Towards a RESTful Process of Conference Management System

Pongsathorn Noimanee, and Yachai Limpiyakorn

Abstract— **Several conferences have been regularly organized worldwide each year. The organizing process consists of many major steps including call for papers, paper submission, article reviews, author notification and participant registration. The networking computer system could facilitate managing the activities that involve plenty of remote stakeholders. This paper therefore presents an approach to implementing a conference management system using web service technology fused with the concept of business process modeling to support the workflows of conference management tasks. In this work, the REST protocol is selected for invoking the defined conference managing services, whereas Windows Workflow Foundation is exploited for binding all activities into the process flow. The approach enhances the stateless process to the stateful process. Moreover, the process flow is easily reorganized or customized for the subsequent conferences. The component activities, implemented as a service, within the process flow can be reused as well. The fact that services exchange data through XML standard, the implemented system is platform and language independent.**

Index Terms— **Business Process Modeling, Conference Management System, REST, Web Services**

I. INTRODUCTION

Several conferences have been regularly organized worldwide each year. The organizing process involves plenty of remote stakeholders and consists of many major steps, such as call for papers, paper submission, article reviews, author notification, and participant registration. These activities are resource consuming, for example, distributing papers to the proper reviewers, collecting the reviews results, notifying the authors, and participants registering.

Emerging of the conference facilitator software would ease these routine burdensome tasks. Examples of some commercial software in the market are CyberChair [1] and PCS (Precision Conference Solutions) [2]. However, these commercial systems are costly, platform dependent, and difficult to modify or reuse for the subsequent conferences.

This paper therefore presents an approach for implementing a conference management system using web

Pongsathorn Noimanee Author is with the Computer Engineering Department, Chulalongkorn University, Phyathai, Bangkok 10330 Thailand (e-mail: Pnoimanee@yahoo.com).

Yachai Limpiyakorn Author is with the Computer Engineering Department, Chulalongkorn University, Phyathai, Bangkok 10330 Thailand (corresponding author to provide phone: 662-218-6968; fax: 662-218-6955; e-mail: Yachai.L@chula.ac.th).

service technology fused with the concept of business process modeling to support the workflows of conference management tasks. The Representational State Transfer (REST) is the selected protocol in this work. The Windows Workflow Foundation (WF) is used for managing the process flow, i.e. interconnecting between the provided Application Services, and User Interfaces. Based on the approach, the conference management system implemented is platform and language independent. The services developed in this work can be reused in similar applications. The system also provides the flexibility to modify the steps of the conference workflow.

The contents in this paper are organized as follows: section 2 reviews some existing conference facilitator systems. The technologies selected for the implementation of the proposed approach are described in section 3. Section 4 described the design of the conference management system in this work followed by system implementation presented in section 5. Section 6 finally concludes the work in this paper.

II. CONFERENCE MANAGEMENT SYSTEM

Conference Management Systems typically facilitate the common tasks when organizing a conference. Generally, the system interact with the users of which the roles can be divided into 3 major groups:

- Program Committee Chair is responsible for scheduling of call for papers, paper assignment to Reviewers, and final decision on paper acceptance.

- Program Committee or Reviewer is responsible for reviewing papers and sending the results back to Program Committee Chair

- Author is responsible for submitting abstract and full paper to the conference, as well as registering for the conference attendance.

In addition to the users described above, other roles may include general participants, and System Administrator or Cyber Chair who co-ordinates between those three groups of users.

CyberChair [1] and PCS [2] are examples of the conference management system. CyberChair was developed based on the paper published in the European Conference on Object-Oriented Programming (ECOOP'97). The system is implemented with Python Language on UNIX platform. Fig. 1 illustrates the workflow of the CyberChair conference management system.

PCS is a web-based conference management system of which the prominent features emphasize on submission and

review processes. Moreover, the system installation is easy,
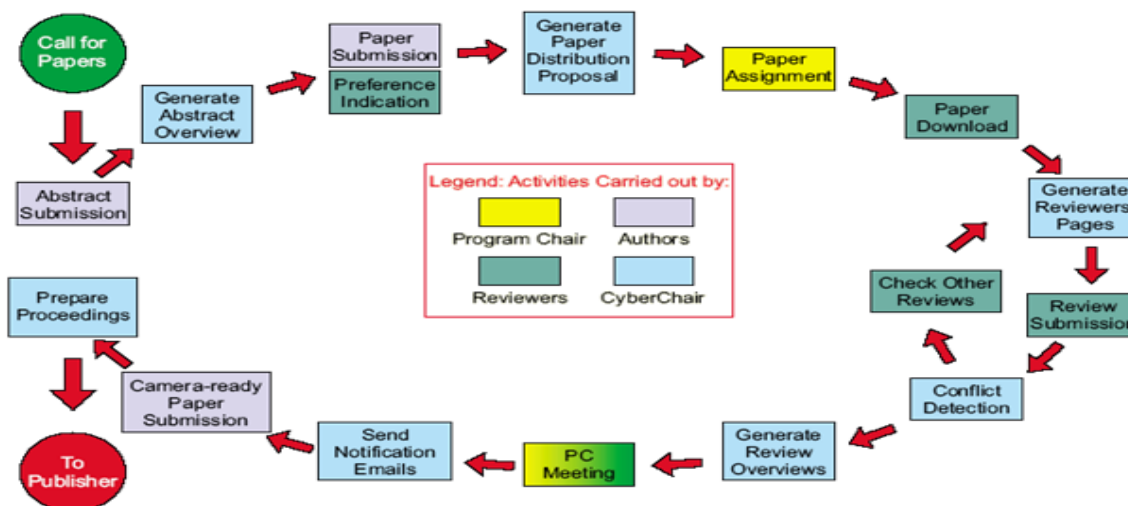and the interfaces are user-friendly

Fig. 1 Workflow of CyberChair [1]

III. TECHNOLOGY

A. Representational State Transfer – REST

REST [3] is an architectural style proposed to design
applications in such a manner that their functionality is
implemented as a set of resources that can be referred to
using an URI identifier. Clients interact with the application
resources via the GET, PUT, POST and DELETE methods
provided by the HTTP protocol to exchange well-defined
representations of them. Resources are stateless (no state
between requests) and loosely coupled (no relationships
with other resources).

REST was proposed by Roy T. Fielding in his doctoral
thesis. In 1994, Fielding developed REST which was the
primary architect of HTTP 1.1, and authored the uniform
resource identifier (URI) generic syntax. REST was
invented as a way to help communicate the basic concepts
underlying the Web. REST is not a strict discipline, nor is it
embodied by a rigid standard or specification. However, it is
generally recognized as having the following three
elements:

- **Resources**: The domain is analyzed and decomposed
into a core set of fundamental entities called resources.
Resources can represent specific entities or concepts. A
method of identifying resources is established, e.g. via
uniform resource identifiers (URIs), queries or search
expressions.

- **Representations**: A form of externally representing the
core resources is established. In general, multiple
representations may be offered for any given resource. For
example, a mapping service might offer graphical
representations as well as data representations.
Representations may be statically fixed or may be requested
on the fly (e.g. through request headers).

- **Operations** : A small set of common operations against
the resource types are identified. These comprise the
common interface described earlier. The most common

REST operations are: add(), delete(), update(), and find().

Web service is a software system designed to support
interoperable machine-to-machine interaction over a
network. An architecture that supports web services, known
as a Service-Oriented Architecture (SOA), covers the
following aspects [4]:

- The dynamic discovery of registered services. This
includes searching for services that meet certain criteria,
especially business criteria such as delivery time, price, etc.

- The organization of services, so that one can easily
understand what a service offers.

- The description of services, so that a service can be
properly invoked. This includes formats and protocols for
invoking the web service.

The current technologies that enable the communication
between various agents include REST described earlier, and
SOAP (Simple Object Access Protocol) [5]. In this research
work, the conference management system is selectively
implemented with REST. The advantages of REST can be
summarized as followings [6], [7]:

- **Building Clients**: REST conforms to the uniform
interface requirements. Therefore, it is much easier for
client developers to understand compared to systems
providing SOAP interfaces. Contrast with the obscure
WSDL file provided by SOAP service provider, the list of
resources a REST server must provide is simple and
unthreatening.

In addition, as opposed to impenetrable lists of
parameters submitted to arbitrarily named methods, REST
manipulates the state of their resources via a standardized
transfer of representations. This would simplify the
complexities of the XML messages.

The statelessness of REST also ensures that a corrupted
or incorrect session might not cause unforeseen problems in
client-server interactions.

- **Building Servers**: The simplicity of REST eases the
developers of client applications. Unlike in a SOAP system,
clients in a REST system are required to submit all the
information necessary to process a request in the header of

the request itself. Therefore, the body of the message need not be examined by the server to route it to the proper segment of code.

On the other hand, the features of statelessness and self-descriptive messaging ease the server developer site in REST systems. Session management is one of the first problems many developers run into when scaling an application to multiple servers.

- **Building for the Future**: REST systems are easier to maintain and extend. Comparing two systems, REST and SOAP, identical in functionality, the steps required to expose new functionality via API in the REST system could be as simple as adding code to handle a new method on an existing resource and adding links to the appropriate pages. Clients can then automatically access the new functionality by browsing the site normally. For the SOAP system, it is required to implement the changes needed to respond to the new message, to modify the WSDL file and distribute the updated WSDL file to clients, who must subsequently process the description file and undergo the appropriate updates to reflect the new functionality. The REST approach is clearly less painful than the SOAP approach. Moreover, unlike SOAP that can hide implementation flaws with the added abstractions, the REST systems are easier maintained due to the existence of constrained set of methods, which result in fewer places to examine when debugging.

B. Business Process Modeling – BPM

Business Process [8] is a collection of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how the work is done within an organization, in contrast to a product's focus on what. A process is thus a specific ordering of work activities across time and place, with a beginning, an end, and clearly defined inputs and outputs, i.e. a structure for action.

Windows Workflow Foundation (WF) [9] is a framework that enables users to create system or human workflows in their applications written for Windows Vista, Windows XP, and the Windows Server 2003 operating systems. It consists of a namespace, an in-process workflow engine, and designers for Visual Studio 2005. Windows Workflow Foundation can be used to solve simple scenarios, such as showing UI controls based on user input, or complex scenarios encountered by large enterprises, such as order processing and inventory control. Windows Workflow Foundation comes with a programming model, a rehostable and customizable workflow engine, and tools for quickly building workflow-enabled applications on Windows.

Scenarios that Windows Workflow Foundation addresses include:

- Enabling workflow within line-of-business applications.
- User-interface page flows.
- Document-centric workflows.
- Human workflows.
- Composite workflows for service-oriented applications.

- Business rule-driven workflows.
- Workflows for systems management.

Windows Workflow Foundation provides a consistent and familiar development experience with other .NET Framework 3.0 technologies, such as Windows Communication (WCF) and Windows Presentation Foundation (WPF). The Windows Workflow Foundation API provides full support for Visual Basic .NET and C#, a specialized workflow compiler, debugging within a workflow, a graphical workflow designer, and developing your workflow completely in code or in markup. Windows Workflow Foundation also provides an extensible model and designer to build custom activities that encapsulate workflow functionality for end users or for reuse across multiple projects.

IV. SYSTEM DESIGN

This paper proposes an approach to implementing a conference management system using Web Services technology and Business Process Modeling. The system architecture layers are depicted in Fig. 2. The process layer enhances the stateless processes to the stateful processes that suit the workflow characteristics of conference management activities, which are the collaborative among the conference hosts, authors, reviewers, and participants. For the implementation, Windows Workflow Foundation (WF) is used to bind all of the process elements to form the workflow of the conference management system. REST is used to invoke the registered services via HTTP Protocol as illustrated in Fig. 3.
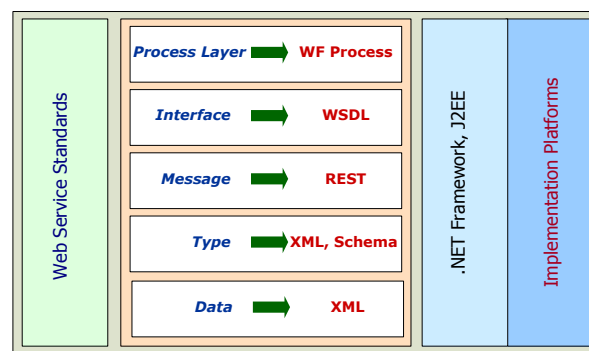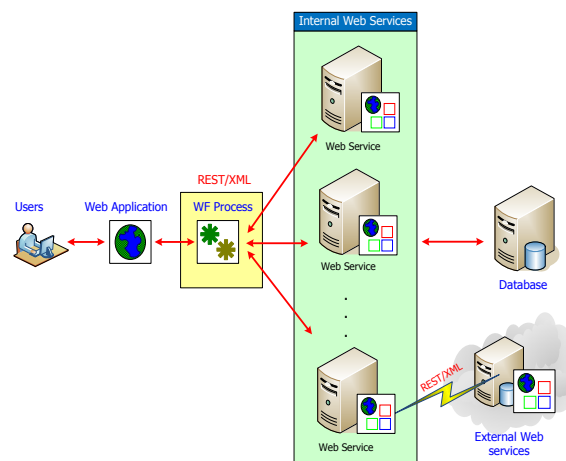

Fig. 2 Architecture Layers

Fig. 3 Connection of Application Services with WF Process

The functionalities of the implemented system mainly cover the activities of configuration for system administrator, paper submission, paper reviews, participant's registration, hotel booking (can invoke the conference hotel's provided service), and support services (e.g. emails, log file). Log files and user log-in information are stored for the audit purpose. The system also provides the service to invoke legacy services in the past.

The process elements of all activities are designed and developed as an application service. Examples of process elements are "Register for Paper Submission", "Send Paper", "Keep Track of Paper Submission Progress", "Paper Assignment", "Reviewer Submission", "Send Notification E-mails", etc. A cluster of related process elements is an

activity.

Five user groups are defined in the system consisting of:
- *Administrator*: configure and maintain the system.
- **Program Committee Chair**: assign papers to Reviewers and make decisions on paper acceptance.
- *Authors*: submit papers and register for presentation.
- *Reviewers*: review papers and submit the results.
- *General Users*: access to conference information and register for participation.

The overview of the system architecture associated with the defined user groups is depicted in Fig. 4.
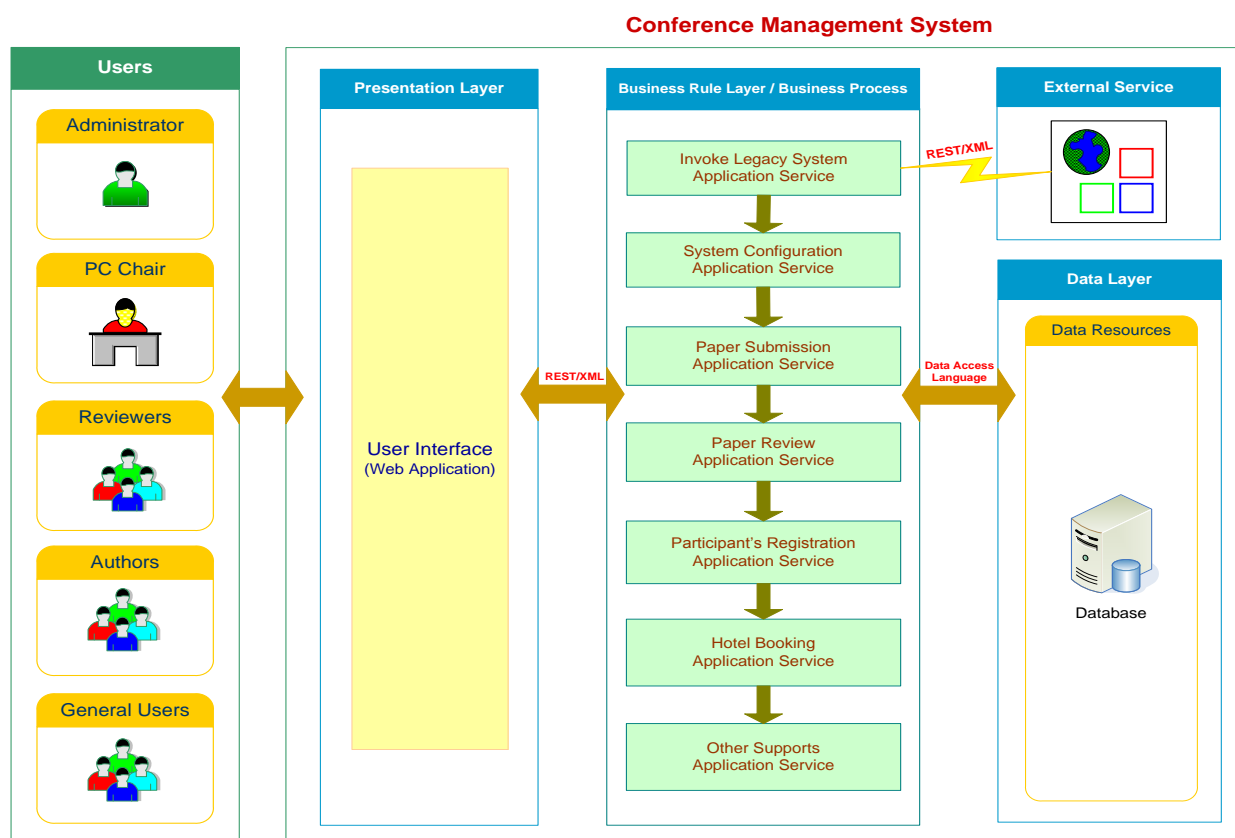

Fig. 4 Architecture of the Conference Management System

V. IMPLEMENTATION

According to the system architecture that has been analyzed and designed in the previous part III, the conference management system was developed using WF as a tool for managing business processes, while REST is used for calling web services via HTTP protocol and receiving data in desired XML format. Several interfaces were developed to interact with the users.

The home page of the conference web presented in Fig. 5 provides the links to various screens of service applications designed for different groups of users:
- **Administrator**: Configure Initial-Context of System, for example, Configure call for papers page; Edit and Update configuration; Configure user authorization; and Configure sending notification e-mails.

- **Program Committee Chair (PC Chair)**: Configure, Edit and Update for paper submission, call for papers; Assign papers to reviewers (Fig. 6); other configurations such as Sending notification e-mails.
- **Reviewers**: Resister and Edit own profile; Bid for papers (with reviewer's expertise or interest) (Fig. 7); Download assigned papers from PC-Chair; and Submit the review's results to the system.
- **Authors**: Register and Edit own profile; Submit papers (more than one upload is allowed before due date) (Fig. 8); and Track the progress.
- **General Users**: Browse the conference general information; Register for conference participation; View the information of accommodation and transportation; Hotel booking.

Fig. 5 Conference Management System Home Page

Fig. 6 Paper Assignment to Reviewers Page

Fig. 7 Reviewer Bid for Papers Page

VI. CONCLUSION

Today commercial conference management software is costly, some are platform or language dependent, or hardly reusable. This paper has designed and developed a conference management system using web-service technology and business process modeling. The approach accommodates the workflow characteristics of conference management systems with stateful processes. The conference tasks are implemented as an application service, and thus the system is platform or language independent. Due to its simplicity, the REST protocol is used for calling web services via HTTP and receiving data in XML format. The approach also eases the modification of the conference process flow due to the concept of business process modeling, which is implemented with Windows Workflow Foundation in this work to manage the processes. The implemented system facilitates the major tasks necessary for organizing a conference namely: call for papers, paper submission, article reviews, author notification, and conference registration. The user groups of the system consist of hosts, authors, reviewers, and general participants.

Fig. 8 Edit Information of Author's Own Paper Page

REFERENCES

[1] Richard van de Stadt. "Cyberchair : A web-based groupware application to facilitate the paper reviewing process" [Online]. Available from: http://www.cyberchair.org/wbgafprp.pdf [2008, December 18].

[2] "PCS: Precision Conference Solutions" [Online]. Available from: http://precisionconference.com [2008, December 18].

[3] Steve Vinoski, "REST eye for the SOA guy", IEEE Internet Computing, Vol. 11, No. 1. (2007), pp. 82-84.

[4] Mark Endrei, Jenny Ang, Ali Arsanjani, Sook Chua, Philippe Comte, Pal Krogdahl, Min Luo, and Tony Newling, " Patterns: Service-Oriented Architecture and Web Services", 1st ed., IBM, pp. 31 – 37.

[5] "SOAP Version 1.2 Part 1: Messaging Framework" [Online]. Available from: http://www.w3.org/TR/soap12-part1 [2008, December 18].

[6] Leonard Richardson, and Sam Ruby, "RESTful Web Services", 1st ed., O'Reilly, 2007.

[7] Ben Scofield, "Practical REST on Rails 2 Projects", 1st ed., Apress, 2008, pp. 1 – 17.

[8] Matt Cumberlidge, "Business Process Management with JBoss jBPM", 1st ed., Packt Publishing, 2007, pp. 5 – 16.

[9] Kenn Scribner, "Microsoft Windows Workflow Foundation", 1st ed., Microsoft Press, 2007.