

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



CC creative commons
COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

BY: **Attribution.** You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

The Architecture of a Web Service-Based Remote Control Service System

Xi Guo

Department of Computer Science
Loughborough University
United Kingdom LE11 3TU
x.guo@lboro.ac.uk

Paul W. H. Chung

Department of Computer Science
Loughborough University
United Kingdom LE11 3TU
p.w.h.chung@lboro.ac.uk

ABSTRACT

There is increasing attention recently in the development of remote control and monitoring. Many applications and experiments have been done to apply internet technology in this field with the aim of developing a general and expendable architecture. In this paper, a novel Web Service-based Remote Control Service System (WSRCSS) is proposed. Compared with other designs, this system focuses on offering a service development and delivery environment. The system architecture and data structure for the system are given. Related works are also discussed in this paper.

Keywords

Remote control, web service, system architecture.

1. INTRODUCTION

Since 1901, when Leonardo Torres-Quevedo laid down the modern wireless remote-control operation principles, the development in this area has not stopped [1]. The idea of controlling home devices remotely has now drawn increasing attention. Various approaches have been explored to help people access their home equipment, such as lights, heater, and safety and security devices, remotely. [2-5]. The advantages of using internet for remote control includes easy access and low cost. The development of architectures for remote control is discussed in [6]. The current focus for research in architecture issues is in many-to-many control, which is a hotbed for discussion. Service-oriented architecture forms the basic framework for remote control, in which controlled devices are considered as services. Different variations are proposed in the literature. Wang, et al. [7] proposed a web-based remote control service system. Ha, et al. [8-10] proposed a Ubiquitous Robotic Service Framework. A Gator Tech Smart House is introduced in [11, 12]. Many of the current designs realize the importance of integrating web resources for remote control. However, little consideration is given to establishing a remote control services development environment.

In this paper, a novel web service-based architecture that supports many-to-many control is proposed. Not only could this

architecture offer multi-access for users at anytime and in anywhere, it also provides an environment for developing remote control services. The rest of the paper is organized as follows. Firstly, the system architecture is introduced. The Service Server, a central part of the system, is then described. This is followed by the design of the data structure. After the discussion on related work, the paper ends with the conclusion and future work section.

2. SYSTEM ARCHITECTURE

As illustrated in Figure 1, the architecture of the Remote Control Service System consists of the following layers:

UAL (User Application Layer) which implements the user applications;
SLL (Service Logic Layer) which provides different services for use at the UAL level;
SDL (Service Development Layer) which supports service development and resource organization;
RL (Resources Layer) which maintains knowledge and system information;
HLL (Hardware Logic Layer) which manipulates the controlled hardware.

UAL is located on the user side. A user can access the system using mobile phones, computers, and PDAs. There is no specific program required, except the use of an internet browser for logging into the system and accessing the services.

SLL is the service logic layer. In this layer, different services can be developed. Instead of just simply realizing control processes, intelligent services can be developed and provided. Example services may be session maintenance and dynamic user interface (UI) generation. The role of session maintenance service is to supervise the transactions between a user and certain devices under control. In the event of a communication connection is broken between the server and a user then the controlled devices will be brought to a safe state automatically and the session will be resumed when the communication connection is re-established. The dynamic UI service will allow different user interfaces to be created dynamically to suit specific user requirements.

SDL consists of Service Development Support Logic (SDSL) that organizes different web resources to support service development. On the one hand, it offers opportunities for new resources to be inserted into the system. On the other hand, it allows third parties to develop new control services for use at the SLL level.

RL is a layer where important remote control resources are maintained and stored. These resources could be accessed by

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

iiWAS2008, November 24–26, 2008, Linz, Austria.

(c) 2008 ACM 978-1-60558-349-5/08/0011 \$5.00.

SDSL and used to assist the intelligent control services development.

HLL consists of applications on the controlled device side. An application in this layer is mainly used for accessing and controlling devices, collecting device information and transferring it to the server. These devices are connected to the server via the internet through gateways, computers or other internet access methods. Since they are also considered as web service provided over the internet, a unified machine access API is needed.

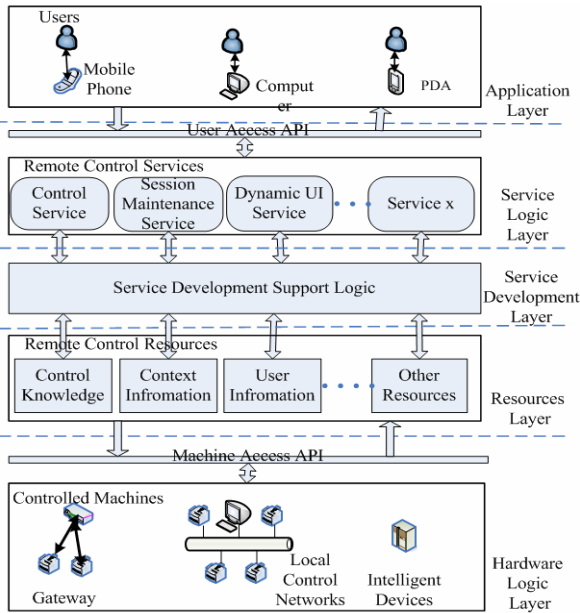


Figure 1. System architecture diagram

Compared with other designs, this architecture addresses the issue of development of control services. By integrating and offering proper APIs for accessing resources, various services, such as basic control services, control session maintenance services, and dynamic UI services, can be developed using this platform. The idea is that controlled devices are considered to be web resources available over the internet. These web resources are organized at the SDL level using SDSL. By incorporating knowledge about control, devices and users, intelligent services can be provided.

3. SERVICE SERVER

As described above, the architecture focuses on supporting service development using integrated resources. Besides controlled devices, there are also other related resources available, such as device knowledge and context information. But how are these resources organized? In most designs, resources are connected directly to certain kind of services. This approach limits how the resources can be flexibly adapted to develop new services. To address this problem, a service server is proposed which separates services from resources. Figure 2 shows the detailed architecture of the Service Server, which is located between SLS (Service Logic Subsystem) and HLS (Hardware Logic Subsystem). According to the system architecture, each layer has a corresponding subsystem. SLS applies various service functions as well as the APIs in SLL. HLS offers both sensor and machine control applications and APIs for web services on the controlled

machines side. The Service Server, which is an independent server on the internet, consists of SDS (Service Development Subsystem) and RS (Resources Subsystem).

SDS offers methods for organizing resources to support the development of services. It consists of the following components: SDCC (Service Develop Control Center) which controls the whole SDS and offers the service developer control resource using methods; Framework which controls the access to the resources; SDC (Service Data Center) which provides the access methods for the related control background knowledge; and DSC (Device Service Center) which manages the remote controlled machine web service resources. By using related control background knowledge in SDC, intelligent control service could be developed and applied to control device web service via DSC.

RS is the place where all the knowledge and web resources are collected and stored. There are two kinds of resources. One is the web service for sensors and machines which need to be controlled. The register, which is used for finding and binding of these web services, will be implemented by an agent in DSC. This supports enables the communication between the control logic and the remote devices. The control background knowledge captures the semantic that is suitable for intelligent service development.

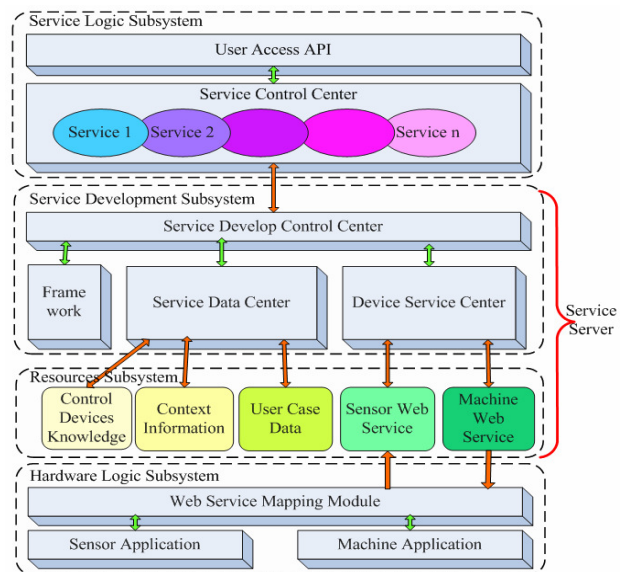


Figure 2. Service Server architecture

The result for using such a service server is to separate the control logic from the control resources, including the remote communication ability and the supporting data resources. Also, remote control services are independent from basic device web services, which are considered as a special kind of web resource. Therefore, service developers can focus on control logic development without having to worry about resource management. The Service Server also allows new services or new knowledge to be inserted using the plug-and-play model.

4. DATA STRUCTURE IN SERVICE SERVER

This section considers in more details the support that the system provides for service development. For remote control, both the

abilities to communicate with the remote machine and to control it intelligently are important. In this system, remote devices are accessed as web resources, so communication can be established using normal web service. However, domain knowledge is needed to support the development of intelligent services. The knowledge needs to be represented in a way that provides the semantic for reasoning.

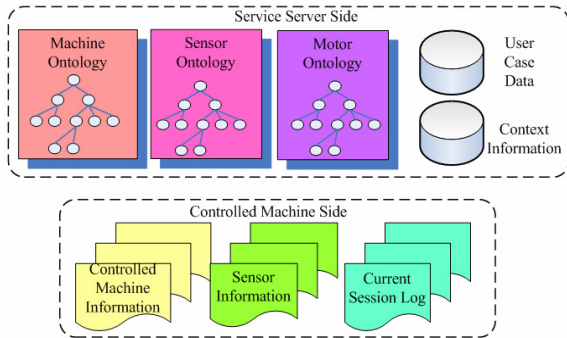


Figure 3. Data represented in the WSRCSS

Figure 3 shows the data that needs to be represented in the system. The data is separated into two parts: controlled machine side and service server side. On the controlled machine side, Controlled Machine information and Sensor Information are stored to indicate the features of a specific instance of an individual machine, such as its brand, types and ID. Current Session Log keeps a record of the current control process of that machine so that it could be accessed if the control process is disturbed. On the other hand, data on the service server side represent knowledge in more general terms. There are ontology resources for machines, sensors and motors, which provides class level information for the resources. User Case Data represents the user control process history, which can be used for reasoning about control services. In order to generate better solution according to current situation, Context Information is also included in the data source.

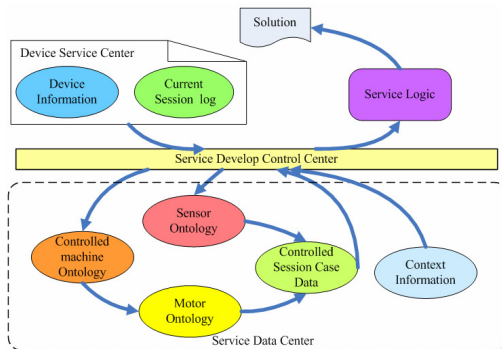


Figure 4. Data flow in the system during service developing

The data is organized and stored in the Service Server. The information can be accessed by service logic if required. Figure 4 illustrates the data flow for intelligent service logic. When a machine is registered in the system, devices information, which includes controlled machines and sensors, will be transferred to DSC. The data will be passed to SDCC and analyzed using ontology knowledge from SDC, so that the Service Server will know what kind of machine is being controlled and what the allowed actions are. The Service Server will also have information

about the sensors. This information, as well as the necessary context information, is then transferred to Service Logic, which is developed in SCC. Obtaining this information, Service Logic begins reasoning and generates new solutions for control service.

5. RELATED WORK

Some related works are reported in [7], [9] and [12]. All these projects aim at integrating web resources and develop servers or mid-wares to support certain remote control services. In [7], the control server is used to apply web resources in to generate and maintain control processes, but the services are fixed and are difficult to change. The design in [9] tried to integrate various web resources using a Service Agent Platform. However, very little consideration is given to service development. The design of a system that offers a smart-space middleware is given in [12]. It addresses the problem of integrating new elements (either knowledge or service) into the system. The advantage is the middleware service composition method. Nevertheless, too much emphasis is on resource integration which means that the design lacks support for service development.

The designs proposed in the previous work described above could be considered as providing methods for integrating web resources to support remote control. However, the idea behind the design in this paper goes a step beyond by separating the control logic from the control resources, and by separating the remote control service from resources. In this way new services could be developed easier. These differences are illustrated in Figure 5. The designs in [7], [9] and [12] follow the model shown in Figure 5 (a), which is to integrate related resources and used them to develop specific control applications. In contrast with that, the design in this paper (Figure 5 (b)) focuses on integrating resources that are offered to a Service Development Environment for developing all kinds of services.

6. SUMMARY AND FUTURE WORK

In this paper a design for a web service-based remote control service system is proposed. It builds on the basic many-to-many architecture. The system not only supports remote control for multi-users, but also offers a flexible platform for developing services. The main idea of the design is to separate the control logic from the control data and independent remote control services from basic resources. Further work is needed to address the problem of data representation and implementation. Once the framework is in place it will be evaluated by developing different services using the approach described.

7. ACKNOWLEDGEMENT

X. Guo is partly funded by a Loughborough University studentship. The authors would also like to thank Dr Zhining Liao for helpful discussions.

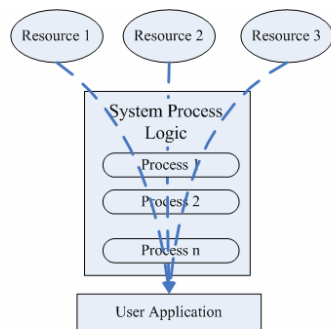


Figure (a)

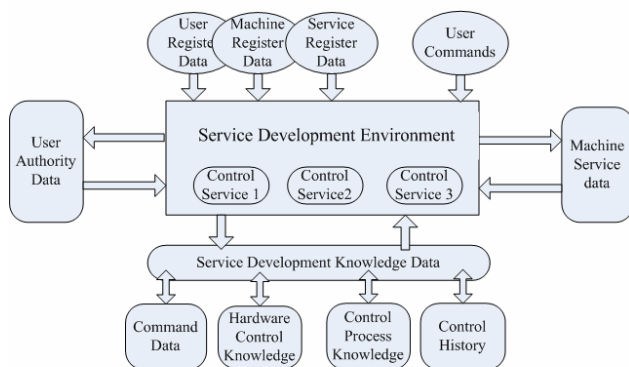


Figure (b)

Figure 5. Difference between WSRCSS and other architectures

REFERENCES

- [1] "IEEE History Center: Early Developments in Remote Control, 1901"
http://www.ieee.org/web/aboutus/history_center/telekine.htm 1, last accessed on 19/08/2008
- [2] Ito, K. and Murai, R. Abstraction and compression of information utilizing real world for controlling remote controlled robot-Application to snake-like robot. In 2007 IEEE International Conference on Networking, Sensing and Control (London, UK, April 2007), 674-679.
- [3] Burger, E.W. and Frieder, O. A Novel System for Remote Control of Household Devices Using Digital IP Phones. In International Conference on Consumer Electronics, (2006), 183-184.
- [4] Kim, K.S., Park, C., Seo, K.S., Chung, I.Y. and Lee, J. ZigBee and The UPnP Expansion for Home Network Electrical Appliance Control on the Internet. In the 9th International Conference on Advanced Communication Technology, (2007), 1857-1860.
- [5] Wang, Q., Liu, S. and Wang, Z., A New Internet Architecture for Robot Remote Control. In IEEE/RSJ International Conference on Intelligent Robots and Systems (Beijing, China, Oct. 2006), 4989-4993.
- [6] Liu, W., Guo, X., Chung, P.W.H. The development of remote control for processing plants Proceedings. In the 35th International Conference of Slovak Society of Chemical Engineering, 2008, paper 221, Slovak Society of Chemical Engineering, ISBN 978-80-227-2903-1.
- [7] Wang, C., Teng, F., Wang, Y. and Ma, G. Web-based remote control service system. In IEEE International Symposium on Industrial Electronics, (2003), 337-341 vol. 331.
- [8] Ha, Y.G., Sohn, J.C. and Cho, Y.J. Service-oriented integration of networked robots with ubiquitous sensors and devices using the semantic Web services technology. In IEEE/RSJ International Conference on Intelligent Robots and Systems, (2005), 3947-3952.
- [9] Ha, Y.G., Sohn, J.C. and Cho, Y.J. ubiHome: An Infrastructure for Ubiquitous Home Network Services. In IEEE International Symposium on Consumer Electronics, (2007), 1-6.
- [10] Ha, Y.G., Sohn, J.C., Cho, Y.J. and Yoon, H. A robotic service framework supporting automated integration of ubiquitous sensors and devices, Elsevier, 2007, 657-679.
- [11] <http://www.icta.ufl.edu/gt.htm>, last accessed on 19/08/2008
- [12] Helal, S., Mann, W., El-Zabadani, H., King, J., Kaddoura, Y. and Jansen, E. The Gator Tech Smart House: A Programmable Pervasive Space. IEEE Computer Society Press, 2005, 50-60.