwegian company. Its main purpose is to allow users to manage and access vital personal data such as diagnoses, medical history, medications, allergies and adverse drug reactions, vaccinations, and important personal documents. The product is composed of three main elements: an online Web-based health profile that allows users to add and edit personal health data (“onWeb”), a WAP-based mobile phone application that gives access to the same data (“onMobile”), and a sealed physical card containing a condensed summary of the holder's health information (“on-Card”) [70].

The SMART project is funded under the Extend Quality Life (EQUAL) initiative of the UK Engineering and Physical Sciences Research Council, and it has been carried out in two stages. The SMART 1 project, entitled “SMART rehabilitation: technological applications for use in the home with stroke patients,” started in November 2003 and completed in December 2006. It aims to explore how technology might be used to facilitate active in-home rehabilitation for people who had stroke [79]. The SMART 2 project started on January 1, 2008 aims to deepen understanding of the potential for technology in the support of self-management. This will be achieved by creating user-centered designs for technology, resulting in a personalized self-management system, and by carrying out extensive studies with the technology in use. Users with three common conditions, namely, chronic pain, stroke, and congestive heart failure, are involved in all stages of the project [47], [68].

There are also some mobile products available that promote liveliness and exercise of people, such as pedometers and activity monitors. These products can monitor a user's activity, usually through counting steps, such that they may see how active they are or have been over a period of time. Some can also calculate distance traveled and calories burned. Generally, they are used to aid weight loss or to increase an individual's activity. A popular, fairly advanced, and commercially available activity monitoring system is the “Nike + iPod.” The system is

configured with an iPod nano, iPod touch, iPhone 3GS, iPhone 4, or iPhone 4s; a pair of Nike+ shoes; and the Nike + iPod Sport Kit or Sensor. The sensor is placed within the specialized Nike+ shoe, tracks the run, and then sends the data to the user's iPod or iPhone. As users run, their iPod or iPhone tells them the time, distance, pace, and calories burned. When the run is finished, the system automatically sends the workout data to the Nike+ Web site from wherever the users are on the road. From there, the user can see the statistics of all the runs that they have done using the system [39].

In these efforts, relatively mature systems like Google Health, HealthVault, Dossia, and World Medical Card are mainly targeted at helping people organize and manage their health information. In other words, they are basically PHR systems and provide few other health care services. Moreover, these systems are limited in terms of their scope of application, e.g., Google Health and HealthVault are only available to users in the U.S., while Dossia offers services only to the employees of some employers in the U.S. The SMART project aims at establishing a personalized self-management system only for users with three conditions, namely, chronic pain, stroke, and congestive heart failure, and it is still in progress. Other products like pedometers and activity monitors often provide a single function such as monitoring the users' activity to promote their liveliness and exercise. By contrast, the platform proposed in this paper is more comprehensive and powerful. It supports personal health information management, personal health risk assessment and guidance, dynamic personal health monitoring and real-time early warning, active recommendation of personalized medical treatment, active seasonal disease warning, and other health care services for individuals. Moreover, for such diseases as cerebral apoplexy, coronary heart disease, and diabetes, the platform can provide remote medical and care services. The adoption of service-oriented technologies makes the system have higher reusability, flexibility, and extensibility. In particular, the adoption of service composition techniques can help provide more value-added services by integrating the basic ones, and thereby better meet the users' diverse and personalized demands. In addition, the proposed platform will gradually open to our partners and other third-party developers, which allows them to register and deploy health-care-related services developed on our cloud platform. With the growing number of services in the platform, the service composition approach will become prominent. In this way, more and more valuable services can be created for users, and this goes beyond what the existing systems can do.

B. Automatic Service Composition

Notations for service description and composition have been proposed in some standardization efforts. WS-BPEL (originally known as BPEL4WS, BPEL for short) [72] is an XML-based language supporting process-oriented service composition [12]. It combines IBM's WSFL and BEA Systems' WSCI with Microsoft's XLANG specification. BPEL is currently being standardized by the Organization for the Advancement of Structured Information Standards. Web service technologies like BPEL fall short on their restricted capability to support

static service composition. Their limitation comes from the total absence of semantic representations of the available services. OWL-S [43] ServiceModel is another initiative. It comes from semantic Web community which focuses on reasoning about Web resources by explicitly declaring their preconditions and effects with terms precisely defined in ontology. It is a solution in response to the foregoing limitation, i.e., lack of semantic information. However, BPEL and OWL-S both focus on representing service compositions where the flow of a process and bindings between services are known in advance. This often leads to a static, manual, and labor-intensive task, and thus, they are not appropriate for large-scale service composition.

Web services are created and updated on the fly, and it is certainly beyond the human ability to analyze them and to generate the composition plan manually [46]. Therefore, building composite services with automated or semiautomated methods is critically important. A large number of methods for this purpose have been proposed. Most of them are based on AI planning techniques.

Berardi *et al.* [5]–[7] present a formal framework where services are characterized in terms of their behavioral descriptions and then provide techniques for computing service composition in the case where the behavioral description is expressed as finite state machines. Their basic idea consists in reducing the problem of composition into satisfiability of a suitable formula of deterministic propositional dynamic logic. SWORD [44] is a developer toolkit for building composite Web services using rule-based plan generation. It uses an entity-relation (ER) model to specify the Web services, and the reasoning is based on the entity and attribute information provided by the ER model. McIlraith *et al.* [30], [31] adopt and extend the Golog language for automatic construction of Web services. They address the Web service composition (WSC) problem through the provision of high-level generic procedures and by customizing users' constraints. Narayanan and McIlraith [34] first use DAML-S ontology to provide semantic markup of the content and capabilities of Web services. Then, they define the semantics for a relevant subset of DAML-S in terms of a first-order logical language. With the semantics in hand, they encode service descriptions in a Petri net formalism and provide decision procedures for service simulation, verification, and composition. Medjahed *et al.* [32] present a technique to generate composite services from high-level declarative description. They define formal safeguards for meaningful composition through the use of composability rules. In [36], [50], and [73], the SHOP2 planner is applied for automatic composition of services, which is a hierarchical task network planner. Sirin *et al.* [49] present a semiautomatic method for WSC. Each time a user has to select a service, all possible services, which match with the selected service, are presented to the user. Oh *et al.* [41], [42] present a WSC algorithm named as WSPR, which is a polynomial-time AI-planning-based heuristic algorithm. Tan *et al.* [54] present a compatibility-based method to analyze and compose services, and colored Petri nets are used to represent and analyze the compatibility between two services. Tan *et al.* [55] propose a Petri-net-based approach to compose services from the perspective of data. The proposed

method represents both data relations and service composition rules with colored Petri nets.

Despite a relatively large body of work in the area of automated service composition, few efforts have specifically focused on introducing more complex control structures in the process models of composite services. Most existing methods for automatic service composition are AI planning based, and the resulting composite services are of sequential structures only. In such models, component services are composed together in a 1-D linear fashion. No matter how diverse the application needs are and how the environment changes, it calls services in accordance with an established order. Hence, they are applicable to only simple cases with a deterministic environment, and cannot meet the diverse needs of users and adapt to the dynamically changing environment. In addition, user preferences are very important for service composition but largely ignored by these existing methods. In particular, in the field of medical and health care, user and domain preferences are very common and have a significant impact on service composition.

From another perspective, Web services are essential software components. Thus, WSC can be conceived as a problem of program (software) synthesis [31]. Program synthesis is a method of software engineering used to generate programs automatically, which was an active research area in the last 40 years. A substantial amount of work on automatic program synthesis has been published, and an overview of major achievements can be found in [3], [4], [13], [14], [16], [18], [27], [28], [51]. Among these approaches, deductive program synthesis methods are based on the observation that proofs are equivalent to programs because each step of a proof can be interpreted as a step of computation, which transforms program synthesis into a theorem-proving task [45]. Based on this, some work has applied deductive program synthesis theory to automatic service composition [22], [29], [45]. In these methods, the descriptions of available services and user requirements need to be translated into logical axioms and logical sequent to be proven, and service composition descriptions need to be extracted from the proofs generated by certain theorem prover. This process is complicated and cumbersome, and such methods are trapped in the problems of efficiency and usability for users. Nevertheless, the feature that the “proof to program” methods guarantee the completeness and correctness of the composition result is still very attractive. By contrast, the completeness and correctness must be proved when adopting other methods, and the proof may be very complex and difficult. In particular, with regard to our concerns about service composition in this paper, there has been much work on automatic program synthesis where different control structures (e.g., branch, iteration, and recursion) are considered. We can learn from these methods to improve our composition techniques in the future work.

In [56], we have considered branch structures appearing in the process model of a composite service and have presented the corresponding solutions for two simple types of user preferences. The problem is resolved from the perspective of service composition techniques purely. In this paper, we present a personalized health care information service platform, and adopt

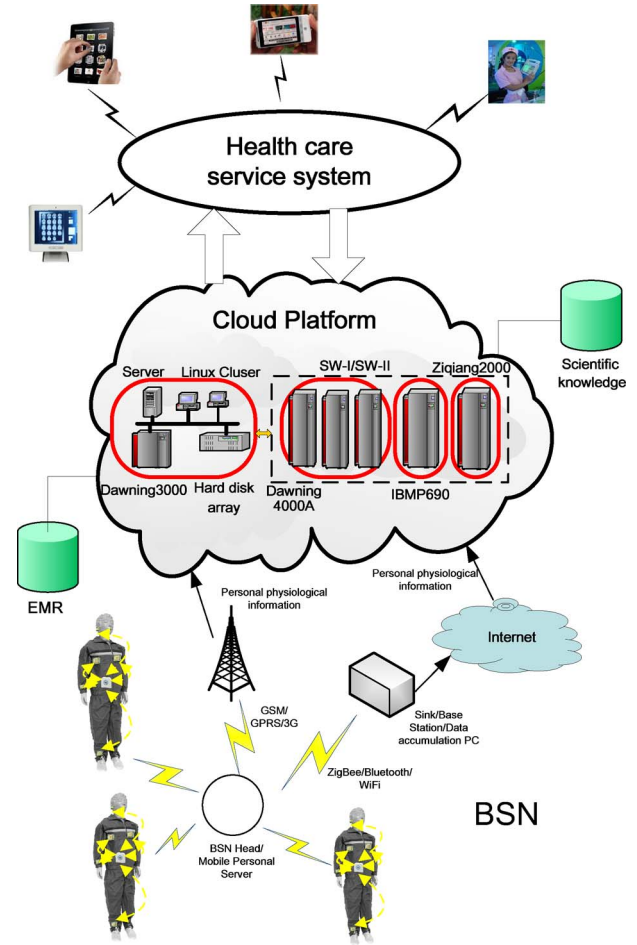


Fig. 1. PHISP architecture.

the idea of SOA and Web service technologies to design and implement the platform. In particular, in order to realize the personalized customization and active recommendation of health care services to individuals, several key techniques for service composition are integrated in the platform, which include WSC techniques supporting branch structures and parallel structures. WSC techniques supporting parallel structures are completely new.

III. SYSTEM ARCHITECTURE

In this section, we present our health care service platform: PHISP. Its conceptual architecture is shown in Fig. 1.

There are three main components, i.e., *body sensor networks (BSNs)*, *cloud platform*, and *health care service system*, described as follows.

- 1) **BSN**: according to different circumstances and needs, appropriate health information collection terminals (i.e., sensors) are configured for different individuals. BSN is used to realize the multimode acquisition, integration, encryption, and real-time transmission of personal health information in living or working environment.
- 2) **Cloud platform**: based on cloud computing technology, it achieves the rapid storage, management, retrieval, and analysis of massive health data, which mainly includes

EMR repository, scientific knowledge base of health care, and personal health data acquired from BSN. Meanwhile, it provides support for the deployment, management, and execution of application-level health-care-related services and systems.

- 3) **Health care service system:** it includes a series of subsystems, such as personal health information management system, personal health risk assessment and guidance system, seasonal disease early warning system, decision-making libraries for various diseases, remote nursing, and medical systems for such patients as cardiovascular, cerebral apoplexy, and diabetic ones.

BSN is a branch of wireless sensor networks (WSNs) [35]. Various sensors form a network and collectively monitor the health condition of their carrier or host. Health monitoring involves the collection of data about vital physiological parameters from different parts of the body and making decisions based on it [11]. BSN is a new kind of solutions for universal health care, disease surveillance, and prevention, and is also an important part of the Internet of things [15], [21]. Its aim is to provide platform that integrates hardware, software, and wireless communication technology and provides necessary conditions for the future development of universal medical and health care systems [17].

Cloud computing [2], [8], [71] has recently emerged as a new paradigm for hosting and delivering services over the Internet. Its emergence made a tremendous impact on the IT industry over the past few years. Large companies such as Google, Amazon, Microsoft, and IBM strive to provide more powerful, reliable, and cost-efficient cloud platforms, and business enterprises seek to reshape their business models to gain benefit from this new paradigm [78]. Unlike these large and powerful public clouds, our cloud platform is more like a private one. In this system architecture, on one hand, it is mainly used for the storage and management of massive health-care-related data. Among these data, some are relatively static, such as scientific knowledge base of medical and health care and EMRs repository. On the other hand, the cloud platform provides support for the upper health care service system, such as system deployment and computing power. The partners can deploy health-care-related services that they developed on this cloud platform, without having to build their own infrastructure, and in the future, certified health care services developed by other providers can also be allowed to deploy and run on this cloud. In addition, a “cloud” storage space will be allocated to each end user of our system, which allows users to upload, save, and download their health care data anytime and anywhere. Therefore, the adoption of the cloud is necessary.

The health care service system in this platform consists of a series of subsystems just as previously described. It relies on the cloud platform and provides services to the outside world over the Internet. Most of the functions in the system are packaged in the form of “services,” which will be described later.

In PHISP, we first deploy some general physiological information sensors, including body temperature sensor, noninvasive blood pressure detector, glucose detector, oxygen saturation sensor, electroencephalogram sensor, electrocardiogram sensor, pulse detector, activity sensor, gyroscope, accelerators, and

some environmental information detection sensors, such as temperature detector and humidity sensor [81]. Furthermore, for three types of specific diseases, namely, cerebral apoplexy, coronary heart disease, and diabetes, we deploy some special sensors in order to collect some special physiological signals, such as cerebral blood flow perfusion information, brain wave, blood vessel elasticity information, retinal blood flow perfusion signal, microcirculation perfusion signal, and peripheral blood flow perfusion signal. These various sensor nodes in BSN can collect important physiological signals, activity, or movement signals, and the environmental information of the human body under living and working environments. Then, they process these signals and transfer them to a local base station nearby the human body. After preliminary processing, these data are transferred to the cloud platform through Internet or mobile communication networks. The storage, management, retrieval, and analysis of the massive health information can be completed on the cloud platform. Based on the real-time personal health information, together with EMRs and scientific knowledge of health care, the health care service system can perform the query and management of personal health information, risk assessment and counseling of personal health, real-time dynamic monitoring and early warning of personal health, personalized customization and active recommendation of health care service, and early warning of seasonal disease. Furthermore, for cerebral apoplexy, coronary heart disease, and diabetes, we complete the related decision-making library and remote medical and care system, respectively, so as to provide intelligent health care services for individuals with such disease.

There is no doubt that the design and implementation of such a platform is very complicated, and many important issues need to be solved. Take the information processing in the platform for example, how to collect, encrypt, and transmit those important physiological signals, activity or movement signals, and the environmental information surrounding the human body? How to represent such information? How to process, integrate, and analyze it? Also, in terms of reliability, how to ensure the reliability of the inputs and outputs of the system? As the system is for public use, it is extremely important that the information provided to the users is correct and reliable. In addition, because of the importance and privacy of personal health information, how to keep it safe and private is also very important. In our platform, we use many methods and techniques to deal with these issues. However, for such a large and complicated health care platform, it is difficult to cover all aspects of the platform in a single paper. Therefore, we only provide a brief introduction to some important aspects in this paper.

Security and privacy: there is no doubt that this is an extremely important issue for such a health care platform. We have adopted many technologies to help resolve this issue. Username and password/finger identification are used basically to authenticate the entrance authority and distinguish at the user level. Before transmission or communication, important data including personal profile, vital signs, and medical images are encrypted to improve data security and privacy. The security components supplied by Java Development Kit (JDK) are adopted in our platform to implement the essential authentication control and to achieve the security, integrity,

and privacy of important data. Specifically, key generation in Java Cryptography Extension and X.509 in Java Cryptography Architecture for digital signature complete the encryption and decryption of important data. Private key and public key pair generated by JDK security libraries can guarantee the security and privacy of critical data and the system; meanwhile, the integrity of the data can also be certificated in the transmission, which ensures the reliability of the system. In addition, some other security and privacy protection technologies have also been used in our platform, which are not detailed here.

Diagnostic techniques: through cooperation with some medical institutions, we have established the corresponding basic rule libraries for such chronic diseases as cerebral apoplexy, coronary heart disease, and diabetes, respectively, and then constructed related diagnostic and decision-making systems. However, we have to acknowledge that they are elementary and relatively simple, and continuous efforts are being made to improve them.

Data integration and communication: BSN is a branch of WSN. Many data fusion and integration technologies in WSN can be readily applicable to BSN. In our platform, we have adopted mature data fusion models and methods [33] to integrate data from different sensors. According to the characteristics of our platform and the equipment used, the communication technologies in BSN mainly use ZigBee and Bluetooth as a supplement [82], [83]. Due to space limit, these two issues, as well as other aspects, are omitted.

As mentioned previously, for such a large and complicated project, many issues are involved, and this paper intends to present our most concerned parts of the platform. It focuses mainly on the introduction of the system architecture and service-oriented technologies used to design and implement the platform. In the rest of this paper, we concentrate our attention on the Web service techniques adopted in the platform.

A. Services Encapsulated in PHISP

In PHISP, we adopt the design idea of SOA and Web service technology for the implementation purpose. In the platform, a majority of functional modules are packaged in the form of services. Here, we describe some of them briefly as follows.

- 1) Physiological information acquisition service (Physi_Info_Acquir). This service can acquire some general physiological signals such as body temperature, blood pressure, saturation of blood oxygen, electrocardiogram, and some special physiological signals according to different sensor deployment for different users. User's ID number is required.
- 2) Environmental information acquisition service (Envir_Info_Acquir). For a unique ID number, this service can acquire temperature, humidity, air pressure, etc., from the user's surrounding environment.
- 3) Subjective feelings acquisition service (Subj_Feel_Acquir). It can acquire the user's subjective feelings, food intake, etc., and the information is often provided by the user from a terminal.
- 4) Coronary heart disease oriented data analysis and diagnosis service (Coronary_Analy_Diagno). Based on physiological information, environmental information, and subjective feelings, this service can analyze such information according to a series of analysis models, which are built for coronary heart disease, and then produce preliminary diagnostic results.
- 5) Cerebral-apoplexy-oriented data analysis and diagnosis service (Cere_Apop_Analy_Diagno) and diabetes-oriented data analysis and diagnosis service (Diabetes_Analy_Diagno), which are similar to Coronary_Analy_Diagno.
- 6) Personal health risk assessment service (Health_Risk_Assess). Based on the preliminary diagnostic results and the EMR information of a patient, this service can assess the status of the patient's health risk. We divide it into three levels: $RL = 0$ represents that the patient is in normal condition; $RL = 1$ represents that the patient has signs of the disease; $RL = 2$ represents that the patient is ill and requires a doctoral appointment.
- 7) Personal health guidance service (Health_Guid). According to the preliminary diagnostic results and EMR information of a patient, it can provide the patient with preventive measures, things to be cared, and other health guidance.
- 8) Personal health real-time warning service (Real_Time_Warn). It can warn the user about the signs of certain disease.
- 9) All-round intervention service (All_Round_Interven). When a user has signs of illness, this service can enforce all-round intervention and instruct the patient to deal with the case.
- 10) Active seasonal disease warning service (Season_Disea_Warn). This service can warn users actively about seasonal disease information and provide them with some preventive measures and other related guidance.
- 11) Emergency alarm service (Emerg_Alarm). It can alarm the user and hospital as soon as the user falls ill.
- 12) EMR service (EMR). With a user's ID number, this service can output the user's medical history information.
- 13) Geographic information service (Geograph_Info). With a user's ID number, this service can acquire the user's location information.
- 14) Activating emergency service (Activ_Emerg). After the emergency alarm, this service can activate some related first aid measures.
- 15) Medical teaching resources video-on-demand service (Video_on_Demand). According to user's needs and requests, this service plays the corresponding health care teaching videos.

In addition, SOA provides a unified foundation for migration from legacy systems to a consistent service-oriented framework [77] and offers a new chance to continue to use and reuse the business functions provided by legacy systems. Nowadays, more and more companies and organizations are considering the opportunity of modernizing their software systems by presenting them as well-defined "services" for exploiting the

advantages offered by SOA. This has triggered considerable research efforts [9], [10], [19], [24], [25], [52], [76]. Thus, because of the adoption of SOA and Web service technologies, different health care systems or other legacy systems can be migrated and integrated into our platform relatively easily. In its implementation process, we also reuse a few legacy systems, particularly the ones provided by our collaborators. These legacy systems always show such characteristics as high dependence on their execution environment, low component modularity, and high degree of coupling among components. Therefore, we mainly adopt the wrapping technology to encapsulate these systems as Web services, which is implemented from three levels, namely, data layer, business layer, and application layer. This method belongs to black-box approaches, which leave the original system configuration untouched, as well as its execution environment [10], thereby ensuring its original security and stability in the integration platform.

IV. PRELIMINARIES

Because we design and implement the health care platform based on SOA and Web service technologies, and several key techniques for service composition are integrated in order to realize the customization and active recommendation of medical and health care services, some basic concepts and definitions related to them are explained first.

A. Basic Concepts

Definition 4.1: Web service. An atomic service can be represented by a tuple $WS = (I, O; QoS)$, where

- I represents a set of semantic concepts that are referenced by the input parameters of the service;
- O represents a set of semantic concepts that are referenced by the output parameters of the service;
- QoS represents a set of quality parameters of the service, such as execution time, price, availability, and reputation.

In SAWSDL [48], the input and output parameters of a Web service can reference to concepts of external semantic models (e.g., service ontology) through extension attribute *modelReference*. A service ontology consists of a common language agreed by a community (e.g., medical service industry). It defines a terminology that is used by all participants in the community. Within a community, service providers describe their services using the terms of the community's ontology, while service requesters use the terms of the ontology to formulate queries over the service registry [75]. At present, there have been many mature medical and healthcare terminologies and classification systems, such as The International Statistical Classification of Diseases and Related Health Problems 10th Revision (ICD-10), Systematized Nomenclature of Medicine—Clinical Terms (SNOMED CT), and Logical Observation Identifiers Names and Codes (LOINC). In our platform, we mainly used SNOMED CT as our terminology standard, i.e., service ontology, which provides reference concepts for input/outputs of services in our platform. SNOMED CT is the most comprehensive and multilingual clinical healthcare terminology in the world, maintained by the International Health Terminology

Standard Development Organization. It is a vital component for the safe and effective communication and reuse of meaningful health information and supports ANSI, DICOM, HL7, and ISO standards [65]. In addition, the partners and other third-party service providers will be gradually allowed to publish and deploy their services in our platform in the near future, although these services may be defined on different ontologies. For the management of such services, ontology mapping methods can be used to address this problem, which depicts the relationships between semantic concepts of different ontologies. Specified to the field of medicine and healthcare, Unified Medical Language System (UMLS) provides a mapping structure among many vocabularies and thus allows one to translate among various terminology systems, and its controlled vocabularies include most of the existing terminologies and classification systems, such as ICD-10, MeSH, SNOMED CT, DSM-IV, LOINC, UK Clinical Terms, and Gene Ontology.

Definition 4.2: User request. A user's request can be represented by a tuple $Req = (I_A, O_E; P, Q)$, where

- I_A represents a set of semantic concepts that are referenced by the available inputs that the service requester provides;
- O_E represents a set of semantic concepts that are referenced by the outputs that the service requester expects;
- P represents a set of personalized preferences defined by the service requester;
- Q represents a set of standards of service's quality parameters defined by the service requester.

B. Composite Service Specifications

A composite service is specified as a combination of other composite or atomic services described in terms of service ontology and according to a set of control and data flow dependencies among them. In order to describe conditional branch structures more vividly, we use activity diagrams in UML to represent such dependency [56].

An activity diagram shows a business process or software process as a flow of work through a series of actions. It includes action, control flow, initial node, activity final node, decision node, guard, and merge node. Moreover, it can express concurrence. It uses a fork node to divide a single flow into concurrent flows and uses a join node to combine concurrent flows into a single one.

In UML activity diagrams, an action is denoted by a round rectangle and is labeled with an action name of a given service class as defined in a service ontology. The flow of control is denoted by connectors (transitions) between actions. A decision node represents a conditional branch in a flow, and it has one input and two or more outputs. A merge node is required to merge flows that are split with a decision node, and it has two or more inputs and one output. Both decision and merge nodes are denoted by diamonds. The initial node of an activity diagram is denoted by a filled circle, while the final one is denoted by two concentric circles.

A simplified UML activity diagram specifying a "treating a stroke patient" composite service is shown in Fig. 2. In the example, various information acquisition services and EMR

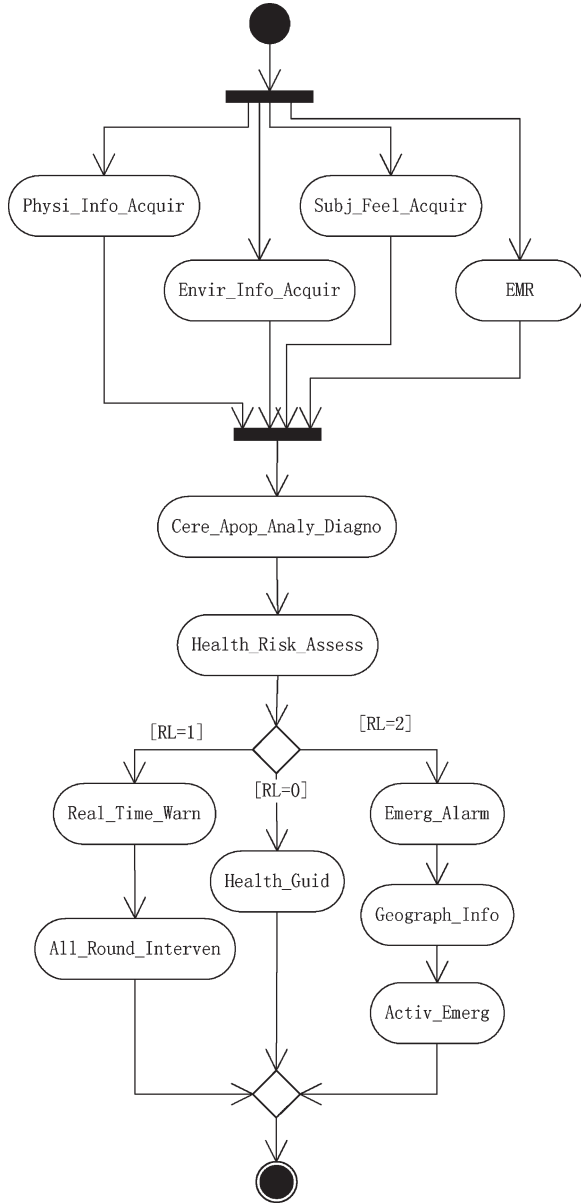


Fig. 2. Activity diagram of composite service: treating a stroke patient.

service can be executed in parallel first in order to acquire physiological signals, environmental information, subjective feeling, and medical history information of the patient. Then, the information can be analyzed by the cerebral-apoplexy-oriented data analysis and diagnosis service, and the personal health risk assessment service can assess the patient's risk level. According to different risk levels, there will be different solutions. The decision node denotes a branch structure. Three transitions (connectors) stem from it, and they are labeled with disjoint guards—conditions that specify whether a token can flow along a connector.

C. Formalizing User/Domain Preferences

User preferences are a key component of WSC. However, this component has been largely ignored in the existing approaches for service composition. It is critical for the following reasons.

First, according to users' requirements, a family of solutions can be induced by a composition method. User preferences enable a user to specify properties of solutions that make them more or less desirable. The composition method can use them to generate preferred solutions [53].

Second, user preferences can affect how the composition is performed. A key component of WSC is the selection of specific services used to realize the composition. In AI planning, primitive actions are selected for composition based on their preconditions and effects. Like actions, services are selected for composition based on functional properties such as inputs, outputs, preconditions, and effects, but they should also be selected based on domain-specific nonfunctional properties such as, in the case of making a doctor appointment, pick a doctor that a patient likes. By integrating user preferences into WSC, preferences over services can be specified and considered alongside preferences over the solutions [53].

Last but not the least, service composition is user-oriented, and it should be centered on the users' needs. User preferences and constraints are also parts of their needs. Therefore, we should not neglect them so as to improve users' satisfaction.

For a specific field, there are often some common requirements or preferences, named as domain preferences. For example, in health care, according to different states of illness through diagnosis and risk assessment, different treatment plans or health care services will be performed. The corresponding relationship between different health care services and the results of diagnosis service is fixed in this field, and this is domain preference for service composition. Domain preferences can also be considered as a kind of user preferences and have a significant impact on service composition.

In this paper, we pay attention to only user/domain preferences in which we are interested in the field of health care and give them a simple expression. There are three types of user/domain preferences with which we are concerned. The first type is that a user prefers a class of services over another if certain conditions are met (e.g., "Lucy prefers to go to a doctor's office by walk over bus, if the walk time is less than 30 minutes and weather is pleasant."). The second type is that the user prefers different services according to different conditions. The third type is that the user assigns priorities over services with similar functionalities. We use the following forms to express them, respectively:

- 1) the first type: $condition?WS_1 : WS_2$;
- 2) the second type: $switch(condition)\{case\ C_1 : WS_1; case\ C_2 : WS_2; \dots; case\ C_n : WS_n; \}$;
- 3) the third type: $WS_1 \gg WS_2 \gg \dots \gg WS_n$.

The expressions that we adopt are similar to some general expressions in programming languages, especially the first two, which are the conditional expressions commonly used in C, C++, and Java. For the first one, it finds the truth value of expression "*condition*" first. If it is TRUE, then execute service WS_1 , and otherwise WS_2 . For the second one, it also finds the truth value of expression "*condition*" first and then execute the corresponding service accordingly. The third one means that WS_1 has higher priority than WS_2 and WS_2 has higher priority than WS_3 , etc. In other words, WS_2 will be executed

only if WS_1 fails, and WS_3 will be executed only if WS_1 and WS_2 both fail, etc.

The proposed expressions are simple and easy to understand and use. They can help users express their preferences and developers build applications quickly.

D. Formalizing the WSC Problem

Definition 4.3: WSC. Suppose that there is a Web service community $C = \{WS_1, WS_2, \dots, WS_n\}$, where $WS_i = (I_i, O_i; QoS_i)$, $1 \leq i \leq n$, and a user request $Req = (I_A, O_E; P, Q)$. A WSC problem is to find a finite set of services from the available service community C such that: 1) these selected services can be organized in a certain mode of construction to form a new value-added executable service WSC ; 2) WSC can accept inputs I_A that are provided by a user and produce outputs O_E that the user expects; 3) the quality of selected services and the value-added service should reach the standards defined by the user (i.e., Q); and 4) the process model of a final composite service should reflect user/domain preferences P .

V. KEY TECHNIQUES INTEGRATED IN THE PLATFORM

In PHISP, we integrate several key technologies and methodologies for service composition so as to provide customizable and personalized health care services for individuals. Next, we give a detailed introduction to two important techniques.

Under SOA, service composition is used to integrate service components together to meet new needs. It is considered to be a key factor in stimulating the Web evolving from an information-delivering center to a function offering platform to better support the automated use of various services, and it is of great importance in the emerging paradigm of SOA.

A large number of methods have been proposed for service composition. Among them, dynamic composition methods can generate a process model automatically, and most of these methods adopt AI planning techniques. According to them, the process model of a composite service is of a sequential structure. In such models, component services are composed together in a 1-D linear fashion. No matter how diverse the application needs are and how the environment changes, it calls services in accordance with the an established order. Hence, they are applicable to only simple cases with a deterministic environment. In order to meet the demands of more complex applications and changing environments, more control constructs need to be proposed and applied for service composition. How to achieve automatic service composition supporting them is a new challenge.

A. WSC Techniques Supporting Branch Structures

Service composition in the field of health care faces some problems and difficulties. For example, the needs of users are often uncertain, diverse, and personalized, and there are also some domain preferences. Composite services with only sequential process models cannot meet the diverse needs of users and reflect those user or domain preferences. Thus, most of the existing methods fail to address this issue. For example,

classical AI planning techniques can derive only linear sequences of actions. They require that the outcome of each action is known in advance. However, many services cannot meet such requirement since their outcomes can be determined only after their execution (e.g., diagnosis service or risk assessment service), and their different outcomes can lead to different processes of the composite service.

In our platform, we introduce branch structures into the process model of composite service when needed in the process of service composition. This makes the composite service reflect user and domain preferences and meet the diverse and personalized needs of users, thereby overcoming the deficiencies of the existing methods that rely on sequential structures.

In our previous work [56], we have considered this problem and proposed solutions for two simple types of common user preferences. One is of the form “*condition?* $WS_1 : WS_2$,” which is exactly the first type mentioned in the previous section. The other one is of the form “ $WS_1 \gg WS_2$,” which is a special case of the third type. In the following, we propose a solution for the second type of preferences.

Algorithm **Service Composition Planner (SCP)** is the main algorithm, and algorithm **Processing Preference with Multiple Cases (PPMC)** is its subprocedure. They generate all of the feasible composite services; each of which is specified by an activity diagram.

Algorithm SCP(C, Req): Service Composition Planner
Input: $C = (WS_1, WS_2, \dots, WS_n)$, service community, a set of available services;
 $Req = (I_A, O_E; P, Q)$, the user's request.
Output: $ADG = (adg_1, adg_2, \dots, adg_m)$, a set of activity diagrams representing all the feasible composite services.

- 1: Establish the initial node, and denote it by N_0 ;
- 2: The corresponding state of N_0 is S_0 ;
- 3: $NS = \{N_0\}$; $RS = \emptyset$;
- 4: **while** $NS \neq \emptyset$ **do**
- 5: $\forall N' \in NS$, suppose that its corresponding state is S' ;
- 6: **if** $S' \supseteq O_E$ **then**
- 7: $RS = RS \cup \{N'\}$; $NS = NS - \{N'\}$;
- 8: **else**
- 9: According to S' , do one step reasoning to obtain the executable services, noted as $\Omega(S')$;
- 10: $NS_{temp} = \emptyset$;
- 11: **for each** service $w \in \Omega(S')$ **do**
- 12: Introduce an action node into the graph, labeled with the name of service w , and draw a connector from N' to it;
- 13: Denote the new action node by N_i , and compute its corresponding state S_i ;
- 14: $NS = NS \cup \{N_i\}$, $NS_{temp} = NS_{temp} \cup \{N_i\}$;
- 15: **end for**
- 16: **if** There exists the second type of user preferences **then**
- 17: Call subprocedure PPMC;
- 18: **end if**
- 19: $NS = NS - \{N'\}$;
- 20: **end if**

```

21: end while
22: if  $RS = \emptyset$  then
23:   print There is no solution!
24:   Exit.
25: else
26:   Introduce an activity final node into the diagram,
     denoted as  $N_f$ ;
27:   for each action node  $N_j \in RS$  do
28:     Draw a connector from  $N_j$  to  $N_f$ ;
29:   end for
30: end if
31: Remove all the action nodes that cannot be connected to
     the final node  $N_f$  from the diagram, and also their
     associated connectors;
32: Obtain each path from the initial node to activity final
     node in the generated diagram, which is a feasible solu-
     tion to the service composition problem. Note that
     each branch structure should be included in one com-
     posite service—it cannot be divided.

```

In the algorithm, each action node N in the generated activity diagram represents an available service from community C . Moreover, an information state S is associated with N to denote the current status after executing the service that the node represents. It contains the currently available input and output parameters and such information as what conditions are presently met. This is the basis for the next step reasoning. In service composition based on input and output parameters only, state S can be just a collection of parameters obtained by executing a sequence of services—starting from the initial node and ending at the current node along a path of the diagram. The associated state of the initial node N_0 is $S_0 = I_A$, which is the set of available inputs that the service requester provides. At S , we may have a set of services $\Omega(S)$, each of which can be invoked.

After every step of reasoning, we check the set of newly obtained executable services to see whether any concerned service appears or not. If yes, we have to introduce a branch structure into the activity diagram under certain conditions.

At the end of the algorithm, the set RS records all of the action nodes whose corresponding states can satisfy the user's requirement. According to these nodes, we can find out all of the feasible solutions from the constructed diagram. $RS = \emptyset$ indicates that there are no solutions to this WSC problem.

It is worth noting that we can use any existing reasoning method to compute state S and the set of invocable services $\Omega(S)$ under the state in our algorithm. Many of such methods can do this job, with some simply based on inputs and outputs of services [20], [23], [40], [41] and some based on inputs, outputs, preconditions, and effects [46], [49]. At the end of the algorithm, we may obtain several solutions to a specific WSC problem. For each feasible one, according to the QoS parameters of each component service, we can compute the value of QoS parameters of the whole composite service. Then, we can select the best solution that can satisfy user requirements for QoS parameters and have optimal performance. The formulas to compute QoS parameters can be found in [1], [26], [74], and [75], and they will not be discussed here.

Algorithm PPMC is a subprocedure used to introduce multiple branch structures into the activity diagram when some conditions are satisfied.

In the reasoning process, once the services mentioned in the user's preference (i.e., w_1, w_2, \dots, w_n in PPMC) appear, we will examine whether the truth value of condition formula P can be acquired and will prepare to introduce a branch structure into the diagram. We need to check three cases as follows.

Algorithm PPMC: Processing Preference with Multiple Cases

```

1: if  $A = \{w_1, w_2, \dots, w_n\} \subseteq \Omega(S')$  then
2:   Suppose that  $N_1^p, N_2^p, \dots, N_n^p \in NS_{temp}$  representing
     the action nodes corresponding to services  $w_1, w_2, \dots, w_n$ 
     respectively;
3:    $flag = \text{false}$ ;
4:   According to state  $S'$ , examine whether the truth value
     of formula  $P$  in user preferences can be found;
5:   if found then
6:     Introduce a decision node into the diagram, and draw
     a connector from  $N'$  to it;
7:      $flag = \text{true}$ ;
8:   else if There is some node  $N_l \in NS_{temp} - \{N_1^p, N_2^p, \dots, N_n^p\}$ ,
     such that the truth value of  $P$  can be found
     under its corresponding state  $S_l$  then
9:     Introduce a decision node into the diagram, and draw
     a connector from  $N_l$  to it;
10:     $flag = \text{true}$ ;
11:   else
12:     Suppose  $NS' = NS_{temp} - \{N_1^p, N_2^p, \dots, N_n^p\}$ ;
13:     while  $NS' \neq \emptyset$  do
14:        $\forall N_t \in NS'$  with state  $S_t$ ;
15:       if  $S_t \supseteq O_E$  then
16:          $RS = RS \cup \{N_t\}$ ;  $NS' = NS' - \{N_t\}$ ;  $NS = NS - \{N_t\}$ ;
17:       else
18:         According to  $S_t$ , do one step reasoning to obtain
         all the executable services, denoted as  $\Omega(S_t)$ ;
19:          $NS'_t = \emptyset$ ;
20:         for each service  $w \in \Omega(S_t)$  do
21:           Introduce an action node  $N'_i$  into the diagram,
           labeled with the name of service  $w$ , draw a
           connector from  $N_t$  to it, and compute  $S'_i$  of  $N'_i$ ;
            $NS'_t = NS'_t \cup \{N'_i\}$ ;
22:         end for
23:          $NS' = NS' \cup NS'_t - \{N_t\}$ ;  $NS = NS \cup NS'_t - \{N_t\}$ ;
24:         if There is some node  $N'_l \in NS'_t$ , such that the
         truth value of  $P$  can be found under its corre-
         sponding state  $S'_l$  then
25:           Introduce a decision node into the diagram,
           and draw a connector from  $N'_l$  to it;
26:            $flag = \text{true}$ ;
27:           Exit the while loop;
28:         end if
29:       end if
30:     end while
31:   end while

```



```

32: end if
33: if  $flag = \text{true}$  then
34:   for  $i = 1$  to  $n$  do
35:     Draw a connector from the decision node to  $N_i^p$ , set
     a guard along this new connector with the condition
     " $C_i$ ," then delete the original connector pointed to it;
36:   end for
37:   Introduce a merge node  $N_M$  into the diagram, draw a
   connector from action node  $N_1^p, N_2^p, \dots, N_n^p$  to it
   respectively; and compute the corresponding state
    $S_M$  of  $N_M$ ;
38:    $NS = NS - \{N_1^p, N_2^p, \dots, N_n^p\} \cup N_M$ ;
39: else
40:   The conditional branch is failed. Delete the related
   nodes;
41:   Exit.
42: end if
43: end if

```

First, P can be evaluated under the current state S' . In this case, we can introduce a decision node into the diagram directly, whose input comes from the action N' (which is being processed in SCP).

Second, P cannot be evaluated under S' . However, in the set of services $\Omega(S')$ that can be executed currently, there is a service such that the state obtained after executing it can result in the truth value of P . Under this circumstance, after the action corresponding to the service, we introduce a decision node.

Third, if both aforementioned cases are not satisfied, we continue the reasoning process based on the action set NS_{temp} which is corresponding to the invocable service set $\Omega(S')$, until there exists an action whose state can result in the truth value of P . Here, the reasoning process is similar to the one in SCP.

After the decision node is introduced, no matter which case is satisfied, we draw connectors from it to N_1^p, N_2^p, \dots , and N_n^p (corresponding to the services designated), respectively. Meanwhile, set the corresponding guard (C_1, C_2, \dots , and C_n) for these control flows. Then, we introduce a merge node into the diagram, whose inputs originate from N_1^p, N_2^p, \dots , and N_n^p . After that, we compute the corresponding state S_M of merge node N_M and add N_M into NS to complete the connection with SCP. The merge node is also with a state for reasoning here for the simplicity of our algorithm.

If none of these three cases is met, the multiple branching structure cannot be introduced into the diagram at present.

We present three types of preferences in the previous section, which are common in the field of health care. The first one has corresponding solution in our previous work [56]. For the third one, we have presented solution for its special case in [56], and only minor modification to the previous algorithm is needed for this more complex case. Thus, these two types of preferences will not be discussed here.

Remarks: The complexity of the algorithm depends on the reasoning mechanism used in the process of service composition. It is polynomial in the size of action nodes in the activity diagram constructed during the execution of the algorithm, but the number of action nodes may be exponential in the

number of service parameters in the worst case. In our current platform, the number of available services and their parameters are relatively limited, and thus, the search scope is not large for a specific request even if we use a simple nonoptimized reasoning mechanism for services. Thus, taking into account the characteristics of our platform, the proposed algorithm is tractable and admissible. Moreover, through the testing and verification of the prototype system already implemented, the efficiency of solution finding is acceptable. In addition, the algorithm can generate all of the feasible composition solutions. If only one solution needs to be generated once a time, a polynomial-time algorithm can be easily obtained by minor modification to the algorithm, since the number of available services in the platform is finite (e.g., $|C|$), thereby the length of resulting composite services is limited by $|C|$. There have been some polynomial-time methods for automatic service composition [41], [80], which can be employed here, because our method does not specify a concrete reasoning mechanism for services in the process of composition and may adopt any efficient one in existence.

As a matter of fact, we pay more attention to an idea of taking the key factors such as user/domain preferences into service composition. The introduction of branch structures is based on user/domain preferences in the service requests. In the reasoning process of composition, once the concerned services appear, a branch structure reflecting the user preference will be introduced after the truth value of condition formula P is acquired. In addition, the introduction of branch structures does not increase the overall complexity of the approach.

Regarding the correctness of the algorithm, we analyze it from the following aspects. First, during the execution of the algorithm, the set of executable services obtained from every step of reasoning will be checked. When the services contained in user preference appear, we will try to introduce a branch structure reflecting the preference into the activity diagram being constructed. The process of introduction is related to the condition in the user preference, i.e., whether its truth value can be acquired. Then, three cases are examined successively in the algorithm: the current state of reasoning has accumulated enough information such that the truth value of preference condition P can be obtained; otherwise, for a sibling node of the current node being processed, its state has enough information which can result in the truth value of condition P . If both cases are not satisfied, the reasoning process will be continued until some newly added action node can result in the truth value of condition P , or the algorithm ends, and such nodes are not found. Since constructing an activity diagram is actually a traversing process of the state space, the three cases considered in the algorithm are complete and cover all of the possible situations. In other words, for a specific service request, if there exist composite solutions that contain branch structures to reflect the user preference based on the available services community, they can be constructed by this method. Second, the algorithm can terminate, whether there are feasible composite solutions or not. This is because there are only a finite number of available services ($|C|$) in the services community. Thus, the number of executable services at every step of reasoning and the length of each path in the generated activity diagram are all limited

by $|C|$. Therefore, the set NS in the while loop of the main algorithm will eventually be emptied, and then, the algorithm terminates.

B. WSC Techniques Supporting Parallel Structures

In the process of service composition, some component services participating in a composite service can be executed in parallel, and thus we should seek the maximal parallelism in their execution. This can improve the efficiency of composite service execution. To the best knowledge of the authors, no prior work has done so. In the prior art, these component services that can be executed in parallel are arranged in sequence in a feasible solution. As an undesired side effect, this leads to a number of solutions that include the same collection of component services, and their only difference is the execution order of their component services. Since they are actually the same, such methods greatly increase the solution cost and complexity and contribute to low efficiency of the composite service. Take the “treating a stroke patient” composite service in Fig. 2 as an example. By existing methods [23], [41], [42], [46], [56], $4! = 24$ feasible solutions will be derived. They are identical in practice, and their only difference is the execution order of services *Physi_Info_Acquir*, *Envir_Info_Acquir*, *Subj_Feel_Acquir*, and *EMR*. Moreover, for any one of these solutions, the execution time of these four services is the sum of them, while it rests with the maximum one among four services through parallel processing, as shown in Fig. 2.

Generally speaking, identifying all of the possible parallel cases of component services is very difficult in service composition. However, when we consider this problem inside an enterprise, the situation becomes relatively easier. In our platform, we present a few simple rules and adopt some techniques to support parallel structures in the process of composition.

We introduce the concept of **layer** first. In the process of service composition, starting from the initial state, after every step of reasoning we can obtain a set of newly executable services. We state that these services are at the same layer, and those information states that are associated with services at the same layer are also called states at the same layer. Elements of each information state set are in the form of propositions, and such layer containing information states is called a proposition layer. Then, we define a binary mutual exclusion (BME) relation between services. Two services at a same layer in a composition process are mutually exclusive if no valid solutions could possibly contain both of them. Identifying such relationships can help construct parallel structures.

We notice and record BME relationships by two simple rules. Specifically, there are two ways in which services WS_1 and WS_2 at a given layer are marked to be mutually exclusive:

- 1) **BME 1**: if either of the services disables a precondition of the other; and
- 2) **BME 2**: if their preconditions are marked to be mutually exclusive in the previous proposition layer.

Two propositions p and q in a proposition layer are marked to be mutually exclusive if any of the services that generates proposition p is mutually exclusive with any that generates q .

Preconditions of a service mentioned in the two rules specify the conditions that must be true in order for an agent to execute the service. Generally, they include knowledge preconditions and physical preconditions, which are in the form of propositions. In our platform, we compose services mainly based on input and output parameters, and thus, such preconditions are mainly knowledge preconditions. Knowledge precondition represents that the agent *knows* the input parameters of a service. In other words, we use the form $know(x)$ to express that service parameter x is available, which is for the sake of convenience and scalability. With these two rules, we can identify parallel component services to a certain extent. An identification rule is given as follows.

- 1) **P-Rule**. The services that are not marked to be mutually exclusive with each other in the same service layer can be considered to be executed in parallel.

With this parallelism identification rule in hand, we can easily extend our previous service composition algorithms to support parallel structures. After every step of reasoning, a set of newly executable services can be obtained, and they are called in the same layer. In this layer, find out those services that are not exclusive with each other, and introduce a parallel structure for them. Practically, introduce a *fork node* into the diagram first, and draw a corresponding connector to it; then, draw connectors from it to the nodes that represent those parallelizable services, respectively, and delete the original connectors pointed to them. Finally, introduce a *join node* into the graph, and draw connectors from these nodes to it, respectively. Moreover, unite the corresponding state sets of these parallelizable action nodes, and make this new proposition set as the information state of the *join node*.

Note that, by this method, we cannot guarantee to find out all of the parallel situations in the process of service composition. What we find is similar to that of “one-step” parallel cases. This can be shown more intuitively and vividly by an illustrating scenario, which is taken from [56], and we only give a fragment of it here for conciseness. This is an online shopping scenario, and we just choose three of the available services, namely, *SearchProduct*, *OrderProcessing*, and *AccountQuery*. Only input and output parameters of them are given for simplicity. The input sets of these three services are $\{ProductName\}$, $\{ProductCode, ProductQuantity\}$, and $\{AccountNumber, AccountPassword\}$, while their output sets are $\{ProductCode, ProductPrice, ProductDescription\}$, $\{OrderInformation, Payment\}$, and $\{AccountBalance\}$, respectively. The available inputs provided by a service requestor are $\{ProductName, ProductQuantity, AccountNumber, AccountPassword\}$.

In Fig. 3, the activity diagram fragments of service composition for this example are illustrated. We show three cases in the figure: the first one is that using previous methods that do not consider parallelism [Fig. 3(a)] [20], [23], [40], [41], [46], [56], the second one is that using the proposed method in this work [Fig. 3(b)], and the third one is the real parallel situation for this example [Fig. 3(c)].

Remarks: Here, we adopt a simple strategy to identify the possible parallel cases of participating components in the process of service composition. Through two simple rules, we can

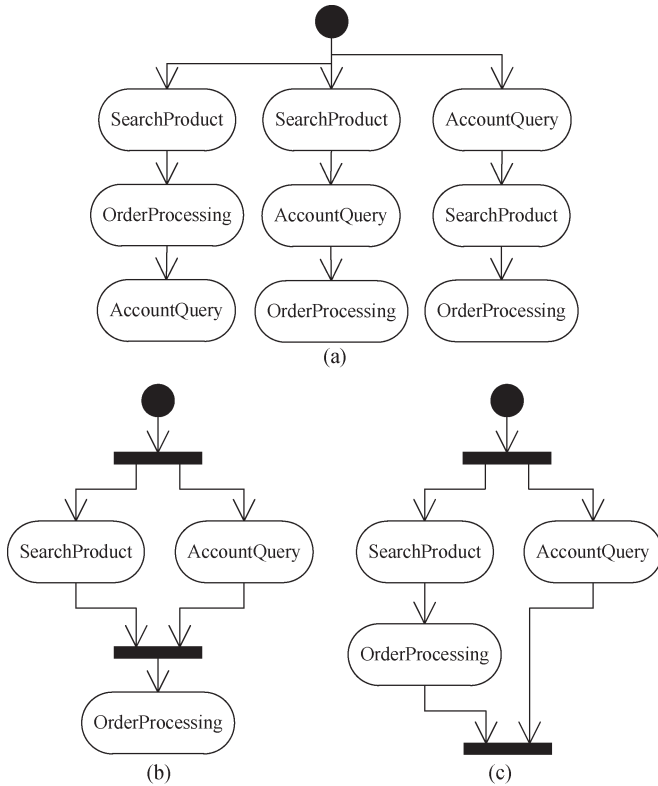


Fig. 3. Activity diagram fragments of service composition for an online shopping scenario.

find out the mutually exclusive relations between component services at the same layer and mark them in the composition process, which reveal what services cannot be involved in the same solution. In other words, the services that are not mutually exclusive in the same layer can be executed in parallel. Obviously, this strategy is simple and general, which can be utilized easily by existing methods for service composition to support parallel structures.

With respect to the correctness of the strategy, services in the same layer refer to those that can be executed at the same information state, and the BME relationships recorded in the reasoning process of service composition reflect what services cannot coexist in the same composite solution. Based on this, the services without BME relationships in the same layer can be executed simultaneously because they can be invoked at the same information state and do not conflict with each other. Therefore, the parallel situations identified in this way are correct. However, not all of the parallel situations can be found with this strategy. In addition, these rules can be integrated into other service composition algorithms to assist them in identifying the possible parallel situations, and they do not affect the correctness of those algorithms and their solving process.

Because both the size of each service layer and the length of resulting composite services are limited by the scale of services community $|C|$, the time to identify and record mutually exclusive relations is polynomial in $|C|$.

C. Illustrating Scenario

In PHISP, we can provide personalized customization and active recommendation of health care services for the general

public, which can be achieved through service composition according to the user's health status information and demands. Moreover, for three types of specific diseases, i.e., cerebral apoplexy, coronary heart disease, and diabetes, we establish their related decision-making library and remote medical and care system, respectively, so as to provide intelligent health care services for individuals.

As an illustrating example, we consider a male stroke patient as a user of PHISP. Suppose that he makes a request for his physical health status monitoring and the corresponding health care services. The activity diagram of a solution that PHISP provides is shown in Fig. 2.

In this solution, four services can be invoked first. They are physiological information acquisition service (Physi_Info_Acqui), environmental information acquisition service (Envir_Info_Acqui), subjective feeling acquisition service (Subj_Feel_Acqui), and EMR service (EMR). They are not marked to be mutually exclusive with each other, and thus can be executed in parallel. Physi_Info_Acqui service can acquire some general physiological signals such as body temperature, blood pressure, saturation of blood oxygen, electrocardiogram, and some special physiological signals for cerebral apoplexy, such as cerebral blood flow perfusion information, brain wave, and blood vessel elasticity information. Envir_Info_Acqui service can acquire temperature, humidity, air pressure, and other environmental information. Subj_Feel_Acqui service can acquire the user's subjective feelings, food intake, etc., and this information is often provided by the patient from the terminal. EMR service can read the patient's medical history information. Based on such information, the cerebral-apoplexy-oriented data analysis and diagnosis service (Cere_Apop_Analy_Diagno) can analyze the data according to a variety of models, like an analysis model of blood vessel elasticity. Then, combining the analytic results with EMR information of the patient, the personal health risk assessment service (Health_Risk_Assess) assesses the patient's risk level (RL). There are three cases of the results: $RL = 0$ or in normal condition, $RL = 1$ or having signs of the disease, and $RL = 2$ or ill. According to different risk levels, there are different processes as determined by domain preferences and practical needs. When $RL = 0$, the personal health guidance service (Health_Guid) is executed next to provide some guidance. When $RL = 1$, warn the patient first by the personal health real-time warning service (Real_Time_Warn), and then, enforce all-round intervention, and instruct the patient to deal with the case by means of the all-round intervention service (All_Round_Interven). When $RL = 2$, alarm his hospital and doctor directly by the emergency alarm service (Emerg_Alarm), acquire the patient's location information by the geographic information service (Geograph_Info), and then activate emergency service (Activ_Emerg).

VI. IMPLEMENTATION

Based on SOA, Web services, and WSC technologies, we have designed and constructed a medical and health care platform in collaboration with many universities and medical



Fig. 4. Starting page of the prototype system.

institutions, such as Peking University People's Hospital, Beijing, China; Zhejiang University, Hangzhou, China; Huazhong University of Science and Technology, Wuhan, China; Wuhan University Medicine School, Wuhan; and Shantou University Medical College, Shantou, China. It has been used to provide teaching, research, medical treatment, and health care services for college students and teachers, researchers, medical professionals, and ordinary people.

The platform is constructed by using our national CNGI-CERNET2 backbone network, which is the test platform for the next generation of Internet technology (i.e., IPv6) in China. By cooperating with the central server, our medical platform can fully exploit the resources and memory space in each network node under the circumstance of IPv6 wired and wireless network. The CNGI-CERNET2-based platform breaks the function restriction of a traditional medical platform. By exploiting the advantage of IPv6, the system can guarantee the reliability and high speed of the data communication over wired and wireless networks, which is one of the effective means to solve the problems caused by the current network technology of medical industries.

The platform can satisfy the needs of diverse user groups and can provide them with various resources and services. It can meet the diverse medical and health care needs of ordinary people. For cerebral apoplexy, coronary heart disease, and diabetes, we have set up several analysis models and diagnostic and decision-making systems, which can provide remote medical and care services. A large number of universities and institutes have joined the CNGI-CERNET2 network in China. The platform can thus provide convenient information communication and comprehensive and up-to-date medical information for the users of these universities and institutes with IPv6 access network. In addition, it can meet the need of the online cooperative diagnosis and first aid of the hospitals located in different places. Fig. 4 simply shows the starting page of our health care platform.

VII. CONCLUSION AND FUTURE WORK

In the field of medical treatment and health care, existing work, including projects, researches, software products, and standards, mainly focuses on informatization for hospitals or

medical institutions within the organizations. However, few efforts are directly oriented to the patients, their families, and other general users. With the rapid development of information and communication technologies and the change of medical and health care service models, creating various public-oriented health care service systems has become a trend. In this paper, we have presented a health care platform PHISP. It supports personal health information management, personal health risk assessment and guidance, dynamic personal health monitoring and real-time early warning, active recommendation of personalized medical treatment, active seasonal disease warning, and other health care services for individuals. Moreover, for some specific diseases, PHISP provides remote medical and care services.

With the help of information and communication technologies, especially the service-oriented technologies, we have designed and implemented the health care platform based on SOA and Web service technologies, which makes our developed system have higher reusability, flexibility, and extensibility. In particular, in order to realize the personalized customization and active recommendation of health care services for individuals and to provide them with intelligent health information management, several key techniques for service composition are integrated in the platform. Two important techniques are presented, which can introduce branch and parallel structures into the process models of composite services. Their use can significantly help meet the users' diverse and personalized needs in changing application environments.

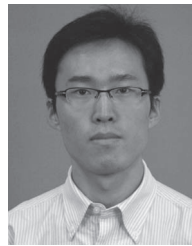
In our implemented platform, some functions and services require improvements. For example, for specific diseases mentioned in this paper, we have just adopted some relatively simple data analysis models and diagnostic and decision-making systems. Furthermore, the medical and health care platform that we have implemented is only a research and demonstration system. For its large-scale commercial applications, much more work is needed. Further improvement and optimization of the proposed composition methods supporting branch and parallel structures is another future research, including learning from the work on automatic program synthesis and the technique of multiagent planning and optimization [84]–[88].

REFERENCES

- [1] D. Ardagna and B. Pernici, "Adaptive service composition in flexible processes," *IEEE Trans. Softw. Eng.*, vol. 33, no. 6, pp. 369–384, Jun. 2007.
- [2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A Berkeley view of cloud computing," EECS Department, Univ. California, Berkeley, CA, Tech. Rep. UCB/EECS-2009-28, Feb. 2009.
- [3] R. Balzer, "A 15 year perspective on automatic programming," *IEEE Trans. Softw. Eng.*, vol. SE-11, no. 11, pp. 1257–1268, Nov. 1985.
- [4] D. Basin, Y. Deville, P. Flener, A. Hamfelt, and J. F. Nilsson, "Synthesis of programs in computational logic," in *Program Development in Computational Logic*, vol. 3049, LNCS, M. Bruynooghe and K.-K. Lau, Eds. New York: Springer-Verlag, 2004, pp. 30–65.
- [5] D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella, "Automatic composition of e-services that export their behavior," in *Proc. ICSOC*, 2003, pp. 43–58.
- [6] D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella, "Synthesis of underspecified composite e-services based on automated reasoning," in *Proc. ICSOC*, 2004, pp. 105–114.
- [7] D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella, "Automatic service composition based on behavioral descriptions," *Int. J. Coop. Inf. Syst.*, vol. 14, no. 4, pp. 333–376, 2005.

- [8] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, Jun. 2009.
- [9] G. Canfora, A. R. Fasolino, G. Frattolillo, and P. Tramontana, "Migrating interactive legacy system to Web services," in *Proc. CSMR*, 2006, pp. 24–36.
- [10] G. Canfora, A. R. Fasolino, G. Frattolillo, and P. Tramontana, "A wrapping approach for migrating legacy system interactive functionalities to service oriented architectures," *J. Syst. Softw.*, vol. 81, no. 4, pp. 463–480, Apr. 2008.
- [11] S. Cherukuri, K. K. Venkatasubramanian, and S. K. S. Gupta, "Biosec: A biometric based approach for securing communication in wireless networks of biosensors implanted in the human body," in *Proc. ICPPW*, Oct. 2003, pp. 432–439.
- [12] F. Curbera, R. Khalaf, N. Mukhi, S. Tai, and S. Weerawarana, "The next step in Web services," *Commun. ACM*, vol. 46, no. 10, pp. 29–34, Oct. 2003.
- [13] Y. Deville and K. K. Lau, "Logic program synthesis," *J. Logic Program.*, vol. 19/20, no. S1, pp. 321–350, 1994.
- [14] P. Flener, "Achievements and prospects of program synthesis," in *Computational Logic: Logic Programming and Beyond*, vol. 2407, LNAI, A. C. Kakas and F. Sadri, Eds. New York: Springer-Verlag, 2002, pp. 310–346.
- [15] N. Gershenfeld, R. Krikorian, and D. Cohen, "The Internet of things," *Sci. Amer.*, vol. 291, no. 4, pp. 76–81, Sep. 2004.
- [16] A. T. Goldberg, "Knowledge-based programming: A survey of program design and construction techniques," *IEEE Trans. Softw. Eng.*, vol. 12, no. 7, pp. 752–768, Jul. 1986.
- [17] J. Gong, R. Wang, and L. Cui, "Research advances and challenges of body sensor network," (in Chinese), *J. Comput. Res. Develop.*, vol. 47, no. 5, pp. 737–753, 2010.
- [18] S. Gulwani, "Dimensions in program synthesis," in *Proc. PPDP*, 2010, pp. 13–24.
- [19] H. Guo, C. Guo, F. Chen, and H. Yang, "Wrapping client-server application to Web services for Internet computing," in *Proc. PDCAT*, 2005, pp. 366–370.
- [20] S. V. Hashemian and F. Mavaddat, "A graph-based approach to Web services composition," in *Proc. SAINT*, Feb. 2005, pp. 183–189.
- [21] International Telecommunication Union, ITU Internet Reports 2005: The Internet of Things, Nov. 2005. [Online]. Available: <http://www.itu.int/osg/spu/publications/Internetofthings/>
- [22] S. Lämmermann, "Runtime Service Composition via Logic-Based Program Synthesis," Ph.D. dissertation, Department Microelectron. Inf. Technol., Royal Institute of Technology, Stockholm, Sweden, Jun. 2002.
- [23] Q. A. Lang and S. Y. W. Su, "AND/OR graph and search algorithm for discovering composite Web services," *Int. J. Web. Serv. Res.*, vol. 2, no. 4, pp. 48–67, Oct. 2005.
- [24] G. Lewis, E. Morris, and D. Smith, "Analyzing the reuse potential of migrating legacy components to a service-oriented architecture," in *Proc. CSMR*, 2006, pp. 14–22.
- [25] M. Litoiu, "Migrating to Web services: A performance engineering approach," *J. Softw. Maint. Evol.-Res. Pract.*, vol. 16, no. 1/2, pp. 51–70, Jan.–Apr. 2004.
- [26] Y. Liu, A. H. Ngu, and L. Z. Zeng, "QoS computation and policing in dynamic Web service selection," in *Proc. WWW*, May 2004, pp. 66–73.
- [27] Z. Manna and R. J. Waldinger, "Toward automatic program synthesis," *Commun. ACM*, vol. 14, no. 3, pp. 151–165, Mar. 1971.
- [28] Z. Manna and R. J. Waldinger, "Fundamentals of deductive program synthesis," *IEEE Trans. Softw. Eng.*, vol. 18, no. 8, pp. 674–704, Aug. 1992.
- [29] M. Matskin and J. Rao, "Value-added Web services composition using automatic program synthesis," in *Proc. CAiSE 2002 Int. Workshop WES*, May 2002, pp. 213–224.
- [30] S. McIlraith, T. C. Son, and H. Zeng, "Semantic Web services," *IEEE Intell. Syst.*, vol. 16, no. 2, pp. 46–53, Mar./Apr. 2001.
- [31] S. McIlraith and T. C. Son, "Adapting Golog for composition of semantic Web services," in *Proc. KR*, Apr. 2002, pp. 482–493.
- [32] B. Medjahed, A. Bouguettaya, and A. K. Elmagarmid, "Composing Web services on the semantic Web," *Vldb J.*, vol. 12, no. 4, pp. 333–351, Nov. 2003.
- [33] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: Methods, models, and classifications," *ACM Comput. Surv.*, vol. 39, no. 3, p. 9, Aug. 2007.
- [34] S. Narayanan and S. McIlraith, "Simulation, verification and automated composition of Web service," in *Proc. WWW*, 2002, pp. 77–88.
- [35] A. Natarajan, M. Motani, B. de Silva, K.-K. Yap, and K. C. Chua, "Investigating network architectures for body sensor networks," in *Proc. ACM SIGMOBILE Int. Workshop HealthNet*, Jun. 2007, pp. 19–24.
- [36] D. Nau, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman, "SHOP2: An HTN planning system," *J. Artif. Intell. Res.*, vol. 20, pp. 379–404, 2003.
- [37] "Information for health: A strategy for building the national health information infrastructure (NHII), Report and recommendations from the National Committee on Vital and Health Statistics (NCVHS)," U.S. Dept. Health Human Services, Washington, DC, Nov. 15, 2001.
- [38] Dept. Health, NHS Executive, Information for Health: An Information Strategy for the Modern NHS 1998–2005, London, U.K., 1998..
- [39] Apple, Nike + iPod. [Online]. Available: <http://www.apple.com/ipod/nike/>
- [40] S. C. Oh, B. W. On, E. J. Larson, and D. Lee, "BF*: Web services discovery and composition as graph search problem," in *Proc. IEEE*, Mar. 2005, pp. 784–786.
- [41] S. C. Oh, D. Lee, and S. R. T. Kumara, "Web service planner (WSPR): An effective and scalable Web service composition algorithm," *Int. J. Web Serv. Res.*, vol. 4, no. 1, pp. 1–23, 2007.
- [42] S. C. Oh, D. Lee, and S. R. T. Kumara, "Effective Web service composition in diverse and large-scale service networks," *IEEE Trans. Serv. Comput.*, vol. 1, no. 1, pp. 15–32, Jan.–Mar. 2008.
- [43] W3C Member Submission, OWL-S: Semantic Markup for Web Services, Nov. 22, 2004. [Online]. Available: <http://www.w3.org/Submission/OWL-S/>
- [44] S. R. Ponnekanti and A. Fox, "SWORD: A developer toolkit for Web service composition," in *Proc. WWW*, Honolulu, HI, May 2002, pp. 83–107.
- [45] J. Rao, "Semantic Web service composition via logic-based program synthesis," Ph.D. dissertation, Dept. Comput. Inf. Sci., Norwegian Univ. Sci. Technol., Trondheim, Norway, 2004.
- [46] J. Rao, P. Küngas, and M. Matskin, "Composition of semantic Web services using linear logic theorem proving," *Inf. Syst.*, vol. 31, no. 4/5, pp. 340–360, Jun. 2006.
- [47] B. A. Rosser, E. Keogh, C. Eccleston, and G. A. Mountain, "SMART2: Development towards a technology-based system for self-management of chronic pain," in *Proc. IASP 13th World Congr. Pain*, Montreal, QC, Canada, Aug. 2010.
- [48] W3C Recommendation, Semantic Annotations for WSDL and XML Schema (SAWSDL), Aug. 28, 2007. [Online]. Available: <http://www.w3.org/TR/sawSDL/>
- [49] E. Sirin, J. Hendler, and B. Parsia, "Semi-automatic composition of Web services using semantic descriptions," in *Proc. Web Services: Model., Arch. Infrastruct. Workshop ICEIS*, Angers, France, Apr. 2003, pp. 17–24.
- [50] E. Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau, "HTN planning for Web Service composition using SHOP2," *J. Web Semantics*, vol. 1, no. 4, pp. 377–396, Oct. 2004.
- [51] C. H. Smith, A general theory of automatic program synthesis, Dept. Comput. Sci., Purdue Univ., West Lafayette, IN, Tech. Rep. 80-360. [Online]. Available: <http://docs.lib.purdue.edu/cstech/290>
- [52] H. Sneed, "Integrating legacy software into a service oriented architecture," in *Proc. CSMR*, 2006, pp. 3–14.
- [53] S. Sohrabi, N. Prokoshyna, and S. McIlraith, "Web service composition via the customization of Golog programs with user preferences," in *Conceptual Modeling: Foundations and Applications*, A. T. Borgida et al., Ed. New York: Springer-Verlag, 2009, pp. 319–334.
- [54] W. Tan, Y. S. Fan, and M. C. Zhou, "A Petri net-based method for compatibility analysis and composition of Web services in business process execution language," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 1, pp. 94–106, Jan. 2009.
- [55] W. Tan, Y. S. Fan, M. C. Zhou, and Z. Tian, "Data-driven service composition in building SOA solutions: A Petri net approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 686–694, Jul. 2010.
- [56] P. W. Wang, Z. J. Ding, C. J. Jiang, and M. C. Zhou, "Automated Web service composition supporting conditional branch structures," *Enterp. Inf. Syst.*, to be published. [Online]. Available: <http://dx.doi.org/10.1080/17517575.2011.584132>
- [57] P. W. Wang, Z. J. Ding, C. J. Jiang, and M. C. Zhou, "Web service composition techniques in a health care service platform," in *Proc. ICWS*, Jul. 2011, pp. 355–362.
- [58] The National Health Service (NHS) Website. [Online]. Available: <http://www.nhs.uk/>

- [59] The NHS Connecting for Health Website. [Online]. Available: <http://www.connectingforhealth.nhs.uk/>
- [60] The Dossia Website. [Online]. Available: <http://www.dossia.org/>
- [61] Wikipedia, Dossia. [Online]. Available: <http://en.wikipedia.org/wiki/Dossia>
- [62] The Google Health Website. [Online]. Available: <http://www.google.com/health/>
- [63] Wikipedia, Google Health. [Online]. Available: http://en.wikipedia.org/wiki/Google_Health
- [64] The Microsoft HealthVault Website. [Online]. Available: <http://www.healthvault.com/>
- [65] The IHTSDO Website. [Online]. Available: <http://www.ihtsdo.org/>
- [66] The Canada Health Infoway Website. [Online]. Available: <https://www.infoway-inforoute.ca/>
- [67] The National E-Health Transition Authority (NEHTA) Website. [Online]. Available: <http://www.nehta.gov.au/>
- [68] The Smart Consortium Website. [Online]. Available: <http://www.thsmartconsortium.org/>
- [69] The World Medical Card Website. [Online]. Available: <http://www.wmc-card.com/>
- [70] Wikipedia, World Medical Card. [Online]. Available: http://en.wikipedia.org/wiki/World_Medical_Card
- [71] Y. Wei and M. B. Blake, "Service-oriented computing and cloud computing: Challenges and opportunities," *IEEE Internet Comput.*, vol. 14, no. 6, pp. 72–75, Nov./Dec. 2010.
- [72] OASIS Standard, Web Services Business Process Execution Language (WS-BPEL) Version 2.0, Apr. 11, 2007. [Online]. Available: <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>
- [73] D. Wu, B. Parsia, E. Sirin, J. Hendler, and D. Nau, "Automating DAML-S Web services composition using SHOP2," in *Proc. ISWC*, Oct. 2003, pp. 195–210.
- [74] P. C. Xiong, Y. S. Fan, and M. C. Zhou, "QoS-aware Web service configuration," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 4, pp. 888–895, Jul. 2008.
- [75] L. Zeng, B. Benattallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for Web service composition," *IEEE Trans. Softw. Eng.*, vol. 30, no. 5, pp. 311–327, May 2004.
- [76] Z. Zhang and H. Yang, "Incubating services in legacy systems for architectural migration," in *Proc. APSEC*, Dec. 2004, pp. 196–203.
- [77] L. J. Zhang, J. Zhang, and H. Cai, *Services Computing*. Beijing, China: Tsinghua Univ. Press, 2007.
- [78] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Serv. Appl.*, vol. 1, no. 1, pp. 7–18, May 2010.
- [79] H. Zheng, R. J. Davies, H. Zhou, J. Hammerton, S. J. Mawson, P. M. Ware, N. D. Black, C. Eccleston, H. Hu, T. Stone, G. A. Mountain, and N. D. Harris, "SMART project: Application of emerging information and communication technology to home-based rehabilitation for stroke patients," *Int. J. Disabil. Hum. Dev.*, vol. 5, no. 3, pp. 271–276, Jul. 2006.
- [80] X. Zheng and Y. Yan, "An efficient syntactic Web service composition algorithm based on the planning graph model," in *Proc. ICWS*, Sep. 2008, pp. 691–699.
- [81] W. Shi and M. C. Zhou, "Pacemakers and their related sensors: A survey," *IEEE Sensors J.*, vol. 12, no. 6, pp. 1817–1827, Jun. 2012.
- [82] J. S. Choi and M. C. Zhou, "Recent advances in wireless sensor networks for health monitoring," *Int. J. Intell. Control Syst.*, vol. 14, no. 4, pp. 49–58, Dec. 2010.
- [83] J. S. Choi and M. C. Zhou, "Design issues in Zigbee-based sensor network for healthcare applications," in *Proc. IEEE Int. Conf. Netw., Sens. Control*, Beijing, China, Apr. 11–14, 2012, pp. 238–243.
- [84] C. Chu, F. Chu, M. Zhou, H. Chen, and Q. Shen, "A polynomial dynamic programming algorithm for crude oil transportation planning," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 1, pp. 42–55, Jan. 2012.
- [85] Y. Fang, S. Chu, S. Mammara, and M. C. Zhou, "Optimal lane reservation in transportation network," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 482–491, Jun. 2012.
- [86] Z. Guo, M. C. Zhou, and G. Jiang, "Adaptive optimal sensor placement and boundary estimation for monitoring mass objects," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 1, pp. 222–232, Jun. 2008.
- [87] N. Wu, F. Chu, C. Chu, and M. C. Zhou, "Hybrid Petri net modeling and schedulability analysis of high fusion point oil transportation under tank grouping strategy for crude oil operations in refinery," *IEEE Trans. Syst., Man, Cybern. C, Appl.*, vol. 40, no. 2, pp. 159–175, Mar. 2010.
- [88] H. Zhu, M. C. Zhou, and R. Alkins, "Group role assignment via a Kuhn-Munkres algorithm-based solution," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 3, pp. 739–750, May. 2012.



PengWei Wang received the B.S. and M.S. degrees in computer science from Shandong University of Science and Technology, Qingdao, China, in 2005 and 2008, respectively. He is currently working toward the Ph.D. degree in the Department of Computer Science and Technology, Tongji University, Shanghai, China.

His research interests include services computing, Web services, and Petri nets.



ZhiJun Ding received the M.S. degree from Shandong University of Science and Technology, Taian, China, in 2001 and the Ph.D. degree in computer science from Tongji University, Shanghai, China, in 2007.

He is currently an Associate Professor with the Department of Computer Science and Technology, Tongji University. He has published more than 50 papers in domestic and international academic publications. His research interests are in service computing, semantic Web, Petri nets, and workflows.



ChangJun Jiang received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1995.

He finished his postdoctoral work at the Institute of Computing, Chinese Academy of Sciences, in 1997 and continued his research at the City University of Hong Kong, Kowloon, Hong Kong, as a Visiting Professor in 1998. He is currently a Professor with the Department of Computer Science and Technology, Tongji University. He is currently the Leading Scientist of the National Basic Research

Program of China (973 program) on the project "Model and theory in Internet information service." He has published more than 200 papers on journals and conferences such as *Sci. China*, *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, *IEEE TRANSACTIONS ON COMPUTERS*, and *IEEE TRANSACTIONS ON FUZZY SYSTEMS*. His current research interests include concurrence theory, Petri nets, formal verification, services computing, massive information service and wireless networks.

Dr. Jiang is a council member of China Automation Federation and Artificial Intelligence Federation, the Director of the Professional Committee of Petri Net of China Computer Federation, the Vice Director of the Professional Committee of Management Systems of China Automation Federation, and an Information Area Specialist of Shanghai Municipal Government.



MengChu Zhou (S'88–M'90–SM'93–F'03) received the B.S. degree in electrical engineering from Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1990.

He joined New Jersey Institute of Technology, Newark, in 1990, where he is a Professor of electrical and computer engineering. He is currently a

Professor with Tongji University, Shanghai, China. His research interests are in Petri nets, sensor networks, semiconductor manufacturing, transportation, and energy systems. He has over 460 publications including ten books, 200+ journal papers (majority in *IEEE Transactions*), and 18 book chapters. He is the Founding Editor of the *IEEE Press Book Series on Systems Science and Engineering*, and Editor of the *IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING*. He is a Fellow of the American Association for the Advancement of Science (AAAS).