

Research Article

Service-Oriented Radio Architecture: A Novel M2M Network Architecture for Cognitive Radio Systems

Xu Dong,¹ Shengqun Wei,² Ying Li,² Lifeng Wang,² and Lin Bai³

¹*Institute of Communications Engineering, PLA University of Science and Technology, Nanjing 210007, China*

²*Institute of China Electronic System Engineering Corporation, Beijing 100141, China*

³*School of Electronic and Information Engineering, Beihang University, Beijing 100191, China*

Correspondence should be addressed to Lin Bai, l.bai@buaa.edu.cn

Received 20 April 2012; Revised 4 July 2012; Accepted 4 July 2012

Academic Editor: Jianhua He

Copyright © 2012 Xu Dong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In future cognitive radio networks, a number of spectrum sensors can be distributedly deployed to monitor the surrounding wireless environment, where the machine-to-machine (M2M) technology is considered to provide the interactions among sensors, cognitive engines, and other system modules. Thus, a flexible M2M network architecture is desired to develop cognitive radio networks. As a distributed system framework, service-oriented architecture (SOA) has been well studied to provide the loose coupling, reusability, and scalability, where various system function modules are encapsulated into web services to build a virtual system. In this paper, based on SOA, we propose a flexible M2M architecture, namely, the service-oriented radio architecture (SORA), to develop the cognitive radio systems. It shows that our proposed architecture provides coordinated implementation of cognitive radio systems based on different development platforms. Furthermore, an SORA-based cognitive radio testbed is implemented, where the standard web service technologies and open source software tools are used to support the characteristics of SORA.

1. Introduction

In cognitive radio (CR) systems, the sensors are employed to monitor the surrounding spectrum environment, where the cognitive engines (CE) and software defined radio (SDR) techniques are needed to generate and carry out different communication strategies according to the actual environment. As an efficient method to develop CR systems, the machine-to-machine (M2M) networks among sensors, CEs, databases, and SDR systems are drawing wide attentions. However, large-scale CR systems have posed a big challenge to develop the M2M networks with respect to the flexibility and compatibility. For example, since different CR systems, testbeds, and prototypes could be implemented on different software and hardware development platforms, the heterogeneity of those platforms would lead to high incompatibility for cooperative development [1]. Then, the interfaces of a certain machine may not be reused by others due to the unique design of data interfaces. Moreover, considering a tightly-coupled architecture of SDR systems,

for example, software communication architecture (SCA) [2], the interaction and integration among SDR system and other machines become even more difficult. Therefore, to deal with the above issues in future CR networks design, a flexible and compatible M2M architecture needs to be considered.

Recently, building a large-scale and multinode testbed over wide areas becomes an urgent need for the research on future CR networks and has already been concerned by the European cooperation in science and technology (COST) IC0902 project [3]. The main objective of IC0902 project is to develop cooperative research activities in the field of CR networks. For the comprehensive cooperative research, different mechanisms, for instance, a larger-scale CR network testbed integrated from different CR testbeds or prototypes, are established, where the challenges to develop the M2M networks mentioned in above need to be carefully considered. To this end, different complex inter-operations and interfaces are implemented among various platforms, especially when components of the system are developed

with different programming languages and hardware platforms.

Since heterogeneous system integration and cooperative development have posed a big challenge over the past decade, as a solution, the service-oriented architecture (SOA) [4, 5] has been used to integrate and develop different interenterprise systems. An SOA-based system consists of the loose-coupled and platform-independent services with well-defined interoperable interfaces. Through web-based publish/subscribe mechanisms, the services can be flexibly orchestrated to develop appropriate applications. Thus, the SOA allows cooperative design and development, where reusability and scalability are considered.

In this respect, SOA has already been used to design the M2M platforms [6]. In addition, Virginia Tech has built a cognitive radio open source system (CROSS) to integrate cognitive engines based on different intelligent algorithms [7], where CROSS utilizes socket connections and SOA approaches for inter-component communications to support portability and interoperability between components developed in different programming languages.

Using the principles of SOA, in this paper, we propose an approach to collaboratively design and develop the CR system, namely, the service-oriented radio architecture (SORA). To be different from the traditional-distributed radio architectures (e.g., SCA), SORA provides a distributed and modular radio communication system framework, which assembles a radio system over networks or even Internet (i.e., based on web service technologies). Moreover, as building a radio system consists of some blocks, the services in SORA are loosecoupled. Comparing to traditional radio system architectures, it shows that our proposed SORA is able to support the underlying heterogeneity, remote system invocation and integration, and meanwhile provide higher scalability, reusability, and flexibility.

The rest of this paper is organized as follows. In Section 2, we introduce the SORA in details. The work on developing an SORA-based CR testbed is shown in Section 3. Finally, the paper is concluded in Section 4.

2. Service-Oriented Radio Architecture

The principles and methodologies of SOA have been widely employed in different wireless communication systems with various applications [8–11]. Specifically, in CROSS, the SOA has been used to design and construct the CR systems. Comparing to the CROSS, a number of web service standards are considered in SORA-based systems to provide higher flexibility and compatibility. The CROSS developers utilize the SOA approaches to integrate their cognitive engines developed in different languages with an usual SDR platform, but the program and protocols of SOA are designed and developed by themselves, where the open standards of SOA are not employed. Thus, when a new system component is added into CROSS, its dedicated SOA framework needs a corresponding change. In contrast, the SORA is based on a series of international standards. The new component can be encapsulated according to the standards so that it

is compatible with the original system without any change. Additionally, in SORA-based system, the components with different functions can be flexibly chosen and integrated to complete the corresponding communication task. Then, the characteristics of SORA will be detailed as follows.

2.1. An Overview. Our proposed SORA is an open architecture, where the principles and methodologies of service orientation are considered to design and integrate radio communication systems. Using SORA, different radio function modules are encapsulated into the network-based components (viz., services). Since radio services are published and discovered over networks, users are able to invoke and orchestrate different required services to implement various radio communication applications.

Two key issues are considered to define the SORA. Firstly, SORA is a distributed and modular system architecture, where the radio function modules can be deployed as services over networks. Secondly, SORA provides an open architecture, where the radio services adopt open and standardized interface descriptions and interoperation contracts. Different characteristics of SORA-based radio systems are summarized as follow.

2.1.1. Loose-Coupling. As the building blocks of radio systems, radio services are abstracted from various radio functions of actual employed hardware platforms, software programs, and operating systems. Thus, different radio services are able to interoperate in a neutral format.

2.1.2. Reusability. Services with different or same radio functions can be used in multiple separate radio communication systems to reduce the cost of development.

2.1.3. Scalability. Radio systems are expediently upgraded as either the software programs of the original services are updated or new services are integrated. Then, the legacy systems can be well compatible with new systems.

2.1.4. Reconfigurability. Reconfiguration is implemented to re-organize the newly involved services or re-define operational parameters of the original services according to the surrounding radio frequency (RF) environment.

2.1.5. Cooperative Development. Academic and commercial communities are able to independently design and develop the radio function modules and systems, which are then provided in a service manner. Therefore, different services are cooperatively integrated to develop a large-scale testbed or demonstration system over networks.

All these characteristics of SORA lead to the significant improvement of flexibility and compatibility during the processes of radio system design, development, and implementation, especially for cooperative research, test, and exploitation.

2.2. Framework. The framework of our proposed SORA is summarized in Figure 1, where three terms are considered, namely, radio infrastructures, services, and applications. Specifically, radio infrastructures provide the basic function of radios (e.g., encoder, modulator-demodulator (MODEM), RF module, and antenna, etc.), services that act the infrastructure functionalities (i.e., blocks of the radio system that can be assembled), and radio communication applications are implemented through design and integration of various services.

The three terms of the framework are specified as follows.

2.2.1. Radio Infrastructures. It is known that different wireless networks (e.g., 2G/3G, Wi-Fi, and WiMAX) exist in the current wireless environment, simultaneously. In such environment, the radio infrastructures can be characterized by the heterogeneity, as they could be based on different development platforms and may belong to different operators and networks. In Figure 1, three components are considered as the infrastructures, including antennas, RF modules, and multiple programmable processors. With the development of digital telecommunications, software-based methods are widely used at the back of antennas and RF modules, where different functions, including encoding, modulation, and equalization, can be implemented in application-specific integrated circuits (ASIC), field programmable gate arrays (FPGA), or digital signal processors (DSP). Therefore, in our proposed structure, these processors are included as the infrastructures of radio systems, while different radio infrastructures are encapsulated as services using the standard web service interfaces.

2.2.2. Services. Using functionality description, different radio infrastructures are encapsulated and abstracted from the corresponding software/hardware platforms. The abstracted infrastructure is regarded as a service to be published, discovered, and invoked over networks. In our proposed architecture, the services are classified into four different categories, namely, waveform services, information security services, sensing services, and cognition services. The waveform services include functions of encoding, modulation, and protocols for interaction; the sensing services provide the RF environment to radios; and the cognition services enable reasoning and learning from the experiences. According to the requirements of different users, these services are employed to form various applications, where the service broker is considered to perform the management for service registration, publication, and discovery.

2.2.3. Applications. Through the service integration, a virtual radio is constructed to execute the different applications of radio communication. The applications, for example, voices, short messages, and multimedia, and so forth, are then converted into digital signals using the source encoder. Note that the source encoder can be regarded as a service.

According to the framework of our proposed SORA systems, we can show that the service layer plays a crucial

role, since different radio infrastructures are encapsulated and abstracted as services to support the heterogeneity of underlying development platforms. Comparing to the traditional radio architectures, the services are deployed distributedly over different networks with our proposed SORA configuration, while a virtual radio system is constructed through integration of appropriate services to meet the requirements of different users. From this perspective, SORA is regarded as an agile virtualization method to assemble radio function modules in a network-based concept.

2.3. Technical Standards. The implementation of SORA is based on a series of web service standards, including the web service description language (WSDL) for service description, the simple object access protocol (SOAP) for service interaction, and the universal description, discovery, and integration (UDDI) for service registration, publishing, discovery, and integration, respectively.

It is noteworthy that these web service standards are open access and can be employed on multiple underlying protocols. For example, multiple physical and link layer protocols for different physical environments are usually considered with different transport and network protocols, including HTTP, TCP, UDP, and IP, for different network environments. Using these web service technologies, SORA can provide various protocols, interfaces, and platforms to support the underlying complexity. Thus, the SORA-based radio systems can be flexibly deployed, developed, implemented, and managed.

The web service standards used in SORA are briefly introduced as follows.

2.3.1. WSDL. IT is a description language, which is based on extensible markup language (XML) to define the functionality of a web service. A service provider creates a WSDL document to describe the interface, message format, and network address. Since the information of a WSDL file is published in a service broker, the users are able to obtain the information of the available operations for the service. The XML Schema is used to define the data types in the WSDL document.

2.3.2. SOAP. IT is a protocol specification to exchange messages for the implementation of web services. SOAP is based on XML with the characteristics of extensibility, neutrality, and independence. Hence, SOAP provides a high-level interoperation capability over different hardware platforms, operating systems, and programming languages.

2.3.3. UDDI. IT is a set of XML-based protocols of the service broker (viz., the registry), which are considered for web service registration, publishing, and discovery. The UDDI registry is interrogated by the SOAP messages to provide an access to the WSDL documents that are required to interact with the web services listed in the registry.

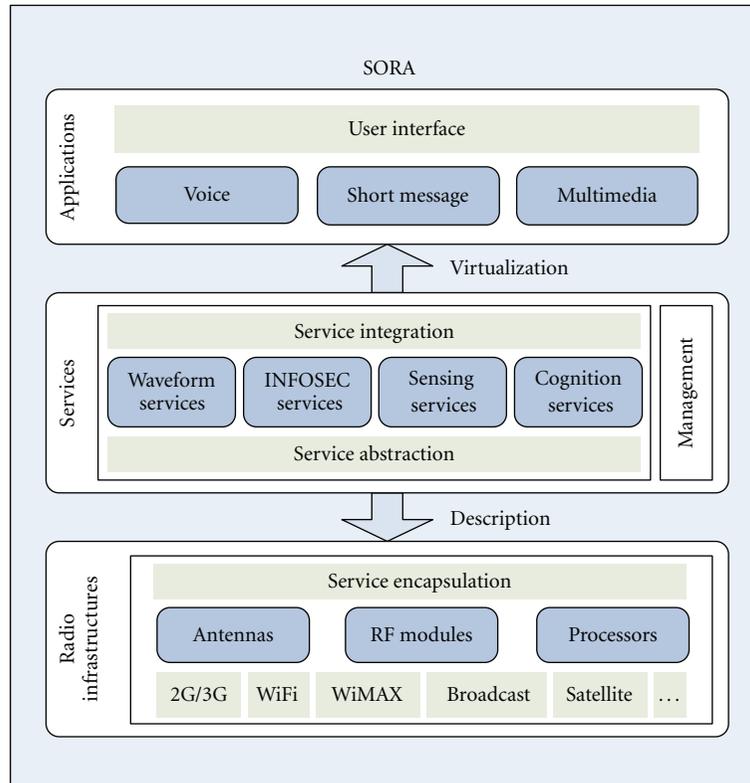


FIGURE 1: Framework of SORA.

According to the above introduction, different web service standards are defined and expressed in XML format. Therefore, the cross-platform sharing and reusing capabilities of XML are inherited to enjoy the flexibility and compatibility of SORA-based radio systems.

2.4. Use Case. A use case to illustrate the cooperation in developing an SORA-based CR access network is shown in Figure 2. In the proposed scenario, there are two operators and two research institutes, which are deployed over wide areas. Each one is considered with an individual network connected to an IP core network. It shows that although Operator A takes charge of a radio access network (RAN) with no cognitive capabilities, there have been some high-performance baseband processing and modem modules. On the contrary, Operator B possesses an expert system. Moreover, it is known that Institute A concentrates on the design and development of various sensors, while the artificial intelligence technologies and learning theory are developed at the Institute B.

Suppose that Operator B would like to build a CR access network for testing, however, due to the limited funds, the cooperation with Operator A is considered, where the research institutes A and B could work together to develop the SORA-based testbed. The process of the testbed development can be divided into four steps, including the service encapsulation, publication, discovery, and orchestration, which are defined as follows.

2.4.1. Service Encapsulation. All the participants, including Operator A, Operator B, Institute A, and Institute B, encapsulate their individual CR function modules as the invocable web services. Thus, these participants act as the service providers that generate the WSDL documents to define different service functionalities.

2.4.2. Service Publication. The WSDL documents are published in a UDDI service registry over the IP core network, where the registry is developed by a third party for public service publication and discovery.

2.4.3. Service Discovery. Operator B accesses to the service registry by utilizing the SOAP messages to find out the required services in the service list. Correspondingly, the locations, interfaces, and operations of different services are discovered using the WSDL documents.

2.4.4. Service Orchestration. According to the order of the signal processing in the CR system, different discovered services are orchestrated by the Operator B. Then, the services are invoked through the SOAP messages, where a virtual SORA-based CR access network is finally constructed.

The process mentioned above can be particularly used to develop the access point (AP) for the CR access networks. The AP is developed as a service client together with antenna and RF modules to provide the functionalities of service

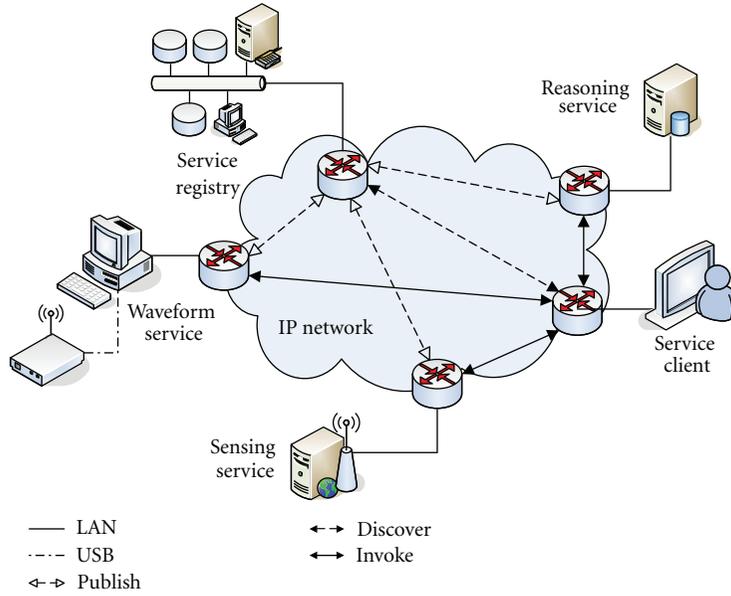


FIGURE 3: CR testbed based on SORA.

3.1.3. *Services.* The service programs and the registry are developed on the Tomcat web servers. Since the Tomcat is based on java, a java runtime environment is built by eclipse to provide the services. Axis2 is used to generate the WSDL documents and manage the SOAP messages for the services.

The hardware and software tools are summarized in Table 1, where the development tools of different service functionalities (including sensing, waveform, and reasoning) and the simulation of the RF environment are introduced.

3.2. *Service Encapsulation.* From Table 1, it is noticed that the sensing, waveform, and reasoning function modules are developed in different software tools and programming languages. Since the Tomcat is a java-based web server tool, the software modules are encapsulated into a java-oriented service.

The service encapsulation plays a key role in the process of development, as it is crucial to develop the function modules of the SORA-based CR testbed. Thus, two steps are considered for the encapsulation as follows. The SOAP interaction is first enabled based on java, and then the WSDL documents is generated for service description.

As a function module interacts with SOAP messages based on java, according to web service deployment standard, the original program module, which can be developed in C, C++, or Python, is additionally equipped with a java interface. This work is implemented on the platform of Eclipse. As shown in Figure 4, the java interface program module communicates with the original function module through a socket connection. As an SOAP request arrives at the java interface, the function module will be informed and return the data processing result to the java interface, then the result will be sent back to the requester through an SOAP message. Thus, the java interface enables the

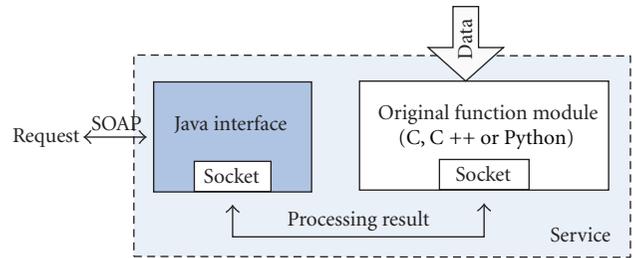


FIGURE 4: Service encapsulation.

original function module to be encapsulated as a service and communicate in SOAP contracts.

To make the encapsulated service independent from the original programming language, the additional java interface should be defined in a neutral manner. This work can be achieved by using Axis2 that generates a WSDL document for the service description. When the java interface is developed in eclipse, a “java” file is generated. The related WSDL document can then be obtained automatically by loading the java file to Axis2. Note that Axis2 can be used to analyze and generate the SOAP messages.

3.3. *DSA Demonstration.* The process of service publication, discovery, orchestration, and invocation is demonstrated using a DSA scenario. In this scenario, the vector signal generator (VSG) randomly generates different waveforms in different frequency bands to simulate the appearances of primary users. As for the secondary user, the SORA-based CR system dynamically chooses and accesses the spectrum holes to communicate while avoiding interference with the primary users.

TABLE 1: Development tools for The CR test base on SORA.

Element	Hardware	Software			
		Function	Solution	Language	Developer
Service registry	Host	Web server	Tomcat	Java	Apache Software foundation
		UDDI	jUDDI	Java	Apache Software foundation
		Database	MySQL	C and C++	MySQL AB
Service client	Host	Orchestration	Center controller	Python	Self-development
		SOAP client	SUDS	Python	Jeff Ortel
Service	Build on each service host	Web server	Tomcat	Java	Apache Software foundation
		Java environment	Eclipse	Java	Eclipse foundation
		SOAP & WSDL enabler	Axis2	Java	Apache Software foundation
	USRP and host	Spectrum sensing	Energy detection	Python	Self-development
	USRP and host	Reconfigurable waveform	GNU radio	C++	Massachusetts Institute of Technology
RF environment	Host	Reasoning	CLIPS	C	NASA-Johnson Space Center
RF environment	Vector signal generator	Dynamic spectrum	Matlab	Matlab	Mathworks

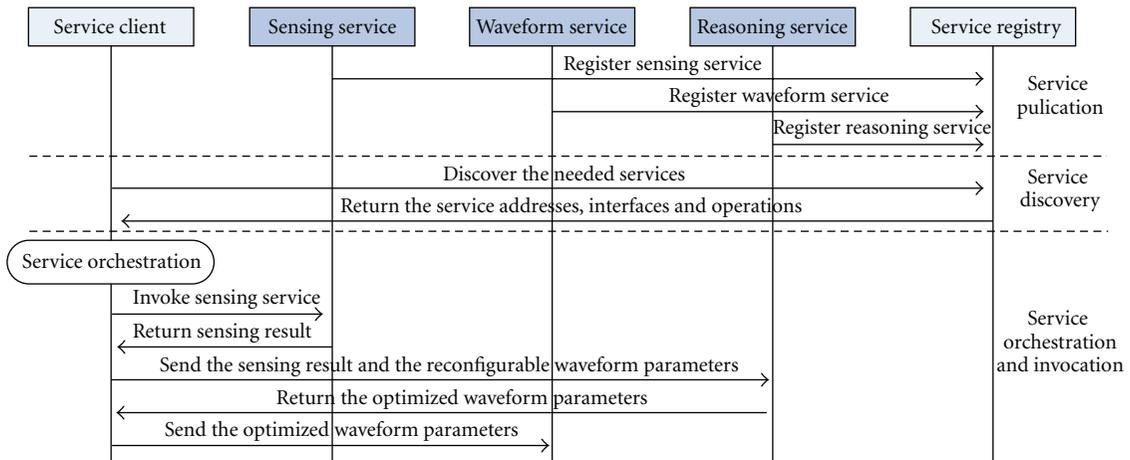


FIGURE 5: The information flow of service publication, discovery, orchestration, and invocation.

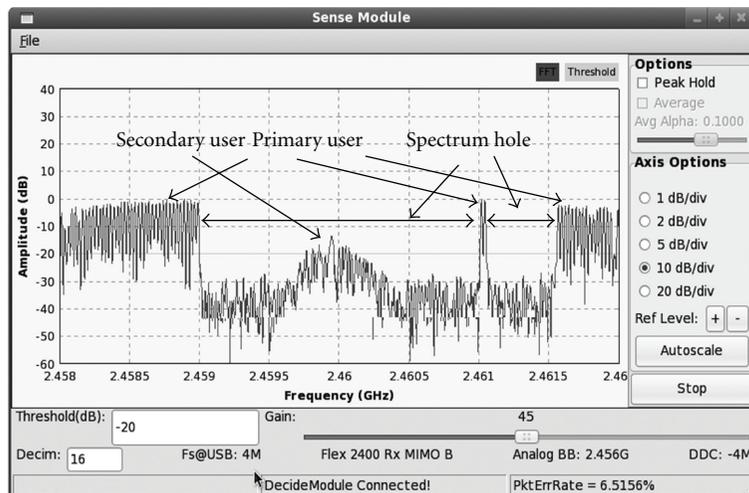


FIGURE 6: DSA demonstration.

The information flow of service publication, discovery, orchestration, and invocation is shown in Figure 5. The WSDL documents are published in the registry by following the process of service registration. On the other hand, the service client discovers the services and obtains their WSDL documents, and then orchestrates the services.

According to the orchestration, the sensing service is invoked by the client to start and keep sensing the spectrum, while the current sensing results are returned to the client. Then, the client sends the sensing results and reconfigurable waveform parameters to the reasoning service. After that, the reasoning service decides the optimized waveform parameters in terms of the relating radio knowledge and returns the results to the client. Finally, the client sends the optimized waveform parameters to the waveform service. Waveforms at the transmitter and the receiver are configured, respectively, and then the communication starts.

When a primary user appears at the frequency band that is being used by the secondary user, the process of service invocation according to the orchestration is implemented again. Thus, the DSA is implemented.

Figure 6 shows a screenshot of the DSA demonstration on the SORA-based CR testbed. It is illustrated that the testbed can dynamically choose and access the spectrum hole with the maximum bandwidth to avoid interference with the primary users and satisfy the spectrum requirement of the secondary user. Then, the spectrum usage efficiency can be increased. Although SORA may not improve the performance of CR systems, it certainly facilitates the flexibility of system development and implementation.

4. Conclusion

In this paper, we have proposed a novel M2M architecture for future CR networks, namely, the SORA. We have first introduced the difficulties faced by the M2M network in future large-scale CR networks and recently cooperative research activities, where the SOA provides a solution. Then, we have proposed our SORA and the components to construct CR networks. Comparing to the CROSS, it has been shown that our proposed SORA provides a higher flexibility and compatibility in cooperative development of CR network. Moreover, we have presented our SORA-based CR testbed to demonstrate the feasibility of the architecture.

In conclusion, the proposed SORA shows the characteristics of loose coupling, reusability, scalability, platform-neutrality, and cost reduction. Therefore, SORA is able to facilitate the M2M communications in future CR networks.

Acknowledgments

This paper was supported by the National Basic Research Program of China (973 Program, no. 2009CB320403), National Natural Science Foundation of China (nos. 60832006, 60832008), and National Science and Technology Major Project (no. 2009 ZX03007-004).

References

- [1] P. Pawelczak, K. Nolan, L. Doyle, S. Oh, and D. Cabric, "Cognitive radio: ten years of experimentation and development," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 90–100, 2011.
- [2] Software Communications Architecture Specification (v2.2), http://www.eng.auburn.edu/users/hamilton/security/SDR/ SRD.Release_V2.2.pdf.
- [3] COST Action IC0902, <http://newyork.ing.uniroma1.it/ IC0902/>.
- [4] Web Service Architecture, <http://www.w3.org/TR/ws-arch/>.
- [5] Service Oriented Architecture, http://en.wikipedia.org/wiki/Service-oriented_architecture.
- [6] S. K. Zhang, J. W. Zhang, and W. Li, "Design of M2M platform based on J2EE and SOA," in *Proceedings of the 1st International Conference on E-Business and E-Government (ICEE '10)*, pp. 2029–2032, Guangzhou, China, May 2010.
- [7] Virginia Tech Cognitive Radio Open Source System, <http://ossie.wireless.vt.edu/trac/wiki/Cross>.
- [8] Q. Duan, "Applying the service-oriented architecture for network discovery and selection in the next generation wireless mobile networks," in *Proceedings of the 12th International Conference on Network-Based Information Systems (NBIS '09)*, pp. 380–385, Indianapolis, Ind, USA, August 2009.
- [9] C. L. Wu, C. F. Liao, and L. C. Fu, "Service-oriented smart-home architecture based on OSGi and mobile-agent technology," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 37, no. 2, pp. 193–205, 2007.
- [10] K. Y. Yee, A. W. Tiong, F. S. Tsai, and R. Kanagasabai, "OntoMobiLe: a generic ontology-centric service-oriented architecture for mobile learning," in *Proceedings of the 10th International Conference on Mobile Data Management (MDM '09)*, pp. 631–636, Taipei, Taiwan, May 2009.
- [11] H. Neema, A. Kashyap, R. Kereskenyi, Y. Xue, and G. Karsai, "SOAMANET: a tool for evaluating service-oriented architectures on mobile ad-hoc networks," in *Proceedings of the IEEE/ACM Symposium on Distributed Simulation and Real-Time Applications*, pp. 179–188, Virginia, Va, USA, October 2010.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

