Technical Architecture of Enabling Body of Knowledge System for Effective Learning and Information Dissemination

Liang-Jie Zhang[1], Jia Zhang[2]
[1]Kingdee International Software Group CO., Ltd, China
[2]Carnegie Mellon University – Silicon Valley, USA
zhanglj@ieee.org; jia.zhang@sv.cmu.edu

ABSTRACT:

In the current era of knowledge explosion, many fields are witnessing a tremendous amount of research and practice reported on a regular basis. How to help people effectively and efficiently study state-of-the-art knowledge in a specific field has become an urgent task yet highly challenging. On top of the Internet as an unstructured knowledge base, this paper reports the design and development of a Body of Knowledge portal (BoK), which can be used as a novel learning environment. Leveraging the key technologies of services computing (Web 2.0, Web services and Service-Oriented Architecture), a BoK provides a uniform gateway for researchers and practitioners to seamlessly study and organize knowledge from heterogeneous data sources. A service-oriented knowledge delivery mechanism is key to a BoK centered on configurable delivery protocols. As services computing having evolved as the foundational discipline supporting modern services industry, we used the field as an example to illustrate the technical architecture that enables the establishment of BoK in services computing. Information dissemination of BoK on mobile delivery platform is explored as well. Performance data analysis is also reported on the BoK infrastructure.

KEY WORDS:
Learning Management System; Knowledge Sharing; Web Services

INTRODUCTION

The Internet environment has triggered many disciplines and fields to emerge and evolve rapidly. For example, services computing has evolved in the last decade as the foundational discipline to study how to leverage IT and Internet computing technology to help people perform business services more efficiently and effectively (Zhang, Zhang et al. 2007). These new fields gain significant momentum and a tremendous amount of research and practice has been reported on a regular basis. How to deliver such exploded state-of-the-art knowledge to the community in a timely manner has become an urgent task yet highly challenging.

This paper reports our design and development of a Body of Knowledge portal (BoK). One major goal of our BoK is to provide an effective online learning platform, as an instance of a *constructivist learning environment* (Wilson 1996), for facilitating community members in studying the latest knowledge and literature in a specific field. According to Wilson (Wilson 1996), an effective learning environment should be "a place where people can draw upon resources to make sense out of things and construct meaningful solutions to problems." Toward this ultimate aim, this paper reports our design and development of the BoK portal 2.0. Our contributions can be summarized in the following three main perspectives: (1) knowledge delivery protocol, (2) system and process, and (3) time and place.

From the knowledge delivery protocol perspective, we designed effective knowledge processing and transmission models. On top of the Internet as an unstructured knowledge base, our BoK provides a uniform gateway that organizes knowledge from heterogeneous data sources. Based on knowledge ontology and a hierarchical resource model, we built a uniform message-based knowledge delivery protocol to facilitate service resource sharing.

From the system and process perspective, we leverage the key enabling techniques in the field of services computing, i.e., Web 2.0, Web services, and service-oriented architecture (SOA), to design a novel architectural model of the environment, featured by its comprising components, interaction patterns between components, and control flow of the system. The BoK portal allows researchers and practitioners to seamlessly review and construct knowledge from heterogeneous data sources such as IEEE Computer Society digital library, IEEE Services Computing community database, related international conference websites and paper review systems, and community-based audio and video conference presentation resources.

From the time and place perspectives, we created an open learning environment where users can learn from anywhere and at anytime. The portal delivers the learning services via mobile channel (e.g., iPhone) in addition to desktop channel. Our iPhone-based service also serves as a learning game.

As an example, our BoK 2.0 was developed to serve for the field of services computing. In this project, we also studied and analyzed the performance and scalability of our BoK infrastructure, from the perspective of its capability of handling increasing complexity when integrating various data resources and delivering services through various channels. It should be noted that our technology to build the BoK learning environment for services computing is not limited to the field of services computing; rather, it can be applied to build knowledge bases and learning environments for other disciplines as well.

The remainder of the paper is organized as follows. After examing the motivation background of the project, we introduce the enabling technologies as the basis for building the BoK portal, present our knowledge delivery mechanisms, and present system process of the BoK design. Then, we introduce our iPhone-based knowledge delivery service as an example of mobile channel service delivery. Afterwards, we present our performance data analysis, and discuss related work. Finally, we draw conclusions.

MOTIVATING CASE AND CHALLENGES

Modern services industry poses a significant IT challenge as how to achieve higher reusability, scalability, interoperability, and cost efficiency. To address this issue, services computing has recently evolved as the foundational discipline to study how to leverage IT and computing technology to help people perform business services more efficiently and effectively (Zhang, Zhang et al. 2007). This is the reason why in the recent decade, services computing has gained significant momentum from both industry and academia. However, applying services computing technology to modernize the traditional services industry is now facing significant challenges. One major issue is the lack of skilled practitioners to meet the increasing and urgent demands from the market.

To address this urgent need, the IEEE Computer Society Technical Committee on Services Computing (TC-SVC) has formed an initiative in 2006 to explore how to promptly deliver the new knowledge to its community members. Throughout this process, we found that learning and teaching services computing poses significant new challenges. In contrast to other disciplines, this field exhibits three noteworthy features: (1) it is new; (2) it evolves rapidly; and (3) it has an urgent need for an enormous amount of skilled professionals.

First, because the field is rather new, it takes time to form a mature curriculum program. Second, because the knowledge of services computing evolves quickly, it is particularly important to constantly and promptly deliver to learners (not only students but also practitioners) the latest technical accomplishments, usually scattered in various publication formats and channels such as journal articles, conference papers, and a variety of styles of presentations (e.g., audio, video, slides, and so on). Third, because the society requires a large amount of skillful professionals at present, we have to find effective ways to train existing practitioners, in addition to training students. These training programs may go beyond the traditional form of continuing education training.

In summary, there is a special need for services computing, a new knowledge area, to transfer evolving knowledge to enormous learners in an effective and efficient way. To serve this need, TC-SVC has been constructing a Body of Knowledge portal for the field of services computing (www.servicescomputing.tv, SC-BoK or BoK for short in this paper). It is community-based, aiming to serve global services computing community members (through free membership) including researchers, educators, students, and practitioners.

ENABLING TECHNOLOGIES

Interestingly, we found that the technologies in the field of services computing itself are suitable for building the BoK. Especially, we adopt three key techniques: Service Oriented Architecture (SOA) [1-3], Web services, and Web 2.0 [4]. In addition, we explore the knowledge delivery on the mobile platform.

Service Oriented Architecture

Services computing covers the science and technology of effectively leveraging computing and information technology to model, create, operate, and manage business services. Its core technology suite centers on SOA, an architectural model that guides the design and development of reusable and interoperable business solutions. On one hand, SOA implies that a solution system shall reuse existing services as components. On the other hand, SOA implies that a solution system shall be delivered as a service and capable of being reused in other solutions. We adopt SOA for the overall portal design, to ensure a reusable and flexible system architecture while leveraging existing software systems (e.g., IEEE Computer Society digital library).

Web services

To date, Web services technology is the best enabling technology to realize SOA. In general, a Web service refers to a programmable Web component with a standard interface, and can be universally accessible using standard network protocols (Zhang, Zhang et al. 2007). Its fundamental supporting standards are Web Services Description Language (WSDL), Simple Object Access Protocol (SOAP), and Universal Description, Discovery, and Integration (UDDI). We adopt the Web services technology to enable the reusability of our portal and ensure the interoperability between comprising components in our portal. Every component is designed and developed as a Web service, so that it can be easily reused and interoperate with other components.

Web 2.0

Web X.0 was coined in the recent years to represent the concept of “Web eXpandoon,” implying the ultimate target technology set rather than certain disaggregated technologies [3][8]. Web 1.0 represents the current Internet environment where people can publish information and anyone who has access to the Internet has access to the published content. Web 2.0 represents the emerging Internet platform where people can create new services using existing published content and services. The upcoming Web 3.0 represents the next-generation Internet where Web content can be semantically understandable to machines. In other words, Web X.0 refers to the open Internet environment from business point of view. In this work, we adopt the latest Web 2.0 technology to construct the BoK environment, to enable configurable and personizable user interfaces and community-based collaborations.

iPhone Development Platform

Apple’s iPhone technology has opened a new era for Internet information publication and retrieval. *iPhone* lays a standard for an all-in-one integrated mobile phone and makes gaming more accessible to non-gamers in a decent way. Since 2008, Apple has opened iPhone development platform (Apple 2008) and allows people to publish iPhone applications and games to the online Apple Store. The programming language of the iPhone development platform is Objective-C.

Observing that numerous people now use mobile phones to browse the Internet, we studied how to deliver the BoK as services through multiple channels including traditional desktops and mobile phones. As a pilot study, we explored services delivery on the iPhone platform. As an example, we developed an iPhone application as a learning game.

SERVICE-ORIENTED KNOWLEDGE DELIVERY

In this section, we will discuss three key considerations regarding service-oriented knowledge delivery: a knowledge ontology model, an extensible resource model, and a knowledge delivery protocol.

Knowledge Ontology

One of our main concerns is what knowledge should be delivered. Instead of delivering course materials, the major goal of BoK is to accumulate and convey the latest technologies in the field of services computing to the community members. Such knowledge is embedded in recently published papers in the field.

IEEE Computer Society sponsors six international conferences in the field of services computing: IEEE International Conference on Web Services (ICWS), IEEE International Conference on Services Computing (SCC), IEEE International Conference on Cloud Computing (CLOUD), IEEE International Conference on Mobile Services (MS), IEEE International Congress on Big Data (BigData), and IEEE Congress on Services (SERVICES). The first five conferences are regular research conferences focusing on various topics in the field. The last one focuses on the applications of services computing techniques in vertical industry sectors, while also carrying a set of workshops whose themes fall into the field. IEEE Computer Society digital library accumulates all published papers in these conferences as well as target journals in the field, i.e., IEEE Transactions on Services Computing (TSC). It serves as a great resource for researchers, students, and practitioners to study the latest advances in the state of the art in the field.

However, most of these papers are at least eight pages long. A reader may not always intend to read the full paper, until she finds the content closely related to her interests. In addition, the digital library is not free of access. Although many research universities and some companies purchase group access, still many users do not have access to the digital library. Therefore, people may not want to buy an article until she wants to read its details. Unfortunately, paper abstract may not provide sufficient information for people to make a decision.

From our experience of attending and organizing conferences, we found that a powerpoint presentation from paper authors is a good way for people to understand the concepts and techniques of a paper to a good level. Conference participants read paper titles and abstracts and identify papers that cause their attentions. Then they go to corresponding paper presentations. Afterwards, they may read the papers if they intend to know all details. We perceive a phenomenon that, although all conference participants have access to the original papers in the proceedings, they ask for presentation slides if they do not manage to listen to some presentations in person.

Thus, since 2007, the TC-SVC has decided to record all paper presentations at the conferences, per authors' agreements, and publish them online after the conferences. For each paper, we thus gather its powerpoint presentation slides and transform it in PDF format, and gather its audio (and video) presentations and transform them into flash files.

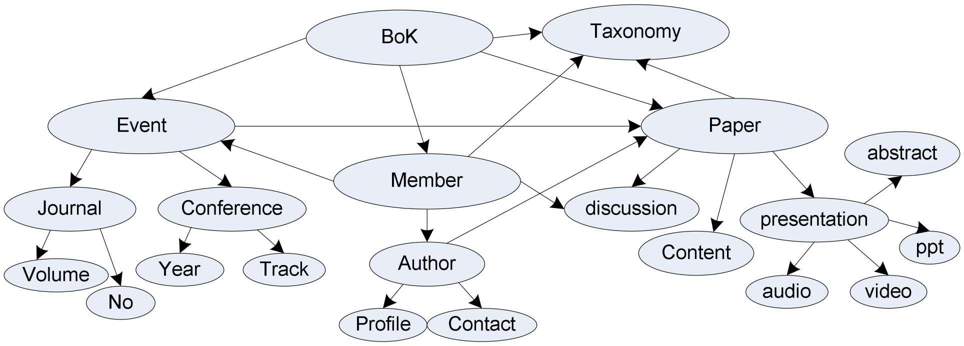


Fig. 1 Knowledge ontology at BoK.

We have built a knowledge base at BoK centered on papers, as shown in Fig. 1. BoK maintains a collection of papers and their associated materials. Each paper contains three types of information: content, presentations, and discussions. Paper content is a link pointing to its paper position at the IEEE CS digital library. Presentation materials may include paper abstract, powerpoint slides, audio presentation, and video presentation. Discussions refer to community members' comments on the paper.

Papers are categorized according to the corresponding events such as journals or conferences. Journal papers are further organized by the volumes and numbers; conference papers are further organized by years and tracks.

BoK maintains a list of its community members. A member may be an author of a paper, a reviewer, or any person who intends to study the technologies in the field. Each member possesses a personal profile and contact information. A member may participate in community events and comment on papers.

As shown in Fig. 1, to provide a landscape view of the field, TC-SVC creates the Taxonomy (IEEE 2008) for BoK to categorize its comprising papers. Each member may identify one or more directions in the Taxonomy as her interested fields.

Resource Model

As shown in Fig. 1, BoK accumulates various types of materials including paper abstracts and paper presentations (in powperpoint, audio and video files). To seamlessly incorporate these heterogeneous resources and provide a uniform view to users, we have developed a service-oriented resource (or service resource in short) model, where any resource data is encapsulated into a service resource that can be used to provide value-added services. Fig. 2 illustrates the internal structure of a service resource in three layers.

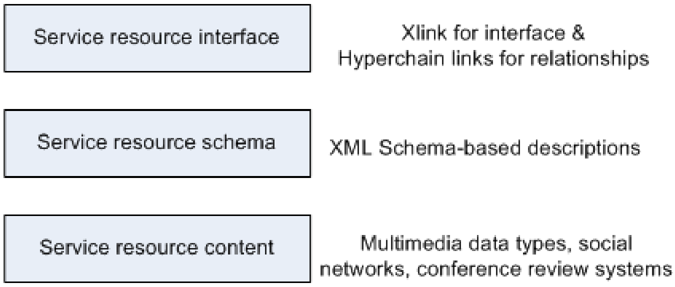

Service resource interface
Xlink for interface &
Hyperchain links for relationships
Service resource schema
XML Schema-based descriptions
Service resource content
Multimedia data types, social
networks, conference review systems

Fig. 2 Service resource model.

The bottom layer represents the actual content of a service resource instance. It may be exhibited in various data types including text (e.g., Web page), Word and PDF document, audio, and video. In addition to multimedia resources, other resources may also be leveraged such as social networks and databases of conference review systems.

The structures of service resource instances are captured in corresponding service resources schema, as shown in the second layer. For example, a schema description may define that three resource instances represent different facets of a paper, i.e., paper abstract, powerpoint presentation, and video presentation. Such XML schema-based descriptions will help users interpret the resources and extract semantic meanings of the resources. For example, one may find that two papers come from the same research group based on their metadata descriptions.

The top layer defines the interfaces of service resources and the relationships between them. Interface of a service resource is defined in the format of the XML Linking Language (XLink) [10], which allows users to create internal and external links within XML documents, and associate metadata with the links. Relationship between service resources is defined using HyperChain links [5], which represents a directed association relationship between service resources.

Our service resource model hides the complexity and heterogeneity of the actual contents of resources. Its hierarchical structure also offers flexibility: content replacement will become transparent to users as long as the related links remain. Furthermore, this model allows resources to be exposed in different links to provide different views to serve different interests.

Knowledge Delivery Protocol

To facilitate service resource sharing, we have established a uniform message-based knowledge delivery protocol.

Knowledge delivery parties

Fig. 3 illustrates the major participating entities in BoK regarding knowledge delivery. The services computing community comprises a number of community members each carrying a unique identity. It provides a collection of services aiming to facilitate knowledge sharing for community members, maintains a set of service resources, and organizes various events (e.g., conferences).

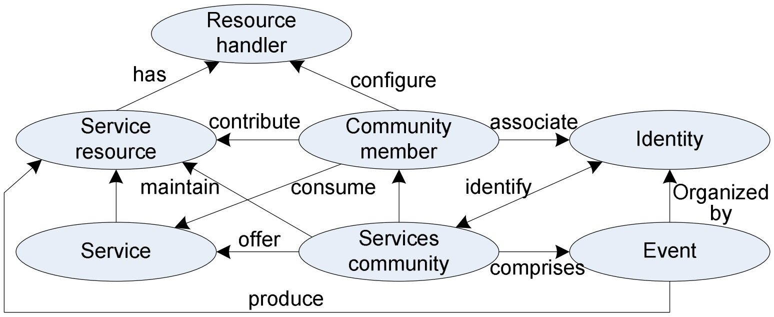


Fig. 3 BoK major participating parties.

Fig. 4 illustrates a common scenario between the above comprising entities. Each service is built on top of a set of heterogeneous service resources. Each service resource is contributed by one or more community members. They can also be reconfigured through resource handlers, which are special objects that can be tuned by members to take operations on the resource. Each event is organized by one or more community members serving in predefined roles.

An event, such as a conference, will produce a collection of resources such as papers, audio podcast, video files, presentations, and so on. Service resources may be categorized by events, or be contributed by individual community members on an *ad hoc* basis.

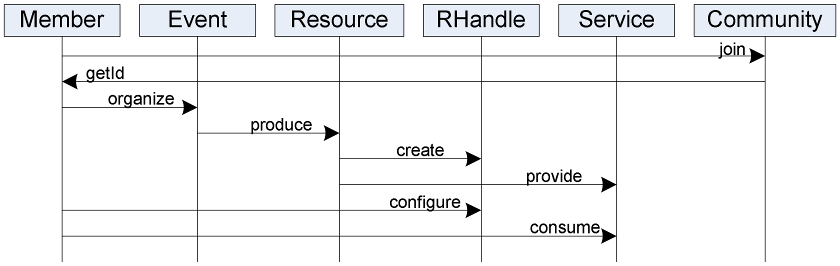


Fig. 4 A scenario between entities.

Table 1. Knowledge delivery primitives.

Type	Primitive Name	
Request	Request for Resource Meta (RFRM)	RFRM_ACK
	Request for Resource Type (RFRT)	RFRT_ACK
	Add Resource Type (ART)	ART_ACK
	Update Resource Type (URT)	URT_ACK
	Annotate for Resource (AFR)	AFR_ACK
	Message for Member (MFM)	MFM_ACK
	Join for Community (JFC)	JFC_ACK
Response	Return with Resource Meta (RWRM)	RWRM_ACK
	Return with Resource Type (RWRT)	RWRT_ACK
	Accept/Reject Resource Type (A/R-RT)	ARRT_ACK
	Accept/Reject Annotation (A/R-A)	ARA_ACK
	Accept/Reject Message (A/R-M)	ARM_ACK
	Accept/Reject Joining (A/R-J)	ARJM_ACK

Knowledge delivery primitives

Based on our study of knowledge delivery patterns, we have identified a set of semi-structured message primitives, as summarized in Table 1. The primitives are divided into two categories: request primitives and response primitives. Each primitive is a pair of a request and an associated instant acknowledgement.

Seven request primitives are identified: (1) Request for Resource Meta (RFRM) that asks for metadata of a resource (paper); (2) Request for Resource Type (RFRT) that asks for a specific data resource of a paper; (3) Add Resource Type (ART) that intends to add a data resource to a paper; (4) Update Resource Type (URT) that intends to update a data resource of a paper; (5) Annotate for Resource (AFR) that intends to annotate a paper; (6) Message for Member (MFM) that intends to contact a community member; and (7) Join for Community (JFC) that intends to join the community.

Six response primitives are identified: (1) Return with Resource Meta (RWRM) that replies to members with the meta data of a paper; (2) Return with Resource Type (RWRT) that replies to members with a specific data resource of a paper; (3) Accept or Reject Resource Type (A/R-RT) that either accepts or rejects a submitted paper resource (e.g., when a user tries to add an audio presentation); (4) Accept or Reject Annotation (A/R-A) that either accepts or rejects a submitted annotation to a paper; (5) Accept or Reject Message (A/R-M) that either accepts or rejects a submitted message to a member; and (6) Accept or Reject Joining (A/R-J) that either accepts or rejects a member's request to join the community.

In addition to be used individually, these knowledge delivery primitives can be used as building blocks to construct comprehensive knowledge sharing and delivery patterns.

Knowledge delivery constructs

Based on the identified knowledge delivery primitives, we have established a concept of knowledge delivery construct. A construct comprises a list of configured primitives to accomplish a specific knowledge delivery pattern. Through different combinations over the set of primitives, different knowledge delivery patterns can be realized.

```
<process name="RFRmicroflow"
   targetNamespace="urn:KDConstructs"
   xmlns:tns="urn:samples:KDConstructs"
   xmlns="http://BoK/constructs/">

   <sequence>
      <invoke name="invokeRFRM"
         partner="Member" portType="tns:RFRMoriginatorPT"
         operation="rfrm" outputVariable="RMRM">
      </invoke>

      <invoke name="returnRWRM"
         partner="BoK" portType="tns:RWRMreceiverPT"
         operation="rwrm" outputVariable="RWRM">
      </invoke>

      <invoke name="invokeRFRT"
         partner="Member" portType="tns:RFRToriginatorPT"
         operation="rfrt " outputVariable="RFRT">
      </invoke>

      <invoke name="returnRWRT"
         partner="BoK" portType="tns:RWRTreceiverPT"
         operation="rwrt" outputVariable="RWRT">
      </invoke>

   </sequence>
</process>
```

Fig. 5 Service-oriented RFR construct.

As shown in Fig. 5, a knowledge delivery construct is represented by a sequence of primitives, in a well-defined order. We can thus model a construct as a mini-workflow, which can be described using the Business Process Execution Language (BPEL) (OASIS 2002) based on Pi-calculus. Fig. 5 shows a section of the BPEL definition of the RFR construct that realizes the request for all knowledge related to a paper. For simplicity, we skipped the section defining messages (messages), partners (various modules in the BoK portal), and variables (messages), and links (expressing synchronization dependencies).

As shown in Fig. 5, each primitive is wrapped as a service. Each primitive is realized by a service call, associated with the corresponding messages. Once represented by BPEL, multiple constructs may combine to form a comprehensive scenario. Such a service-oriented model enables platform-neutral and language-neutral collaboration.

The RFR construct comprises four primitives: RFRM, RWRM, RFRT, and RWRT. It first requests the metadata of a paper. Upon receipt of metadata, it selects a presentation type (e.g., powerpoint presentation) and requests the actual content (e.g., slides). Currently, we have constructed a number of constructs, enabling knowledge retrieval, contribution, commenting, and communication.

Various knowledge delivery activities are accomplished through message passing. All messages are based on XML to enable universal knowledge delivery in BoK. To enable validation and analysis over messages, we have adopted the XML Schema to uniform the format of knowledge delivery messages.

Messages are divided into request messages and response messages. Each message contains one or more primitives that form a transaction, meaning that they form an atomic unit of work in the BoK space. Each transaction aims to serve for a knowledge delivery goal. A message may also contain optional data such as comments and requirements.

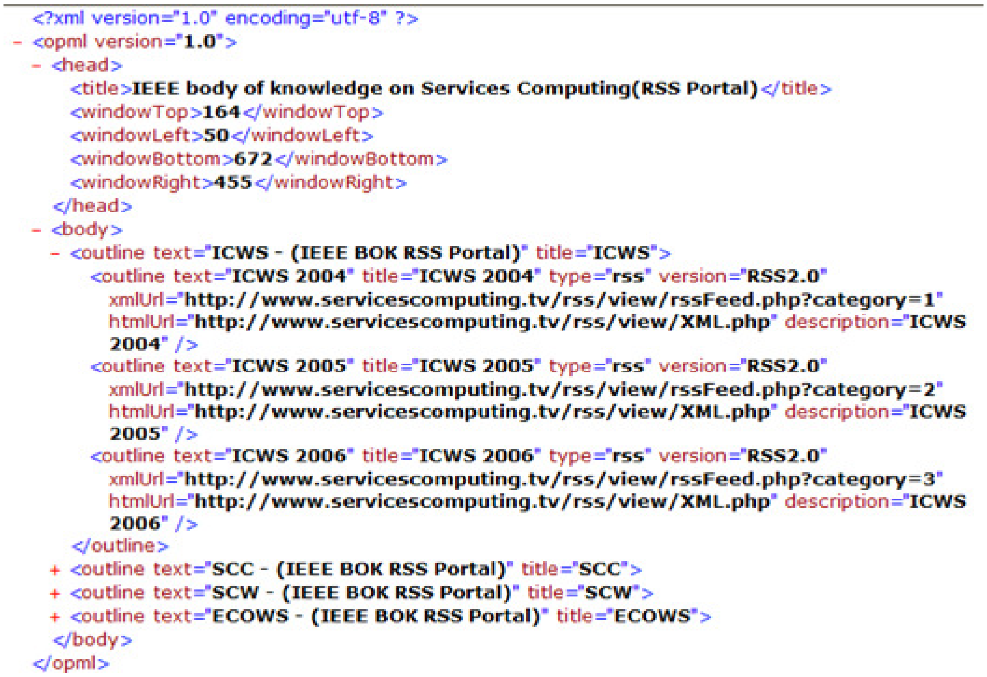
Knowledge delivery format

To enable a simple, self-documenting, and extensible format to represent the wide variety of knowledge data in the BoK, we adopted the Outline Processor Markup Language (OPML) recommended by W3C. OPML is an XML-based format that enables exchange of portable outline-structured information between applications. Users maintain knowledge outliners. Internet services on the knowledge can be browsed in Web browsers and controlled through an outliner.

As shown in Fig. 6, an outline is a tree where each node contains a set of named attributes with values. Outliner lists hierarchical structure of the knowledge base. For example, BoK comprises papers from ICWS conferences categorized by years. Users may use Really Simple Syndication (RSS) to specify personalized subscription to the knowledge base. We are especially interested in the aggregators in OPML that maintain the structure of the document, so they can tell when a set of comments are part of a sub-outline, and should be grouped together. It's also possible to add attributes to individual items.

SYSTEM AND PROCESS

In this section, we report how we designed and implemented the BoK using service-oriented



```
  <?xml version="1.0" encoding="utf-8" ?>
- <opml version="1.0">
  - <head>
      <title>IEEE body of knowledge on Services Computing(RSS Portal)</title>
      <windowTop>164</windowTop>
      <windowLeft>50</windowLeft>
      <windowBottom>672</windowBottom>
      <windowRight>455</windowRight>
    </head>
  - <body>
    - <outline text="ICWS - (IEEE BOK RSS Portal)" title="ICWS">
        <outline text="ICWS 2004" title="ICWS 2004" type="rss" version="RSS2.0"
          xmlUrl="http://www.servicescomputing.tv/rss/view/rssFeed.php?category=1"
          htmlUrl="http://www.servicescomputing.tv/rss/view/XML.php" description="ICWS
          2004" />
        <outline text="ICWS 2005" title="ICWS 2005" type="rss" version="RSS2.0"
          xmlUrl="http://www.servicescomputing.tv/rss/view/rssFeed.php?category=2"
          htmlUrl="http://www.servicescomputing.tv/rss/view/XML.php" description="ICWS
          2005" />
        <outline text="ICWS 2006" title="ICWS 2006" type="rss" version="RSS2.0"
          xmlUrl="http://www.servicescomputing.tv/rss/view/rssFeed.php?category=3"
          htmlUrl="http://www.servicescomputing.tv/rss/view/XML.php" description="ICWS
          2006" />
      </outline>
    + <outline text="SCC - (IEEE BOK RSS Portal)" title="SCC">
    + <outline text="SCW - (IEEE BOK RSS Portal)" title="SCW">
    + <outline text="ECOWS - (IEEE BOK RSS Portal)" title="ECOWS">
    </body>
  </opml>
```

Fig. 6 OPML description.

architecture, to enhance its reusability and extensibility.

Single Access Point

Various resources are scattered in various places, e.g., IEEE Computer Society digital library and conference websites. Users must log onto individual systems to search for interested information. In addition, data from different sources are separated from each other. Users cannot obtain a comprehensive view of interested resources across multiple data sources.

To tackle these issues, BoK provides a single access point that allows users to access information from multiple data sources. More importantly, our BoK synergistically leverages these data sources and provides new value-added services. For example, one can configure personal preferences so that only filtered information will be presented. A single sign-on technique (Pashalidis and Mitchell 2003) is adopted, so that a community member can log into the BoK using her IEEE identity and view integrated information from various sources including IEEE Computer Society digital library.

BoK provides self-serving services to community members. Users can configure and re-configure personal preferences. Fig. 7 shows the service-oriented delivery model of BoK. The hyperchain-enabled common service resources description framework is centered on our service resource model. Service resources are exposed in service hyperchains and grouped into categories based on various purposes, and serve as the underlying knowledge base supporting various BoK services. As shown in Fig. 7, resources are accessed through the standard Web 2.0 technologies (e.g., RSS/wiki) and SOA infrastructure (e.g., message-oriented service integration facilitated by XML schema).

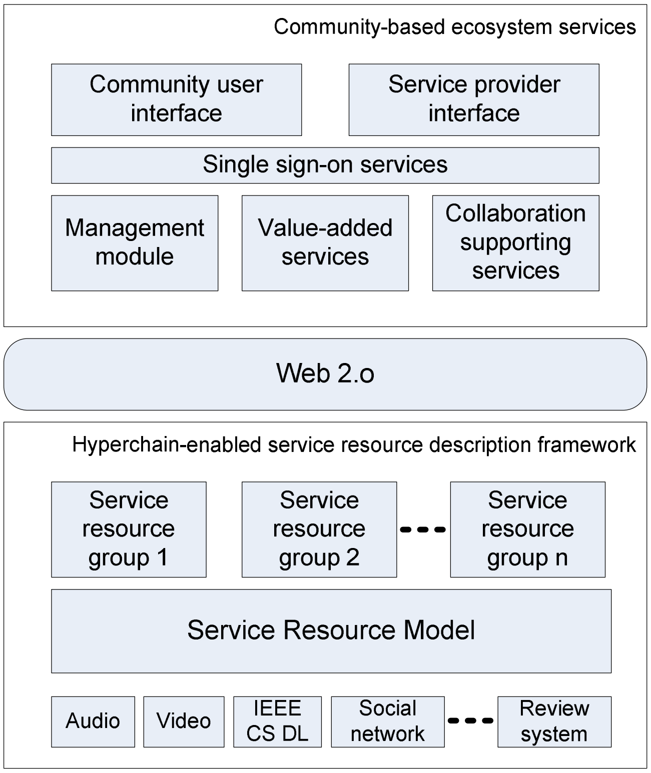


Fig. 7 Services-oriented delivery model of BoK.

BoK represents a portal offering a set of community-based ecosystem services on top of the hyperchain-enabled service resources. A services delivery platform aims to establish a common platform to deliver services efficiently and reuse service resource assets. Services can be categorized in three classes: 1) value-added services including integrated information query, personalizable data filtering, and community newsletter services, 2) management module comprising a collection of administrative functions managing life cycles of service resources such as personal profile management and research tracking facility, and 3) collaboration supporting services that help community members to communicate and interact and share knowledge.

BoK provides two interfaces, as shown in Fig. 7. One interface is for community users who access the portal through its customizable web interface. The other interface is prepared for other service providers, meaning that BoK provides a programmable Web service interface so that other applications can leverage the BoK services to create more value-added services.

System Architecture

To ensure reusability and extensibility, we have adopted the concept of SOA to design and develop the internal structure of BoK. As Fig. 8 illustrates, BoK is built on top of three major data sources: IEEE Computer Society digital library (papers), servicescomputing.tv (paper presentations), and confhub.com (paper review system). Internally, three databases are maintained: a knowledge base caching retrieved paper information for higher system performance, a user database storing all community user information, and a usage database capturing and storing the interrelationships between knowledge, user, and their usages.

BoK is maintained by eight modules: a content manager controls all papers and associated presentations and annotations; a user manager controls all community member information; a confhub manager control access to the paper review system; a sc.tv manager controls access to the multimedia presentation source; a query manager control generation and retrieval of knowledge data; a security manager ensures single sign-on ability; a monitor manager ensures the health of the BoK system; and an analysis manager conducts statistical investigation over the system.

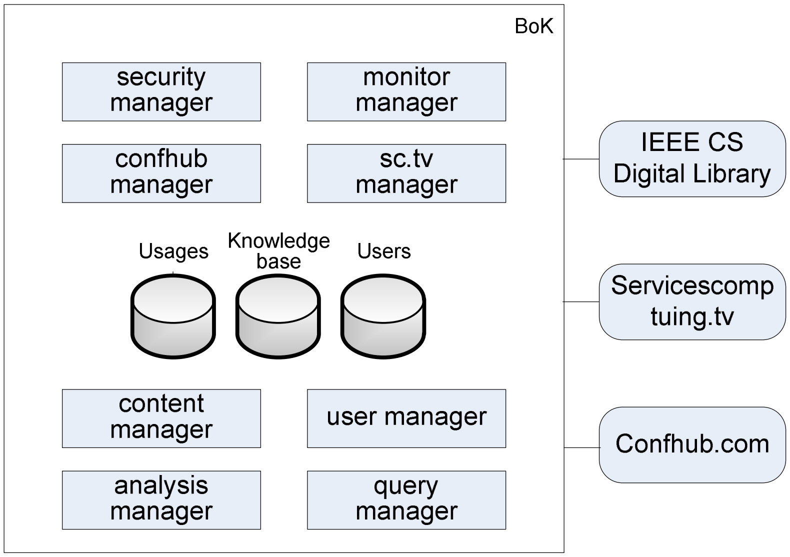


Fig. 8 Architecture of BoK.

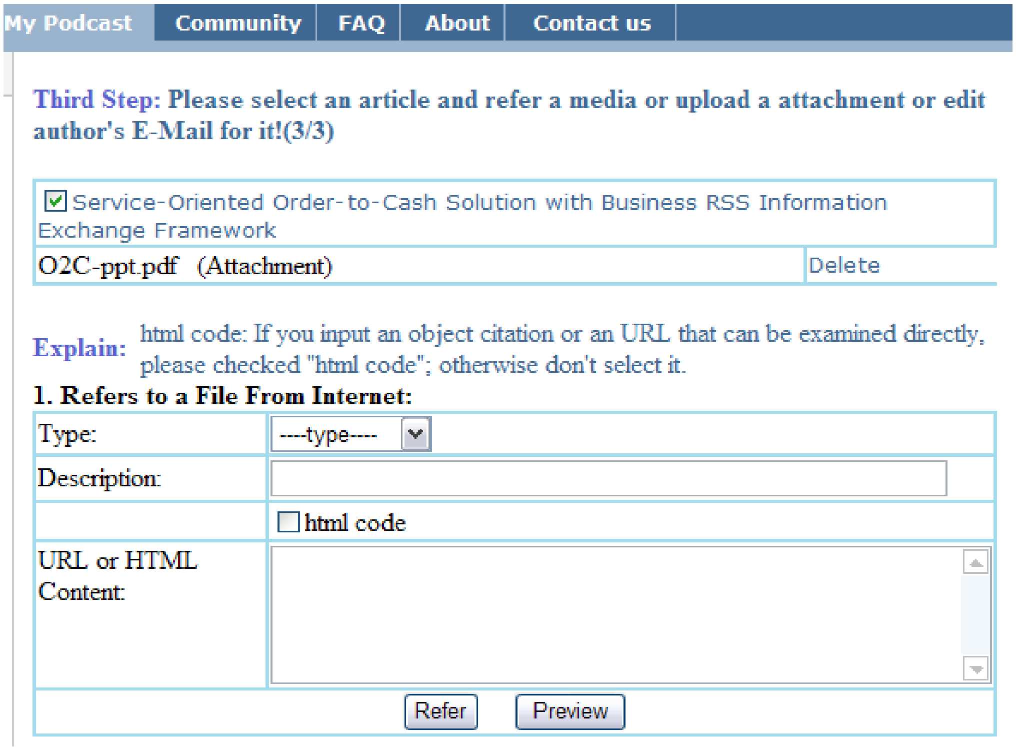
System Implementation

We started to build the BoK since 2006. It is implemented in Hypertext Preprocessor (PHP) language. At the time, we had a couple of options: Java (Niemeyer and Knudsen 2005), PHP (Nixon 2009), and C# (Troelsen 2010). All three languages are widely used and general-purpose programming languages to develop dynamic web applications. In contrast to Java and C# that require compiling with any code change, PHP code is embedded into HTML source documents and interpreted by a Web server with a PHP processor module at real time. Therefore, dynamic code change can be integrated into a running system easily. Moreover, PHP is available as a processor for most modern web servers and is provided by most Web hosting servers free. Therefore, we decided to adopt PHP as the programming language in this project.

TC-SVC purchased the domain name "servicescomptuing.tv" as the hosting domain for BoK. Fig. 9 shows a screen shot of the BoK system. The services computing community provides free community membership. Any user may register and log onto the BoK. However, since IEEE requires membership for accessing its digital library, paper-associated presentations are accessible for IEEE members using their IEEE login information.

BoK provides three major categories of services: *browse service*, *myPodcast service*, and *community service*. Browse services allow users to find interested knowledge in the BoK. Search criteria include *title*, *author*, *keywords*, and so on. Search results will be presented according to user customization.

As shown in Fig. 9, an author may select one of his/her published paper and add presentation files to it. The author will specify the type of the presentation (e.g., ppt presentation) and some descriptions. BoK may examine the file to be uploaded before it is associated to the paper.



Third Step: Please select an article and refer a media or upload a attachment or edit author's E-Mail for it!(3/3)

☑ Service-Oriented Order-to-Cash Solution with Business RSS Information Exchange Framework

O2C-ppt.pdf (Attachment)	Delete

Explain: html code: If you input an object citation or an URL that can be examined directly, please checked "html code"; otherwise don't select it.

1. Refers to a File From Internet:

Type:	----type----
Description:	
	□ html code
URL or HTML Content:	

Fig. 9. Screen shot of BoK.

After a published paper is registered in the BoK and authors key in major data such as introduction to the paper and the work, community members may contribute knowledge such as editing (as people do in a Wiki), making comments, rating/reviewing, making recommendations, and sharing the contents with other people by sending the links.

MOBILE DELIVERY CHANNEL

Our original design of the BoK portal delivered the services through the desktop channel, meaning that the design of the intended content display was oriented to desktop browsers alone. In the recent years, with the advancement of the ubiquitous computing technologies, people are increasingly using mobile devices to browse the Internet using PDAs and cell phones. Many popular Web portals have provided mobile access channels including Yahoo (Yahoo.com) and Google (Google.com). This phenomenon encouraged us to refactor our design and target on delivering the BoK services in multiple channels including mobile channels. Our goal is to provide a better service for the services computing community members to learn the state-of-the-art knowledge in the field. For example, they may browse the BoK for the latest publication or course content on a train or bus to and from work.

Content Adaptation

The challenge is how to quickly adapt the existing BoK portal to provide mobile channel delivery. The portal content has to be transformed into appropriate formats to match the capabilities of receiving mobile devices. For example, most PC screens have a resolution of at least 1024x768 pixels, while mobile phone screens only 176x208 pixels or less.

We studied various content adaptation techniques aiming to optimize the transcoding process supporting different devices simultaneously. Two options are available. First, a number of multi-channel-oriented content publishing tools have been developed, such as Oracle application server wireless (Oracle), WebSphere transcoding publisher (IBM), and Sun (now Oracle) Java system portal server mobile access (Sun). One common feature of these tools is that, raw content is stored in an XML format, and the proper display data is dynamically generated before delivered to the receiving device. The disadvantage of this option is that, these tools require all content be developed using the tools. Completely redesigning and regenerating the entire BoK portal and content is neither effective nor efficient. Furthermore, the BoK portal has its unique feature as many of the content are most appropriate to desktop devices for the best result, including multimedia presentations.

The second option is to build a dedicated content adaptation service to dynamically transform the original content into an appropriate format based on the type of the receiving device. The advantage of this option is apparent, since the existing portal can remain untouched. Therefore, we decided to go with the second option.

Our previous research created a novel Web service that automatically detects the semantic relationships between HTML content components and reorganizes the page layout of the components for a better display on mobile devices. The underlying technique is based on a concept of unit-of-information (UOI), which refers to an atomic semantic unit comprising a set of segments and media objects that have to be presented on the same screen (Yang, Zhang et al. 2007).

The BoK application raises new challenges to the adaptation of BoK content. As discussed in Section 3, SC BoK contains a rich collection of knowledge in various multimedia types including text, image, audio, video, and flash. In addition to the resolution difference, different receiving mobile devices may possess different network bandwidth, processing power, and storage space. These features have to be taken into consideration to reach a higher performance. For example, if a receiving mobile device does not support video and flash display, there is no need to deliver the comprising video and flash components of a document.

Our previous research (Zhang, Zhang et al. 2005) extends Composite Capability/Preference Profiles (CC/PP) (W3C 2001) standard to enable client-side transformation descriptions for specific receiving devices. In this project, we further extend the CC/PP standard to support server-side content adaptation descriptions for various types of receiving devices. For each BoK composite document, its BoK profile includes two parts: one is the declaration of the comprising multimedia contents; the other is the declaration of the supported content adaptation services. Each type of receiving device is defined as a separate resource item.

Fig. 10 shows a segment of a profile of a paper's presentation materials, describing specified content adaptation guidance. The declaration of the "Paper1" materials contains two parts. The first part describes its comprising multimedia presentations: its abstract, images, powerpoint presentation in the PDF format, and its author's oral presentation recorded as a flash. The second part defines its acceptable receiving devices such as a WAP phone. The profile also specifies the content adaptation methods for delivering the Paper1's materials to the WAP phone. As shown, a WAP phone only accepts text information. Note that the normal HTML code cannot be directly retrieved on this WAP phone. A server-side service using the stylesheet named HTML2WML.xsl will be invoked to transform the abstract of the paper from its HTML format to the target WML format.

As shown in Fig. 10, our server-side content profiles are described in Resource Definition Framework (RDF), a *de facto* standard description language widely used in semantic Web for making the Web content more meaningful. At current stage, based on the BoK content on site, we

```
…
<ccpp:component>
   <rdf:Description rdf:about=”CAGuidance”>
      <rdf:type rdf:resource=”Paper1-PresentationPackage”/>
      <Text>Paper1-Abstract.html</Text>
      <Image>Paper1-Image.jpg</Image>
      <PPT>Paper1.pdf</PPT>
      <Flash>Paper1.camrec</Flash>
   </rdf:Description>
</ccpp:component>
…
<ccpp:component>
   <rdf:Description rdf:about=”Services”>
      <rdf:type rdf:resource=”Supported-WML-Service”/>
      <rdf:li>HTML</rdf:li>
      <rdf:transform>HTML2WML.xsl</rdf:transform>
   </rdf:Description>
</ccpp:component>
```

Fig. 10 A segment of server-side content adaptation descriptions.

created a template. Each BoK content document, such as Paper1, instantiates the template and creates a profile as shown in Fig. 10. To ensure performance, the profiles are created using the pattern of lazy instantiation, meaning that they are dynamically created upon request for browsing and/or downloading. Once a profile is created, our BoK portal will keep a copy and associate it with the corresponding document (e.g., Paper1). This strategy further enhances the performance of services delivery. After a content document (e.g., Paper1) is viewed by a user, its profile does not have to be regenerated.

iPhone Channel

In this section, we will report our exploration and experiment of a complementary service delivery channel for services computing learning materials. As an example, we developed an iPhone gaming application as an additional channel and mechanism (game) to promote content study through BoK.

In spite of the promising mobility, cell phone devices used to have inherent shortcomings as a tool for Web browsing, such as their small screens and limited bandwidths. However, the emergence of Apple's iPhone has greatly changed that picture. The emergence of the 3G cell networks (Wikipedia) has greatly improved bandwidth. Meanwhile, Apple, through the effective use of a touch screen (including the zoom feature) combined with the larger screen made possible by the controversial decision to forgo a keyboard, has succeeded in finally bringing the Web to the cell phone.

When this feasibility becomes a reality, in April 2008, Apple published its iPhone Software Development Kits (SDK) (Apple 2008) and opened its development platform (iPhone OS). As a result, developers can develop applications and publish them in the online App Store from which users can download and run on iPhones. To date, App Store has approved and published more than 140,000 applications.

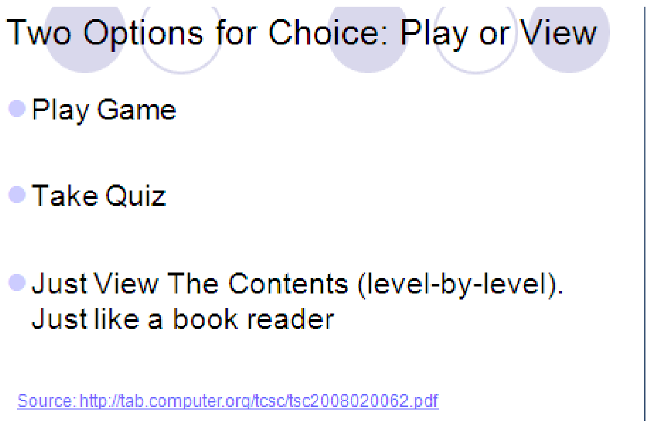
Therefore, we decided to investigate the possibility of exploiting the iPhone platform as an additional channel to deliver BoK learning materials. Furthermore, we explore to deliver iPhone-based service in the format of learning game.

iPhone Application

As a pilot study, we designed our navigation application based on SVC Curriculum (14 BoK knowledge areas and sub-areas). The programming language for the iPhone OS platform is Objective-C (Kochan 2009), a reflective, object-oriented programming language that adds a thin layer over C programming language with Smalltalk-style messaging facility.

The learning game follows the standard of the App Store under the category of Educational application. Fig. 11 shows a screen shot that is the front page of the game, delivering our message as the goal of the application: to help community members learn Body of Knowledge in the field of services computing. Links to the services computing community and related journals and conferences are provided.

The application aims to help users get familiar with structure and the relationships within the IEEE Transactions on Services Computing (TSC) Taxonomy, a landscape view of the emerging Services Computing discipline (IEEE 2008). TSC Taxonomy comprises 14 knowledge areas and their subsequent sub-areas. These key knowledge areas form a foundation for the field to guide



Two Options for Choice: Play or View

- Play Game
- Take Quiz
- Just View The Contents (level-by-level). Just like a book reader

Source: http://tab.computer.org/tcsc/tsc2008020062.pdf

Fig. 11. iPhone application front page.

knowledge organizations (Zhang 2008). Journals and conferences also use these knowledge areas to categorize publications in the field. Therefore, it is important for authors, reviewers, editors, and readers to understand and grasp the knowledge area structures and relationships, so that they know how to position their work and find related work in the literature.

At the highest-level structure of the TSC taxonomy, 14 main knowledge areas of the services computing discipline are organized into four categories in a tree-like structure: Category 1 is about Services and Services Systems; Category 2 about Services Technologies; Category 3 about Services Consulting and Delivery; and Category 4 about Services Solutioning and Management. After a user touches the screen and selects a knowledge area, say M2, the application will display the description of the knowledge area.

The application allows users to browse to the 3rd-level knowledge sub-areas and the 4th-level knowledge sub-areas. Users can either select one knowledge area and view its sub-areas if there exist, or click for a longer time to view the description of the knowledge area.

iPhone Gaming Application

As a learning game, version 1.0 supports the following three basic functions: navigation, game, and quiz. As explained in the previous figures, users can statically navigate through the tree structure of the TSC taxonomy to view the content of each knowledge area. Users can switch to the game playing mode. In this mode, once a random node (knowledge area) appears in the iphone screen, users can select from a list of options about its short descriptions. Meanwhile, users can tap it to obtain the text description about the node. If the answer is correct, users can click the “Continue” or “Resume” button to move to the next sub-area level or select other topics at the same level. The application also provides a quiz mode. Quiz questions are designed using the pre-defined tree structure. For example, “Select the knowledge sub-areas for M3. Web Services.” Three choices are provided: at least one answer is correct; the others are dynamically picked from the tree structure. After a set of 10 questions, a score will be provided.

To ensure flexibility and extensibility, we store the TSC taxonomy on the BoK portal in the format of XML. The iPhone application is able to update its cached TSC taxonomy from the BoK portal upon request. As a result, the TSC taxonomy can be updated by the Services Computing community dynamically, and the iPhone application does not have to go through any code change.

Experiments

Since the goal of our BoK portal is to support the global services computing community users, scalability of the portal is critical to its success. We studied the scalability of the BoK portal, by simulating loading testing over various potential user populations. We also studied transactional information including underlying page elements and server responses.

Experimental Settings

We adopted the professional tool, IBM Rational Performance Tester (RPT) 8.1 (IBM), and exploited its automated performance testing capabilities to validate the scalability of our BoK portal. The communication protocol is HTTP.

Particularly, we focus on the following four types of experimental settings to accurate load emulation: modeling user groups, modeling navigation patterns, modeling system resource usage patterns, and fine-grained analysis.

To accurately mirror user population, we utilized the *user group* capability of RPT to specify testing loads by proportional user groups. From our survey, the BoK users can be divided into the following three main categories: (1) authors whose published papers are indexed in the BoK portal, and they proactively go to the site to edit and manage their paper-related documents; (2) users who constantly review the site for related literature work according to the TSC taxonomy; and (3) users who download documents from the site. The proportions among them are 1:6:3. We thus use this ratio to represent our testing user mix.

Individual users adopt proprietary navigation patterns to access the BoK website, we utilized the *data pooling* capability of RPT to vary the test data set used by each emulated virtual user. We used a text channel (i.e., Microsoft Comma Separated Value (CSV) format) to load in pre-defined usage patterns. In this way, we can model, control, and study the impact of a variety of navigation patterns.

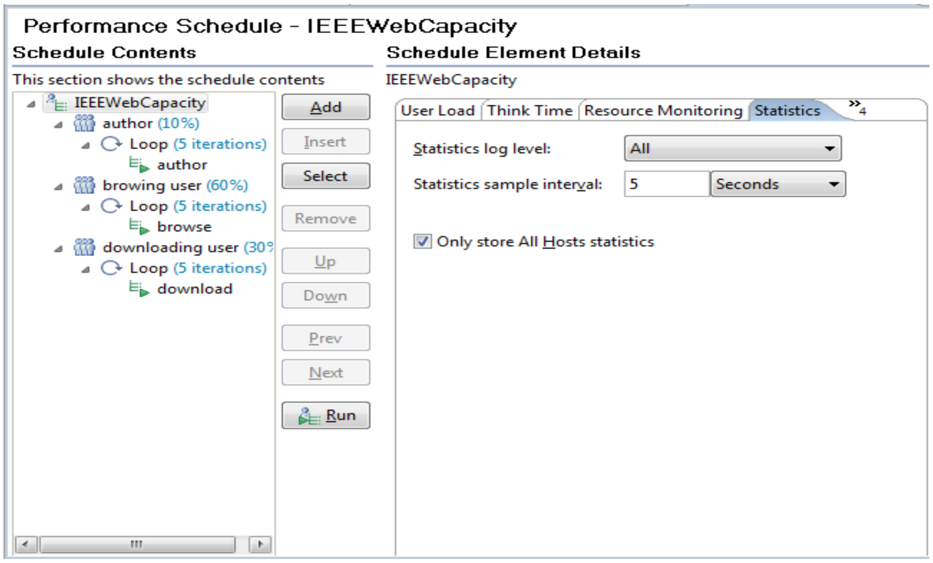
It is critical for us to monitor and investigate system resource consumption patterns of the BoK portal under various load scenarios. We are specially interested in CPU and memory utilization statistics. Integrated with response time and throughout data, these data help us diagnose and predict the portal health under various situations.

To realize a fine-grained testing, we adopted a combination of strategies to design the test cases, including inserting verification points and Java code to validate server response codes and response sizes.

Experimental Results and Analysis

Based on our aforementioned experimental designs, we conducted a collection of experiments to measure the scalability of the BoK portal, under different situations with different numbers of concurrent users that access the BoK portal. We use RPT 8.1 to conduct the experiments and record the test results.

We simulated a set of user populations with 10, 20, 50, 100, 150, 200, 500, and 1000 concurrent users. We will discuss the details of performance schedule for each user population.



Performance Schedule - IEEEWebCapacity

Schedule Contents

This section shows the schedule contents

IEEEWebCapacity
author (10%)
Loop (5 iterations)
author
browing user (60%)
Loop (5 iterations)
browse
downloading user (30%
Loop (5 iterations)
download

Schedule Element Details

IEEEWebCapacity

Statistics log level: All

Statistics sample interval: 5 Seconds

☑ Only store All Hosts statistics

Fig. 12 User population settings.

As shown in Fig. 12, each user population is configured to comprise three user groups: author, browsing user, and downloading user. Their proportions are configured as 1:6:3 based on survey data. Each user group adopts a common navigation pattern. The author group searches for their specific publications, finds the one to edit, and uploads modifications. The browsing user group inputs their interested knowledge areas following the TSC taxonomy, and browses through retrieved documents. The downloading use group inputs a specific document information (e.g., paper title), and downloads a found document in one of the available formats (e.g., abstract in text, presentation in PDF, and presentation in flash). We wrote Java code into the test cases to ensure the correctness of the test results. Below we will discuss some performance analysis diagrams generated from the test log of the experiments.

Fig. 13 shows the page hit rate (left) under a specific user load (right). Here we configured to evaluate three parameters: page attempt rate (the number of page requests sent to the server within a predefined time interval), page hit rate (the number of successful page requests from the server within the same time interval), and total page element hits (the number of successful requests of page elements from the server within the same time interval).

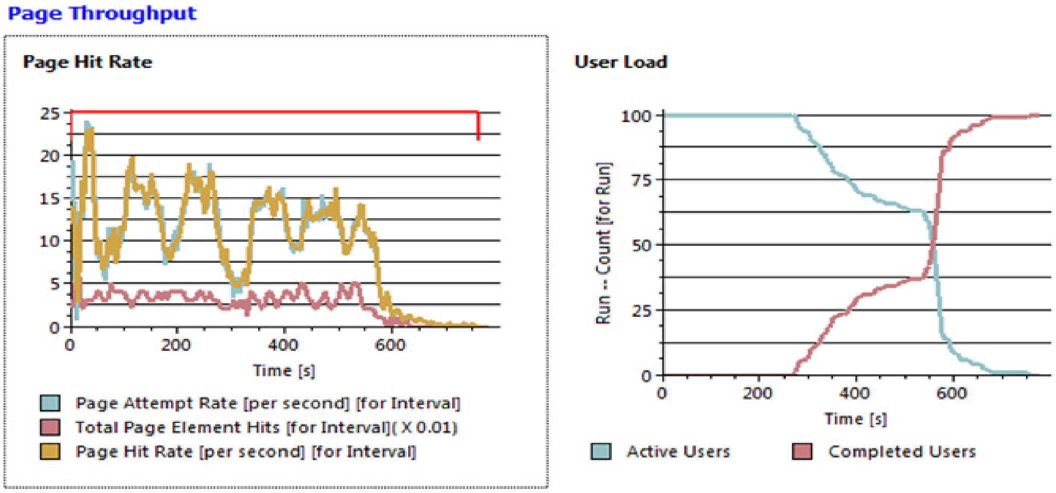


Fig. 13. Page throughput (10 user load).

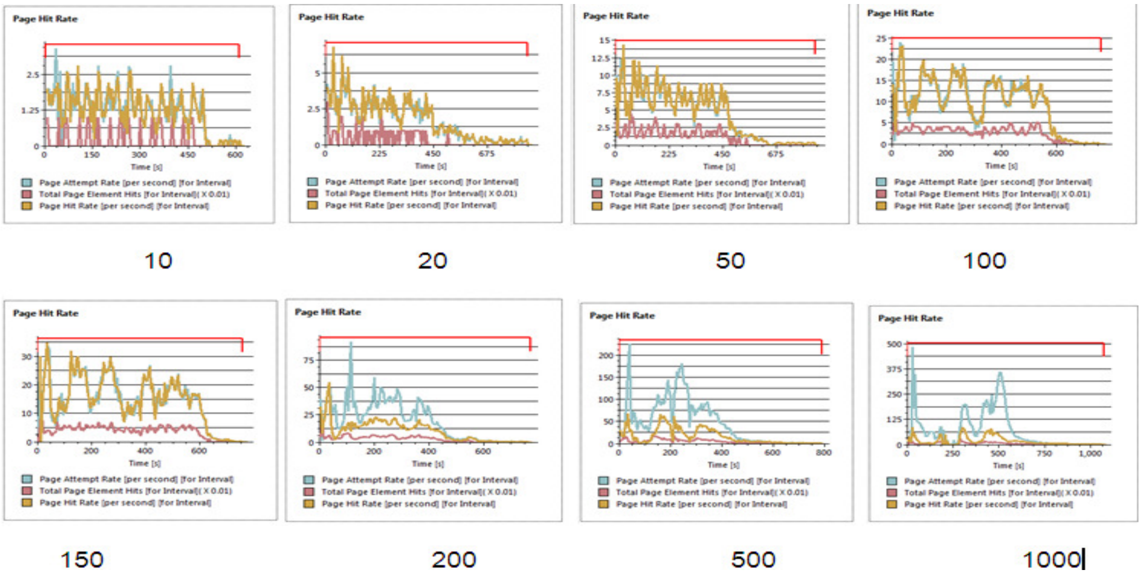
Particularly, Fig. 13 illustrates the page hit rate under 100 user load. The left hand-side diagram records the values of three selected parameters along the time (the current time interval is configured to be 5 seconds): the number of page attempts, the number of successful page hits, and the number of successful page element hits. As shown in Fig. 13, under 100 user load, the page hit rate is constantly high.

The right hand-side diagram reveals two phenomena. First, the testing system does initiate (simulate) all 100 users simultaneously, i.e., all 100 users become active from the beginning. Second, the IEEE portal is able to server incoming requests in a concurrent manner.
To avoid coincidence, we repeated our test runs for 5 iterations and compared the multiple test runs. Fig. 13 shows the average performance testing results (page hit rate) for 5 test runs for 100 user population.

Fig. 14 shows the page hit rate under different user loads: 10, 20, 50, 100, 150, 200, 500, and 1000. We can see that, as the workload increases, the page hit rate and page element hit rate drop. Especially, the page hit rates drops significantly when the user load reaches 200.

Fig. 15 shows the 10 highest average page response time under different user loads (10, 20, 50, 100, 150, 200, 500, and 1000). Since the 10 pages that require long response time almost remain the same, it is safe for us to conclude that the scalability of the portal is satisfactory under different user populations (up to 1000 concurrent users). Among all of our tests, we found that the page requiring the longest response time is the 2007 Web services conference homepage that comprises a significant number of graphical elements (as shown in the lower right corner of Fig. 19).

Further, we recorded all pages that were attempted in every user population in every iteration. We identified the top 10 pages that were mostly attempted, using the following formula. The page usage in each user population of an iteration is scaled up to that in population of a base value of 1000.



10 20 50 100

150 200 500 1000

Fig. 14. Page throughput (10-1000 user load).

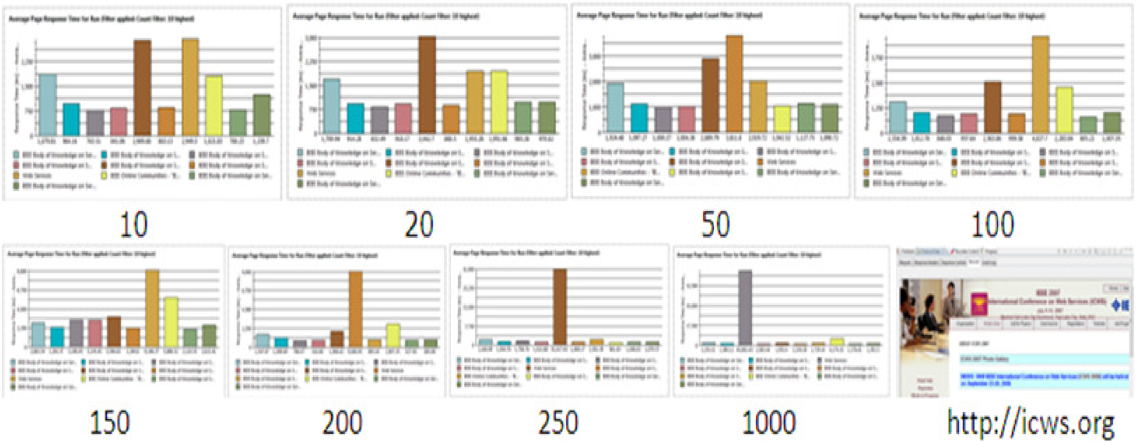


Fig. 15 Highest average page response time (10-1000 user load).

$$PU(p, up, i) = PU(p, up, i) / i * base$$

where: $PU(p, up, i)$ represents the page usage of the page p in the user population up in iteration i.

We then normalize the page usage of page p in the entire test set as follows:

$$PU(p) = \frac{\sum_{I}\sum_{UP} PU(p, up, i)}{\sum_{I}\sum_{UP}\sum_{P} PU(p, up, i)}$$

where P={all pages included in the BoK portal}, I={1-5}, $UP = \{10, 20, 50, 100, 200, 250, 500, 1000\}$.

We did the experiments on two machines. The table below shows the response time (in ms) of the top 10 pages that were mostly attempted in all of our experiments. Each response time is below 1.1 seconds. It proves that our BoK portal has good scalability.

M	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	606.9	627.1	726.9	765.9	780.3	786.5	816.1	827.9	885.1	893.4
2	804.2	806.2	916.1	917.6	962.8	1015.3	1032.3	1064.3	1119.7	1150.3

RELATED WORK

The Web has evolved from a place of sharing information to a networked community of sharing knowledge. The Web 2.0 techniques are suitable for forming community-based service resources [13]. The key ideas of Web 2.0 are user contribution, data centric, and easy access of information through collaboration. Blog, wiki, and mashup are examples of Web 2.0 realizations. On the other side, SOA represents the trend of services integration through a loosely coupled architecture through common standard-based middleware [2-3]. Web 2.0 and SOA are converging to further enable services computing [14].

Our previous conference version (Zhang and Cai 2007) describes the overall system framework of BoK portal 1.0. In contrast, this paper reports our design and development of BoK portal 2.0, focusing on our efforts of exploiting the Web 2.0 technology to turn the portal into a new way of effectively and efficiently delivering state-of-the-art learning materials to the community members in a variety of channels including the mobile channel. We also report our study and experiments on performance and scalability of the system.

The HyperChain technique in designing the resource model in BoK derives from our earlier work on multimedia HyperChain in HotMedia [5] that supports linkages to multimedia resources. In contrast, our work in this research builds a service-oriented resource model that leverages the XLinks technique to expose HyperChain links to hide internal complexity of various resources including audio and video resources.

Sayah and Zhang (Sayah and Zhang 2005) present their annotated business hyperchain technology enabling on-demand business collaboration with the Web services technology. They propose a set of business collaboration primitives serving business collaboration scenarios. In contrast, our knowledge delivery primitives serve as building blocks for constructing configurable knowledge delivery protocols.

Our previous research (Zhang, Zhang et al. 2005) extends the Composite Capability/Preference Profiles (CC/PP) (W3C 2001) standard to enable client-side profiles supporting content adaptation for a specific receiving device. In this project, we continue our research by further extending the CC/PP standard to enable server-side profile descriptions for supporting content adaptation for various types of receiving devices.

CONCLUSIONS

In this paper, we have presented our design and development of the technical architecture of the Body of Knowledge (BoK) in the field of services computing. Facing the knowledge explosion in the field, BoK serves as an organized knowledge base and provides a novel learning environment for community members to study the up-to-date knowledge in the field. We have proved the effectiveness of our design of the knowledge delivery mechanisms and the architectural framework of the portal.

We plan to continue our research in the following several directions. First, we plan to enrich the services of the BoK and allow provide a personalizable knowledge learning portal. Second, we plan to build a social network based on our BoK portal to further facilitate the knowledge sharing and delivery in the community. Third, we will study how to deliver the comprehensive BoK knowledge through the mobile channel.

REFERENCES

Apple. (2008). iOS Dev Center - Apple Developer. from http://developer.apple.com/devcenter/ios/program/.

Google.com. Google. from http://www.google.com/.

IBM. Rational Performance Tester. from http://www-01.ibm.com/software/awdtools/tester/performance/.

IBM. WebSphere Transcoding Publisher. from http://www-306.ibm.com/software/pervasive/transcoding_publisher/.

IEEE, C. S. (2008). TSC Taxonomy. from http://www.computer.org/portal/pages/transactions/tsc/mc/tsc_taxonomy.html.

Kochan, S. G. (2009). *Programming in Objective-C 2.0 (2nd Edition)*. Upper Saddle River, NJ: Pearson Education, Inc.

Niemeyer, P. and Knudsen, J. (2005). *Learning Java (3 edition)*: O'Reilly Media.
Nixon, R. (2009). *Learning PHP, MySQL, and JavaScript*: O'Reilly Media.
OASIS. (2002). OASIS Web Services Business Process Execution Language (WSBPEL) TC. from https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel.
Oracle. Oracle Application Server Wireless. from http://www.oracle.com/technology/tech/wireless/index.html.
Pashalidis, A. and Mitchell, C. J. (2003). Single Sign-On Using Trusted Platforms. *Lecture Notes in Computer Science*, *2727/2003*.
Sayah, J.Y. and Zhang, L.-J. (2005). On-Demand Business Collaboration Enablement with Services Computing. *Decision Support Systems*, *40(1)*, 107-127.
Sun. Java System Portal Server Mobile Access. from http://www.sun.com/software/products/portal_srvr/index.xml.
Troelsen, A. (2010). *Pro C# 2010 and the .NET 4 Platform (Fifth Edition)*: Apress.
W3C. (2001, Mar. 15). W3C Composite Capability/Preference Profile, W3C Working Draft.
Wikipedia. 3G Networks. from http://en.wikipedia.org/wiki/3G.
Wilson, B. G. (1996). *Constructivist Learning Environments: Case Studies in Instructional Design*. Englewood Cliffs, NJ: Educational Technology Publications, Inc.
Yahoo.com. Yahoo. from http://www.yahoo.com.
Yang, S. J. H., Zhang, J., Chen, R. C. S. and Shao, N. W. Y. (2007). A Unit of Information-Based Content Adaptation Method for Improving Web Content Accessibility in the Mobile Internet. *Electronics and Telecommunications Research Institute (ETRI) Journal*, *29(6)*, 794-807.
Zhang, J., Zhang, L.-J., Quek, F. and Chung, J.-Y. (2005). A Service-Oriented Multimedia Componentization Model. *International Journal of Web Services Research (JWSR)*, *2(1)*, 54-76.
Zhang, L.-J. (2008). EIC Editorial: Introduction to the Body of Knowledge Areas of Services Computing. *IEEE Transactions on Services Computing (TSC)*, *1(2)*, 62-74.
Zhang, L.-J. and Cai, H. (2007). Service HyperChain Architecture of Web X.o and A Case Study. In *Proceedings of IEEE International Conference on Web Services (ICWS 2007)*, Salt Lake City, UT, 968-975.
Zhang, L.-J., Zhang, J. and Cai, H. (2007). *Services Computing*: Springer.

ABOUT THE AUTHORS

Liang-Jie (LJ) Zhang, Ph.D., is Senior Vice President, Chief Scientist, & Director of Research at Kingdee International Software Group Company Limited, and director of The Open Group. Prior to joining Kingdee, he was a Research Staff Member at IBM Thomas J. Watson Research Center. Dr. Zhang has published more than 140 technical papers in journals, book chapters, and conference proceedings. He has 40 granted patents and more than 20 pending patent applications. Dr. Zhang received his Ph.D. on Pattern Recognition and Intelligent Control from Tsinghua University in 1996. He chaired the IEEE Computer Society's Technical Committee on Services Computing from 2003 to 2011. He also chaired the Services Computing Professional Interest Community at IBM Research from 2004 to 2006. Dr. Zhang has served as the Editor-in-Chief of the International Journal of Web Services Research since 2003 and is the founding Editor-in-Chief of IEEE Transactions on Services Computing. He was elected as an IEEE Fellow in 2011, and in the same year won the Technical Achievement Award "for pioneering contributions to Application Design Techniques in Services Computing" from IEEE Computer Society. Dr. Zhang also chaired the 2012 IEEE 1st International Conference on Services Economics (SE 2012) and the 2009 IEEE International Conference on Cloud Computing (CLOUD 2009). He can be reached at zhanglj@ieee.org.

Jia Zhang, Ph.D., is an Associate Professor at Carnegie Mellon University – Silicon Valley. Her current research interests center on Services Computing, with a focus on collaborative scientific workflow, service-oriented architecture, semantic service discovery, and Internet of Things. She has co-authored one textbook titled “Services Computing” and published over 120 journal articles, book chapters, and conference papers. Zhang is an associate editor of IEEE Transactions on Services Computing (TSC) and of International Journal on Web Services Research (JWSR). She is a member of the IEEE and can be reached at jia.zhang@sv.cmu.edu.