Towards a Service-Oriented Participatory Design Studio Supported by UCLP

Liu, Sandy; Spencer, Bruce; Brooks, Martin; Jemtrud, M.; Privalov, K.; Spence, John; Savoie, M.; Ho, B.

Publisher's version / Version de l'éditeur:

The Proceedings of the 2006 IEEE International Conference on Services Computing (SCC 2006), September 18-22, 2006, Chicago, Illinois, USA, 2006

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC · CNRC

*Towards a Service-Oriented Participatory Design Studio Supported by UCLP **

Liu, S., Spencer, B., Brooks, M., Jemtrud, M., Privalov, K., Spence, J., Savoie, M., Ho, B.
September 2006

* published in The Proceedings of the 2006 IEEE International Conference on Services Computing (SCC 2006). September 18-22, 2006. Chicago, Illinois, USA. NRC 48738.

Canada

Towards a Service-oriented Participatory Design Studio Supported by UCLP

Sandy Liu, Bruce Spencer, Martin Brooks
National Research Council Canada

Michael Jemtrud and Konstantin Privalov
Carleton University

John Spence, Michel Savoie, Bobby Ho
Communications Research Centre Canada

1. Introduction

The Participatory Design Studio (PDS)[1] provides users across mutiple sites the ability to effectively participate in a common design session. Users are provided a variety of shared resources, including the underlying high-speed network, cameras, displays, sound equipment, large data files, and software applications for communicating and for visualizing the artifact being designed. This paper explores the design and implementation of PDS, and describes how it leverages the benefits of a Service-oriented Architecture (SoA) to provide a highly adaptable, modular, and loosely coupled solution. PDS integrates systems that are hosted on a wide variety of platforms with different management and network domains. This integration goes beyond piping application data from one system to another. Web Services provide a cost-effective solution for integrating these systems into a comprehensive SoA.

The PDS's target user community is architects of buildings, although our design will be generic and applicable to different user communities. Architecture and industrial design are advanced professions requiring collaboration of diverse teams exploiting powerful visualization and modeling tools. Challenging factors such as design complexity, economic and environmental factors, new materials, and construction/manufacturing planning require the design team to access diverse and often distributed expertise. Until now, insufficient bandwidth and crudely coordinated tools have resulted in distributed, task-based modes of collaboration, which often hinders the full participation by members of a distributed design team. Free-flowing multi-person participation is the key to successful problem solving at each stage of the design process. PDS allows manipulating designs, sharing them and visualizing them in 3D and video.

A broadband network (i.e.1-10Gb/s) is ideal for transferring the bandwidth-thirsty multimedia content, although a dedicated broadband network is too expensive for most organizations. The debut of UCLP (User-Controlled Lightpath Provisioning)[3] makes this option viable. The UCLP provisioning Web Services allow users to dynamically assemble a set of lightpaths [2] into a private end-to-end optical network, a so-called APN (Articulated Private Network). Thus the users share the usage of the network and pay only for the time slot when the network is requested. This mechanism greatly enhances the utilization and affordability of the broadband network. It also increases the level of control by the end users, since APN creation is no longer dictated by network administrators, but by the users, e.g. the design teams, possibly with the assistance of the technical staff on site.

The high-speed low-latency APN removes the bandwidth bottleneck. Nevertheless, the design team requires many tools to be integrated, including video-conferencing devices and applications such as the Ruff system from Japan's National Institute of Information and Communications Technology, 3D image rending software such as AutoDesk's Maya, and visualization tools such as IBM's DCV - Deep Computing Visualization. Inspired by CANARIE's vision[4], PDS will provide a user-friendly dashboard for architects to control these tools and instruments with the support of an SoA.

2. System Design

All the core functions of PDS will be provided by Web Services, either as a single service or a combination of services. Figure 1 shows the high level design of the system.

There are three categories of users identified: the physical network administrator, the UCLP user, and the end-user. The end-users are architects and designers who are participating in the design. The physical network administrators are responsible for administrating the optical network and managing the lightpath resources. The UCLP user works with the end-users, and is capable of assembling the light-

[1]This project is funded by CANARIE's Intelligent Infrastructure Program.

[2]A lightpaht is an abstraction of connection between two or more switches in an optical network.

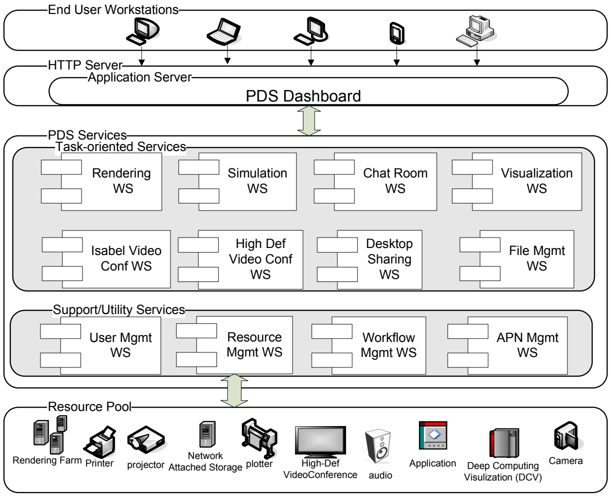

Figure 1. PDS design overview

path resources to create an APN for the end-users. A user's access restrictions is based on his/her category.

The PDS dashboard will be implemented as a web application. A user can access it from any workstation connecting to the Internet (or the APN). The functions of the PDS dashboard are supported by a set of underlying services, shown as the PDS Services block in Figure 1. We divide the services into two groups: task-oriented services and support/utility services. Support/utility services are generic and support the task-oriented services. For instance, each user is authenticated by the User Management Service, which is a utility service; users with the proper security certificates then are granted access to task-oriented services, such as accessing files through the File Management Service.

The Resource Management Service is a key utility service. We define an ontology using OWL DL[5] to semantically define the resources and its dependencies. We define a property `ObjectProperty(pds:requires)` that can be used to specify what is required for a resource to run properly. For example, to configure the High-Definition Video-conference (HDV) system, we need to have a lightpath between the connecting nodes and the Ruff system. The Ruff system requires the display, camera, and audio devices to be configured with appropriate settings. This process is specified in a workflow using BPEL. The Workflow Management Service provides a set of workflows to the users. A workflow repository stores a collection of commonly used configurations such as the setup for HDV. The user-defined workflows are stored in the repository. A verification service determines whether a workflow violates any setup requirements. In addition, a workflow monitoring service is launched when a workflow is being executed.

3. Issues and Discussion

The development of PDS is challenging. Beyong simply wrapping existing equipment and applications, our design accommodates changing demands and includes innovations for issues that are not normally encountered in SoA implementation projects. In this session we discuss a few issues we have encountered.

Our services provision different tools, and for each tool there must exist a service platform to provide access to it. Using Web Services, the straightforward way is to have an HTTP server and an application server running on the machine that co-hosts the application controlling the resource. This is a labor-intensive approach as it requires installation of required servers, a set of related libraries, and configuration of the runtime environment on every machine that interfaces with one or more resources. As the number of Web Services increases, the maintenance work will become unmanageable. In addition, the number of entrances into the network also increases, openning many security holes for malicious access [1]. To increase the manageability, for each network domain we set up an HTTP server outside the firewall, and talk to the application server that is behind the firewall by a private channel (e.g. SSH). The application server then connects to the local resources.

To better manage the resource and workflow, it is desirable for the Resource Management Service to keep track of the states of different resources, but some resources are not designed to work as a service. Some parameters are set within existing applications. One way to get around is to initialize the resource to a ready state before performing an action. Therefore there is no need to keep track of the state, achieving the stateless principle[2].

The PDS project applies SoA in a novel usage area. It also provides a testbed for future research in process management and service computing.

References

[1] K. Birman. The Untrustworthy Web Services Revolution. *IEEE Computer*, 39(2):98–100, February 2006.

[2] T. Erl. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall, 2006.

[3] J. Wu et al. User-managed End-to-end Lightpath Provisioning over CA*Net4. In *Proceeding of the National Fiber Optic Engineers Conference (NFOEC), Orlando, FL, USA*, pages 275–282, September 2003.

[4] B. St.Arnaud. CA*net4 research program update - UCLP roadmap: Web Services workflow for connecting research instruments and sensors to networks. `http://www.canarie.ca`, December 2004.

[5] W3C. OWL Web Ontology Language Reference. `http://www.w3.org/TR/owl-ref/`, 2004.