Automated Teleoperation of Web-Based Devices Using Semantic Web Services

Young-guk Ha[1], Jaehong Kim[1], Minsu Jang[1], Joo-chan Sohn[1], and Hyunsoo Yoon[2]

[1] Intelligent Robot Research Division, Electronics and Telecommunications Research Institute, 161 Gajeong-dong, Yuseong-gu, Daejeon, Korea
{ygha, jhkim504, minus, jcsohn}@etri.re.kr
http://www.etri.re.kr/e_etri/
[2] Computer Science Division, Korea Advanced Institute of Science and Technology, 373-1 Kusung-dong, Yuseong-gu, Daejeon, Korea
hyoon@camars.kaist.ac.kr

Abstract. In this paper, we present SWATS which supports task-oriented automated teleoperation of Web-based devices. The proposed system employs Semantic Web Services technology and AI planning technique to achieve operational automaticity.

1 Introduction

Internet-based teleoperation systems are mainly focused on remote control of networked devices, such as mobile robots or digital appliances through the Internet. In recent years, several attempts have been made to develop Internet-based teleoperation systems using the Web technology: i.e. USC's teleoperated excavation system Mercury and CMU's indoor mobile robot Xavier. In such systems, each control command generally provides a behavior-level control over the device with or without built-in autonomy. That is, to achieve a desired task, a human operator monitors remote devices through Web-cams or sensors, and manually sends a sequence of appropriate control commands and parameters with a Web browser as shown in Fig. 1-a.

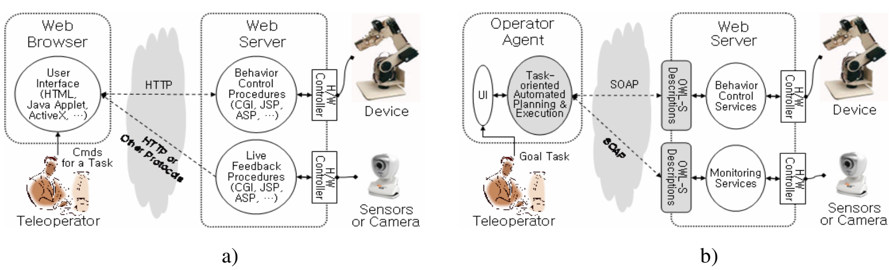

Fig. 1. Comparison of a) Traditional Web-based teleoperation and b) SWATS approach

In this paper, we propose SWATS (Semantic-Web-service-based Automated Teleoperation System) which supports task-oriented automated teleoperation of Web-

M. Ali and F. Esposito (Eds.): IEA/AIE 2005, LNAI 3533, pp. 185 – 188, 2005.

based devices. As shown in Fig. 1-b, SWATS employs Semantic Web Services technology [3] and automated planning to provide operational automaticity. That is, semantics of behaviors and interfaces for Web-based devices and sensors are encoded in OWL-S (Web Ontology Language for Services) [4] as Web services, so that operator agents can automatically plan operation processes for the requested tasks by reasoning about their semantics in OWL-S. And then the operator agents can achieve requested task goals by communicating with the devices and sensors through SOAP (Simple Object Access Protocol) messaging according to the operation processes.

2 Architecture of SWATS

Fig. 2 shows the architecture of SWATS. The OA, as an intelligent planning and service requester agent, plays the major role of automated teleoperation in SWATS architecture. A teleoperator inputs operation task as a goal to achieve and optionally initial operation contexts with the User Interface. Based on the input task and initial contexts, the Task Planning Module discovers required knowledge for planning through the Semantic Discovery Module and automatically generates a feasible task plan based on HTN (Hierarchical Task Network) planning [2]. To search the OKR for planning knowledge described in OWL-S, the Semantic Discovery Module generates appropriate semantic queries based on the input task description and sends them to the OKR. The Process Execution Module translates the task plan into the BPEL [1] process and executes the process through the Web services communication stack.

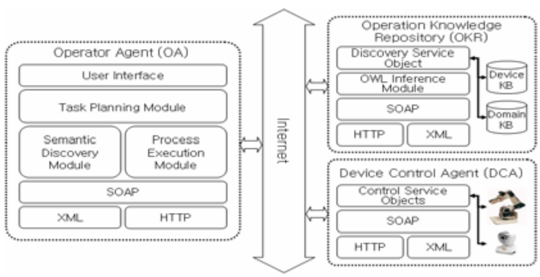

Fig. 2. The architecture of SWATS consists of three major components, which are Operator Agent (OA), Device Control Agent (DCA) and Operation Knowledge Repository (OKR)

The DCA is implementation of behavior control Web services for a device including monitoring services for sensor and camera devices. Each DCA can have Control Service Objects for one device or multiple devices which may work cooperatively, for instance an air-conditioning device and temperature sensors. And the DCA also has a Web services communication stack to communicate with an OA. The OKR contains KBs for task domain, device behaviors and interfaces which are used in automated task planning and process execution. The Domain KB stores OWL-S ontology of composite processes and internal data flows describing task domain knowledge. The Device KB stores OWL-S ontology of atomic processes and corresponding grounding descriptions as device service descriptions. The OKR includes the Discovery Service Object to handle knowledge discovery queries with semantic predicates. It uses the OWL Inference Module to reason about the semantic predicates. The OKR also includes a Web services communication stack because it works as a Web service itself. Fig. 3 shows automated teleoperation procedure of SWATS.

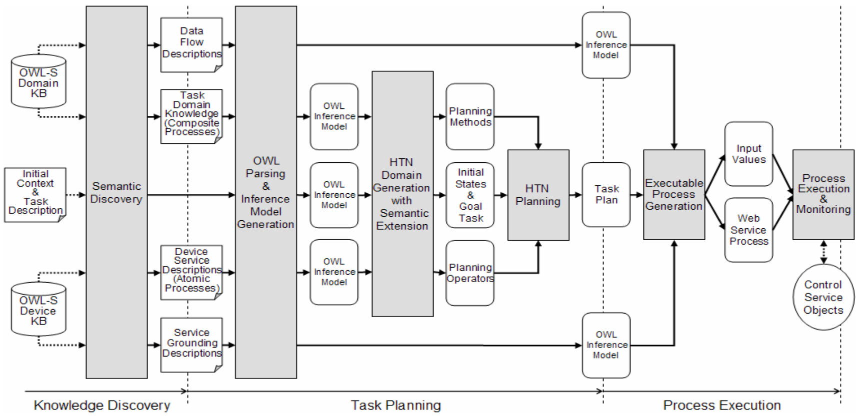

Fig. 3. Automated teleoperation procedure of SWATS consists of knowledge discovery, operational task planning and process execution phases

3 Implementation of SWATS

The prototype implementation of SWATS is shown in Fig. 4. The prototype DCA contains control service objects, i.e. MoveTo, GetImage, SendImage and etc, for the mobile robot. For an experiment, we generate OWL-S descriptions of the control services and task domain knowledge for home telesecurity services, i.e. ReportHomeStatus service which automatically operates a mobile robot to move to the specified place in the house, take a picture and send the picture to the outdoor user.

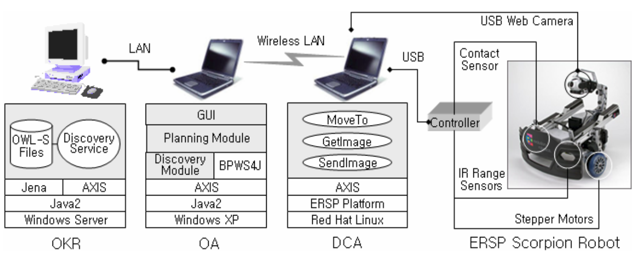

Fig. 4. The SWATS prototype implementation for automated home telesecurity services

References

1. Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., et al.: BPEL for Web Services, http://www.ibm.com/developerworks/library/ws-bpel/ (2003)
2. Erol, K., Nau, D., Hendler, J.: UMCP: A Sound and Complete Planning Procedure for Hierarchical Task-Network Planning, AIPS-94, Chicago (1994)
3. McIlraith, S.A., Son, T.C., Zeng, H.: The Semantic Web Services, IEEE Intelligent Systems, Vol. 16, Issue 2, IEEE (2001) 46-53
4. The OWL Services Coalition: OWL-S 1.0 Release, http://www.daml.org/services/owl-s/1.0/ (2003)