Research Article
User-Centric Design for Mathematical Web Services

Adlin Sheeba1,2 and Chandrasekar Arumugam3

1 Department of Computer Applications, St. Joseph’s College of Engineering, Chennai 600119, India
2 Research Scholar, Centre for Information Technology and Engineering, Manonmaniam Sundaranar University, Tirunelveli 627012, India
3 Department of Computer Science and Engineering, St. Joseph’s College of Engineering, Chennai 600119, India

Correspondence should be addressed to Adlin Sheeba; adlinsheeba@yahoo.com

Received 30 October 2013; Revised 13 January 2014; Accepted 31 January 2014; Published 6 March 2014

Copyright © 2014 A. Sheeba and C. Arumugam. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A web service is a programmatically available application logic exposed over the internet and it has attracted much attention in recent years with the rapid development of e-commerce. Very few web services exist in the field of mathematics. The aim of this paper is to seamlessly provide user-centric mathematical web services to the service requester. In particular, this paper focuses on mathematical web services for prepositional logic and set theory which comes under discrete mathematics. A sophisticated user interface with virtual keyboard is created for accessing web services. Experimental results show that the web services and the created user interface are efficient and practical.

1. Introduction

Traditionally, the web has been conceived as a distributed source of information. The emergence of the web service technology permitted to extend it to a distributed source of functionality. Thus, web services connect computers and devices with each other using the internet to exchange and combine data in new ways. Many web services exist for various domains such as travel and e-commerce. Only few web services are available for computing mathematical functions such as calculator and so on. But to large extent, software is available for computing mathematical data.

Although software available for computing functions in mathematics has produced promising results that are certainly useful, it is not very much suitable for user-centric internet environment. First, software has to be installed in the system and users must be aware of the built-in functions available. Second, those built-in functions are rarely available as web services. Third, user interfaces with mathematical symbols are limited in use. To address these issues, in this paper, mathematical web services with learning material and user interface are created that will be more useful for the service requester to use and compute the desired services.

This work aims to provide MAThematical Web Services (MATWS) for Prepositional logic (MATWS-P) and Set theory (MATWS-S). Beyond the web services, our approach provides sophisticated user interface with virtual keyboard to assist the service requester.

The contribution of the work reported here is summarized as follows.

(1) A framework is proposed for the creation of MATWS-P, MATWS-S, and user interface.

(2) Various experiments have been done to measure the performance of web services and usage of user interface. Initial results show that the proposed strategy is efficient and practical.

The rest of the paper is organized as follows: Section 2 presents a survey on existing system, while Section 3 provides a conceptual framework for our approach. Sections 4 and 5 describe the creation of MATWS-P and MATWS-S, respectively. Section 6 presents the creation of user interface with virtual keyboard. In Section 7 results are discussed and Section 8 gives some concluding remarks.

2. Related Work

Boyatt and Joy [1] developed learning materials for mathematical logic that includes set theory and propositional logic.
Only study material is provided with some basic examples and web services are not provided. The ontology in [2] includes conceptual foundations for algebra and measurement theory. It is designed explicitly for knowledge sharing purposes.

Various software tools are available for computing mathematical functions. AEPLT [3] is an automated tool that has a user friendly interface and aids in the calculation of the propositional logics of compound propositions of conjunction, disjunction, conditional, and biconditional. Satoulsoule [4] is a publicly available tool which displays a formula editor that allows us to enter a set of formulae. The formulae are the traditional formulae of propositional logic, plus indexed conjunction and disjunction which can either be hand-written in a Lisp-like syntax or introduced in a sort of syntax-directed editor. XploRe Quantlet Client [5] provides Console User Interface and Graphical User Interface to access the statistical computing environment XploRe Quantlet Server via internet with its broad range of available statistical methods.

Many commercial software packages are available for computing functions in propositional logic and set theory. MATLAB [6] is a high-level language which has basic built-in mathematical functions and functions for tautology and normal forms are not available. Mathematica [7] is a software program used in many scientific, engineering, and mathematical fields for computing. Groups, Algorithms, and Programming (GAP) [8] provides a programming language and a library of thousands of functions for computational discrete algebra, with particular emphasis on Computational Group Theory. REDUCE [9] provides a full programming language implemented in Standard Lisp. It is an interactive system for general algebraic computations which also supports arbitrary precision numerical approximation and interfaces to gnuplot to provide graphics. SPSS Statistics [10] is a software package used for statistical analysis. It provides Descriptive statistics, Bivariate statistics, Prediction for numerical outcomes, and Prediction for identifying groups. With a vast range of mathematical software covering different, often overlapping areas, it is quite impractical to maintain local versions of all the packages a particular user might find useful. Although many of the packages are available without license fee, others have a significant cost that cannot be justified for only occasional use for some unique functionality. Also, the complicated nature of the software seems to be difficult for a user to access their services. In this paper, we aim at providing web services for propositional logic and set theory with a user interface so that the services can be accessed via internet easily.

3. Proposed Web Service Architecture

A web service can be defined in a simple way as a service located at some point in the Internet that can be accessed through a standard protocol. Web services technology is based on a set of standard protocols such as Universal Description, Discovery and Integration (UDDI), Simple Object Access Protocol (SOAP), and Web Services Description Language (WSDL). WSDL provides a description of the way to connect and communicate with a particular web service by describing its functionality using an XML language.

With all these components, anybody can use the service offered by any web service available on the Internet, and it is not necessary to take into account the programming language in which the service originally is defined. The architecture in the proposed work is composed of three main parts, namely, the service provider, service registry, and the service requester as shown in Figure 1.

The MATWS framework typically involves the following key tasks.

Service Provider. The service provider provides mathematical web services and client interface for accessing those services.

Service Requester. The service requester or service consumer accesses the web services of service provider through the client interface.

Service Registry. The service registry contains mathematical web services and other services provided by different service providers and helps service requester to access the web services of the service provider.

Client Interface. It is a graphical user interface with virtual keyboard through which service consumers access their required services.

Mathematical Web Services. They are web services for propositional logic and set theory.

Many MATWS have been created for evaluating the expressions in MATWS-P and MATWS-S. The service provider hosts a web site for mathematical web services so that any requester can access MATWS-P and MATWS-S through the client interface. Also, WSDL files are made available in the service registry so that client can discover and access web services through WSDL interface. The web services can also be accessed from web service tool by providing proper input. The results are displayed in a step by step manner so that even the beginners can understand the concept easily. A sample set of created web service operations is shown in Table 1.
Table 1: Sample web service operations.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>WS1:</td>
<td>Web Service: Principle Conjunctive Normal Form</td>
<td>Operation:</td>
<td>PCNF</td>
<td></td>
</tr>
<tr>
<td>Input:</td>
<td>expression</td>
<td>Data type:</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>Output:</td>
<td>lit</td>
<td>Data type:</td>
<td>literal</td>
<td></td>
</tr>
<tr>
<td>WS2:</td>
<td>Web Service: Symmetric Difference</td>
<td>Operation:</td>
<td>SymmetricDifference</td>
<td></td>
</tr>
<tr>
<td>Input:</td>
<td>operandVal</td>
<td>Data type:</td>
<td>array of string</td>
<td></td>
</tr>
<tr>
<td></td>
<td>operandName</td>
<td>Data type:</td>
<td>array of string</td>
<td></td>
</tr>
<tr>
<td></td>
<td>expression</td>
<td>Data type:</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>Output:</td>
<td>lit</td>
<td>Data type:</td>
<td>literal</td>
<td></td>
</tr>
</tbody>
</table>

4. Web Services for Prepositional Logic

Propositional logic is a system of symbolic logic using symbols [11] which has important applications in computer science such as to verify that computer programs produce the correct output for all possible input values and to show that algorithms always produce the correct results. Hence it is necessary to create web services for prepositional logic concept such as negation, conjunction, disjunction, implication, tautology, equivalence, normal forms, and expression evaluation. The WSDL description of a tautology web service is given in Table 2.

Letters/sequence of letters/numbers is used to denote propositional variables, that is, variables that represent propositions. The truth value of a proposition is true (denoted by $T$) if it is a true proposition; the truth value of a proposition is false (denoted by $F$) otherwise.

Any given logical expression entered using virtual keyboard can be evaluated. The validity of the expression is verified to avoid wrong calculation of results. After verifying the validity of the logical expression, it is converted to postfix expression. During the conversion of logical expression into postfix expression, the operands are identified. The operand may be a letter or a sequence of letters or numbers. A count is made on the number of distinct operands ($N$) and the distinct operand names are stored in an array.

Partial truth table values are framed for identified operands. If there are $N$ different operands, then there are $2^N$ different lines in the truth table and $N \times 2^N$ truth table values. For each column in the table, $2^N/2$ true ($T$) values and $2^N/2$ false ($F$) values are computed to get all combinations. Then table head is created based on operator precedence and the operand values are substituted and evaluated.

User interface with mathematical symbols, identified operands, partial truth table values, table head, and the resultant truth table framed using HTML are shown in Figure 2 for the logical expression $OP_1 \land (OP_2 \rightarrow OP_5) \leftrightarrow OP_3$.

5. Web Services for Set Theory

In computer science, a set is an abstract data structure that can store certain values, without any particular order, and no repeated values and set theory are most commonly utilized in science and mathematics fields as well as in computer and electrical engineering.

In the web-centric environment if the various set theory concepts such as union, intersection, symmetric difference, Cartesian product, and power set are available as web services, it would be useful for service requester to access and use them effectively. Also a virtual keyboard is needed for mathematical symbols which are shown in Figure 3.

The service requester may give any set expression. The $N$ number of distinct operands in the expression is identified and $N$ dynamic text boxes are created to get set elements. The expression is evaluated with operand names and operand values given by client in the dynamically created text box as input and the result is displayed as a literal.
Table 2: WSDL description of a tautology service.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<wsdl:definitions
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/
xmlns:soapenc="http://schemas.xmlsoap.org/wsdl/soap/encoding/
xmlns:xmns="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://tempuri.org/
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
<wsdl:binding name="ServiceSoap" type="tns:ServiceSoap"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
transport="http://schemas.xmlsoap.org/soap/http"/>
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://tempuri.org/
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
<wsdl:types
xmlns:elementFormDefault="qualified"
targetNamespaces="http://tempuri.org/"
xmlns:elementName="Tautology">
<complexType>
<sequence>
<element minOccurs="0" maxOccurs="1"
name="expression" type="s:string" />
</sequence>
</complexType>
</element>
<element name="TautologyResponse">
<complexType>
<sequence>
<element minOccurs="0" maxOccurs="1"
name="TautologyResult" type="s:string" />
</sequence>
</complexType>
</element>
</wsdl:types>
<wsdl:message name="TautologySoapIn">
<wsdl:part name="parameters" element="tns:Tautology" />
</wsdl:message>
<wsdl:message name="TautologySoapOut">
<wsdl:part name="parameters" element="tns:TautologyResponse" />
</wsdl:message>
<wsdl:portType name="ServiceSoap">
<wsdl:operation name="Tautology">
<wsdl:input message="tns:TautologySoapIn" />
<wsdl:output message="tns:TautologySoapOut" />
</wsdl:operation>
</wsdl:portType>
</wsdl:operation>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```
Table 3: Percentage analysis.

<table>
<thead>
<tr>
<th>Questions</th>
<th>Excellent</th>
<th>Very Good</th>
<th>Satisfactory</th>
<th>Very Poor</th>
<th>Unacceptable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to use</td>
<td>MATWS</td>
<td>AEPLT</td>
<td>MATWS</td>
<td>AEPLT</td>
<td>MATWS</td>
</tr>
<tr>
<td>Number</td>
<td>7</td>
<td>3</td>
<td>8</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>%</td>
<td>35</td>
<td>15</td>
<td>40</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>Buttons access</td>
<td>MATWS</td>
<td>AEPLT</td>
<td>MATWS</td>
<td>AEPLT</td>
<td>MATWS</td>
</tr>
<tr>
<td>Number</td>
<td>10</td>
<td>1</td>
<td>9</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>%</td>
<td>50</td>
<td>5</td>
<td>45</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Efficiency</td>
<td>MATWS</td>
<td>AEPLT</td>
<td>MATWS</td>
<td>AEPLT</td>
<td>MATWS</td>
</tr>
<tr>
<td>Number</td>
<td>7</td>
<td>4</td>
<td>9</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>%</td>
<td>35</td>
<td>20</td>
<td>45</td>
<td>40</td>
<td>20</td>
</tr>
<tr>
<td>Visual appearance</td>
<td>MATWS</td>
<td>AEPLT</td>
<td>MATWS</td>
<td>AEPLT</td>
<td>MATWS</td>
</tr>
<tr>
<td>Number</td>
<td>11</td>
<td>2</td>
<td>7</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>%</td>
<td>55</td>
<td>10</td>
<td>35</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Easy recovery from errors</td>
<td>MATWS</td>
<td>AEPLT</td>
<td>MATWS</td>
<td>AEPLT</td>
<td>MATWS</td>
</tr>
<tr>
<td>Number</td>
<td>10</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>%</td>
<td>50</td>
<td>10</td>
<td>35</td>
<td>20</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. Design of the User Interface

A user interface is the system by which service requesters interact with a machine. User interfaces exist for a wide range of systems and provide

(i) input, allowing the users to manipulate a system

(ii) output, allowing the system to indicate the effects of the users’ manipulation.

The user interface is one of the most important parts of any program because it determines how easily you can make the program do what you want. A good program with a badly designed user interface has little value. The aim of this paper is to produce a user interface which makes it easy, efficient, and enjoyable for the service requester to use. The user interface has a virtual keyboard and some controls that can be created dynamically as and when needed.

6.1. Design of the Virtual Keyboard. A virtual keyboard is provided, which is an arrangement of buttons imitating a physical keyboard. Virtual keyboard has alphabets and mathematical symbols like $\frac{1}{2}$, $\land$, $\lor$, $\downarrow$, and so forth; Pressed Key values are displayed in a label which uses session variable to hold its value even postback occurs. All keys are designed to use their click event to set its value in the session variable. It is believed that the virtual keyboard is a very powerful input component for mathematical expressions, which will prove useful even in the presence of a physical keyboard, where the buttons react to mouse clicks.

7. Results and Discussion

The user interface is evaluated by conducting a case study and the web services are evaluated by using web services performance metrics.

7.1. Evaluating Performance of the User Interface. To evaluate the performance of user interface, twenty-nine end-users are selected randomly to participate in the experiment and they do not have any preknowledge on web services. Nine end-users are not interested with the invocation of web services. Out of the remaining twenty end-users, ten are from mathematical background and are very familiar with Internet and ten have only the basic knowledge about Internet.

To evaluate the user satisfaction of the user interface, a user study [12] has been conducted. The twenty end-users are asked to invoke the web services using AEPLT and MATWS user interface.

After the end-users completed the invocation, a user feedback is obtained by asking them to choose options for the given questions to assess their usability with the user interface. The rational of each question is indicated in parentheses. The specific research questions posed were [13] the following.

(1) Is it easy to operate? (Easy to use)
(2) Are buttons of the right size and can be located easily? (Buttons access)
(3) Is it efficient to use? (Efficiency)
(4) Is it visually pleasing? (Visual appearance)
(5) Does it provide easy recovery from errors? (Easy recovery from errors).

The survey is based on a five-point Likert scale [14] which provides five choices for each question to measure the degree of the answer, such as Excellent, Very Good, Satisfactory, Very Poor, and Unacceptable. Those answers are mapped to the score of 1 to 5. Five represents the most positive answer and 1 represents the most negative answer. Therefore quantitative data is obtained, so that the data can be analyzed with relative ease. Table 3 shows the percentage analysis of the obtained data. The percentage analysis in Figure 4 shows that MATWS
Table 4: Descriptive statistical analysis.

<table>
<thead>
<tr>
<th>Questions</th>
<th>Median</th>
<th>Mode</th>
<th>Range</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to use</td>
<td>MATWS</td>
<td>4</td>
<td>AEPLT</td>
<td>3</td>
</tr>
<tr>
<td>Buttons access</td>
<td>MATWS</td>
<td>5</td>
<td>AEPLT</td>
<td>4</td>
</tr>
<tr>
<td>Efficiency</td>
<td>MATWS</td>
<td>4</td>
<td>AEPLT</td>
<td>4</td>
</tr>
<tr>
<td>Visual appearance</td>
<td>MATWS</td>
<td>5</td>
<td>AEPLT</td>
<td>5</td>
</tr>
<tr>
<td>Easy recovery from errors</td>
<td>MATWS</td>
<td>5</td>
<td>AEPLT</td>
<td>5</td>
</tr>
</tbody>
</table>

Figure 4: Percentage analysis.

Figure 5: Median.

Figure 6: Mode.

Table 4 summarizes the quantitative information of the feedback received from the service requesters. The graph in Figure 5 shows that the middle value in MATWS is 4 and above which denotes that more number of feedbacks are in the Very Good and Excellent category. In AEPLT, it is 3 which has highest percentage in Excellent and Very Good category compared to AEPLT. There is no Unacceptable and Very Poor feedback in MATWS. AEPLT is dominant in the Satisfactory range only.

Table 4 illustrates the quantitative information of the feedback received from the service requesters. The graph in Figure 5 shows that the middle value in MATWS is 4 and above which denotes that more number of feedbacks are in the Very Good and Excellent category. In AEPLT, it is 3 which has highest percentage in Excellent and Very Good category compared to AEPLT. There is no Unacceptable and Very Poor feedback in MATWS. AEPLT is dominant in the Satisfactory range only.

The repetition of the values 4 and 5 most often in MATWS indicates that feedbacks are in the upper level of five point scale as shown in Figure 6. In AEPLT, it is 3 which shows that interface has been rated Satisfactory by most users.

The range 2 in MATWS indicates that the lower level is 3 which means that feedbacks are not in the Very Poor and Unacceptable category. But in AEPLT as shown in Figure 7, it denotes that feedbacks are in both the upper and lower level of the five point Likert scale equally.
is 4 which indicates that the lower level is 1 which shows the users have rated in the lower level of the Likert scale also.

The mean for MATWS is 4 and above. Hence a standard deviation less than one as shown in Figure 8 indicates that the feedback is not in the Very Poor and Unacceptable range. For AEPLT, the mean is from 2 to 4 and standard deviation greater than 1 indicates that the feedback is scattered in all levels.

7.2. Performance of Web Services. Service providers need to know the resource demands of the web services at different workloads to compete with other service providers offering the same or similar service. Clients need to know the response time, throughput of the web services. Hence stress testing is performed on the web services by allowing more number of agents to access the services simultaneously. Web services have been deployed in the local host with Intel Core 2 Duo Processor and 2 GB RAM. The Pylot Web service performance tool is used to analyse the performance of web services.

The following web service performance metrics are considered for evaluating the web services.

(i) Response Time. Response time is defined as the interval from when a user initiates a request to the instant at which the response is received by the application. To calculate the response time, stress testing is performed on the server side by allowing different number of agents 10, 20, 30, 40, and 50 to access services simultaneously. The resultant graph in

Figure 9 shows that the response time increases linearly and MATWS survives more load than AEPLT.

(ii) Throughput. Throughput is calculated as the total number of invocations for a given period of time. The graph in Figure 10 shows that, as time increases, more number of requests is handled by MATWS compared to AEPLT.

8. Conclusion

In the past few years, the web communities have undergone a vast change to the user-centric environment and we argue that MATWS for prepositional logic and set theory with user interface can bring more and more benefits to the service requester. The results indicate that the user interface using virtual keyboard help service requester for easy participation and rich user interaction. Also it has been proved that the created web services are efficient by conducting stress testing on the server side and evaluating the web service performance metrics.

Future work involves creation of more number of web services in other areas of discrete mathematics such as algebra and number theory. The existing discovery techniques are
more suitable for e-commerce based applications and hence it has been planned to create an efficient framework for the discovery of mathematical web services.

Conflict of Interests

The authors Adlin Sheeba and Chandrasekar Arumugam declare that there is no conflict of interests regarding the publication of this paper.

References


