IST-2001-33127

SciX

Open, self organising repository for scientific information exchange

SciX Open Publishing Services (SOPS)

Version: 1.0
Date: March 31, 2004

Table of Contents:

1. ARCHITECTURE SUMMARY

This section summarizes the architecture of the SciX System to an extent needed to understand the documentation for end users and developers. This section is based on the SciX Deliverable D09 where the details of the architecture can be found.

1.1 Introduction

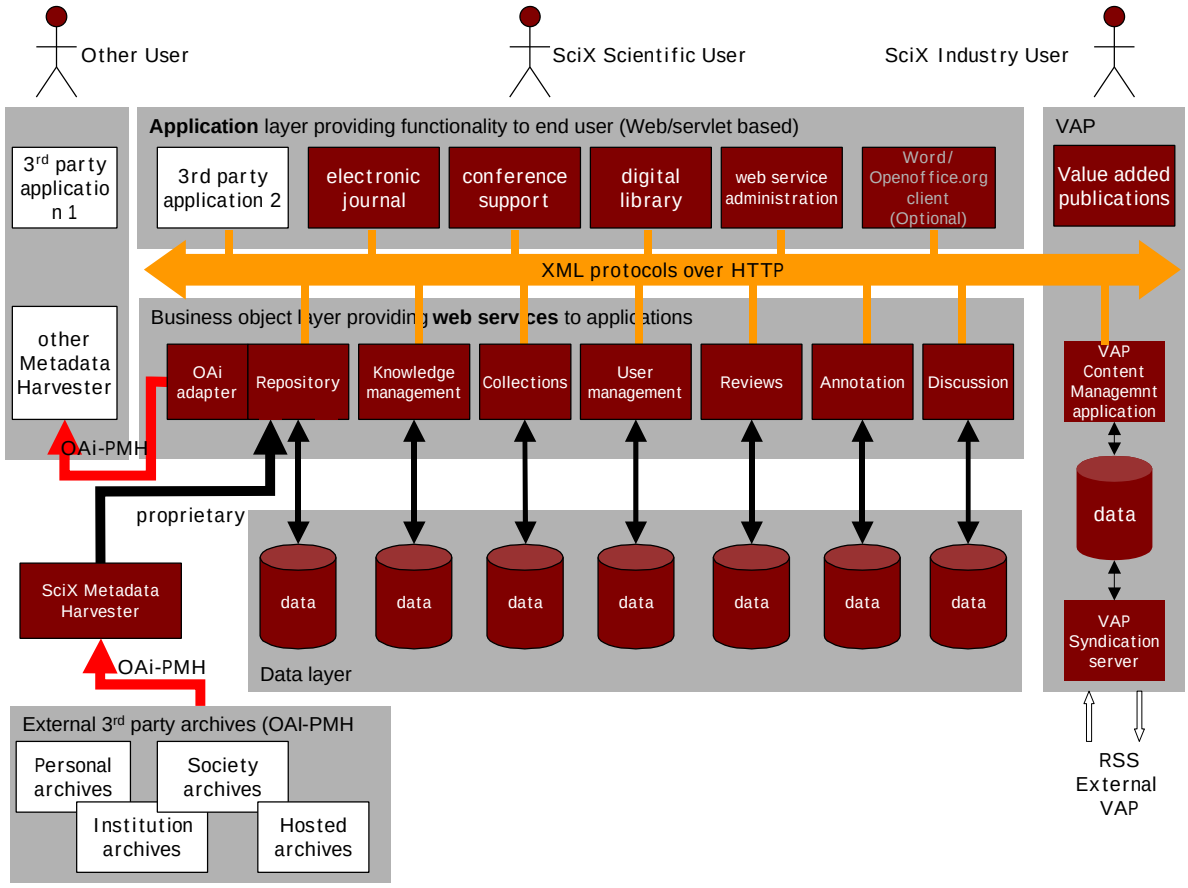

Figure 1: Logical architecture of the SciX pilot showing all major SciX and 3rd party components and ways of communication among them. SciX developments are shown as dark boxes with white text. 3rd party components are shown as white boxes with black text. XML over HTTP communication is denoted by orange, OAI-PMH by red and proprietary or other communication by black arrows.

The SciX system has a modular architecture allowing modules to be included, left out, added, or replaced relatively easily in any particular implementation or application. This is made possible because the schema shown in Figure 1. Schema is not implemented in a monolithic relational database application but rather by a numbed of services and applications. The collaboration among them in applications is established by programmers and system integrators at runtime. The communication among the modules uses either a more efficient proprietary mechanism (if the modules are physically ion the same server) or XML/SOAP if they are running on different servers.

Figure 1 depicts all major components of the SciX pilots. It is important to stress, that it does not show one system, but various components that, when properly combined; result in different applications offering different functionality to the end user. The figure is described top to bottom.

1.2 Applications

Applications are described in relation to Figure 1, left to right.

1.2.1 3rd Party Application 1

This example application's relation to a SciX based system is only through accessing the data in the SciX digital library (managed by SciX repository service) via their metadata harvester that is harvesting a SciX repository using the OAI-PMH protocol. SciX repository includes a module - the OAI Adapter that makes it possible to harvest by that protocol.

1.2.2 3rd Party Application 2

This example application is using SciX services. For example, it could use the SciX repository service but build around it some particular workflow system to support a particular publishing model. This application is talking to SciX services using an XML protocol.

1.2.3 Electronic journal

The electronic journal application provides the functionality to support an electronic journal. It supports the submission, reviewing, rewriting, publishing, reading, citing, annotating, discussing etc. electronic journal papers. The application is pulling together, in a particular way, the resources offered by some of the services in the services layer. The application is used to support the ITcon journal at www.itcon.org.

1.2.4 Conference support

The conference application provides the functionality to support organization or a conference or workshop. It supports the registration of participants, submission of abstracts, reviewing of abstracts, submission of papers, reviewing and publishing. The application is pulling together, in a particular way, the resources offered by some of the services on the services layer. Particularly the workflow in the conference support is different to the journal. In SciX this application is demonstrated by supporting the ECPPM 2004 and ELPUB 2004 conferences at extranet.2004.ecppm.org.

1.2.5 Digital library

This is the most basic of all the applications and mainly provides for access to the repository service. Other services may or may not be included, depending on how feature-packed a digital library application is desired to be. In SciX this application is demonstrated by digital libraries such as cumincad.scix.net, itc.scix.net and several others.

1.2.6 Web service administration

Although Web services are only used through the applications, they do need an administrative user interface that is represented by this box. The administration functions include extending or customizing the schema of the data that the service is handling, setting up access controls, monitoring use through service level log files, doing large scale data management (such as backing up data etc). All SciX applications use this module.

1.2.7 Value Added Publications (VAP)

VAP applications provide the normal industry user with access to industry specific articles such as digests, reviews and summaries that are produced, by VAP editors and publishers, from SciX digital archives. The VAP provides the functionality for maintaining edited articles, collaborative authoring and versioning and publication of articles to Web sites and other VAPs. The application enables e.g. digest editors to use the SciX repository service to search and browse the digital archives, to read and upload papers, produce citations and bibliographic notes.

1.3 Overview of Services

The 3rd layer from the top of Figure 1 shows the SciX services. This section briefly describes the services included in the SOPS. The name in [brackets] is the name of the software module supporting this service.

service	name of software module	short description
Basic SOPS Modules		
Repository service	works.pl	The repository service provides access to "works" and the basic functionality such as upload article, check for uniqueness, create unique article ID, remove article from repository, cross-reference articles, create new version of article, bulk upload articles, harvest articles from external sources, export articles in standard format, enter & maintain author details, enter & maintain institution details, retrieve an article by unique id, retrieve article by key-word search, browse repository via citations, search for similar articles etc.
Repository contributions	contrib.pl	The service is inheriting a lot from the repository [works] service. Its main factor of distinction is that it allows users to add information. This is normally not allowed in the works service.
Selections	selections.pl	A selection is defined as a bag of works elements. It is created by a user, for example to define a reading list for her students or to pick, from the bibliography, a selection

service	name of software module	short description
		of papers on a given topic.
User management	users.pl	While anonymous access is important because many people do not like to be bothered with log-ins and passwords, any active use of the services requires an identification and other related actions. This is managed by the users service. Several repositories may, of course, share the same users service allowing for a single password for several services.
Conference reviews	reviews-conf.pl	Service handles peer reviews of conference papers.
Journal reviews	reviews-jour.pl	Service handles peer reviews of journal papers.
Technical reviews	reviews-tech.pl	Service handles technical review of the work, such as formatting, citation styles etc.
Discussion forum	forum.pl	The service provides support for threaded discussion about works or any other item.
Ratings	ratings.pl	The service allows users to rate (excellent-poor) and justify their rating of the works in the repository.
Series	series.pl	This service allows the creation of a series of works; for example a particular conference track could be in one series, papers related to a particular topic in another; the series may be overlapping.
News	news.pl	The news service allows the managers of an application to post news to the end users; about new features, new added works, announcements of conferences etc.
Add-on modules for SOPS		
OAI server	oai2.pl	This is an optional module for the works service that provides OAI harvesting functionality of the works in the repository.
Office Research	cgiOffice	Office Research function
WSDL generator	WSDL.pl	WSDL generator of the database

service	name of software module	short description
SOAP server	SOAP.pl	Enable SOPS tu function as web service. Two implementations are available that contain different types of data
Intelligent maintenance toolkit		The KM service provides enhanced abilities for analysis categorising, classifying, browsing and searching out information and is based on knowledge management techniques, particularly statistical text analysis and machine text learning. The module may be decomposed to smaller models that might be used separately.

Each service is typically available through a distinct URL, for example the repository service, handled by the works.pl module as http://somewhere.com/cgi/works/

Add-on modules are usually available thorugh SOPS administrative interface. Some of them may be simply accessed through path-info with no additional parameters, ie. WSDL may be accessed from any service, for example: http://somewhere.com/cgi/works/WSDL

1.4 Overview of the Protocols

This section describes, how the various elements of the Figure 1 communicate:

- the orange "bus" in the Figure denotes the communication among the SciX Web services and applications. If two components that need to collaborate are on different machines, they would communicate using XML protocols over HTTP. If the two components that need to collaborate are on the same machine, a more efficient mechanism that does not include HTTP protocol overhead, but rather system calls, would be used. The selection between the two mechanisms is done automatically and transparently to the user (or system integrator) who does not need to care about physical locations of the services.
- the red arrows denote the OAI-PMH protocol. This protocol is increasingly popular for the exchange of digital library metadata. Section 1.4.1 discusses the affected components.
- black arrows denote private and other protocols, private to one SciX service and of no interest to anyone else. SQL is an example of such a protocol for data access.

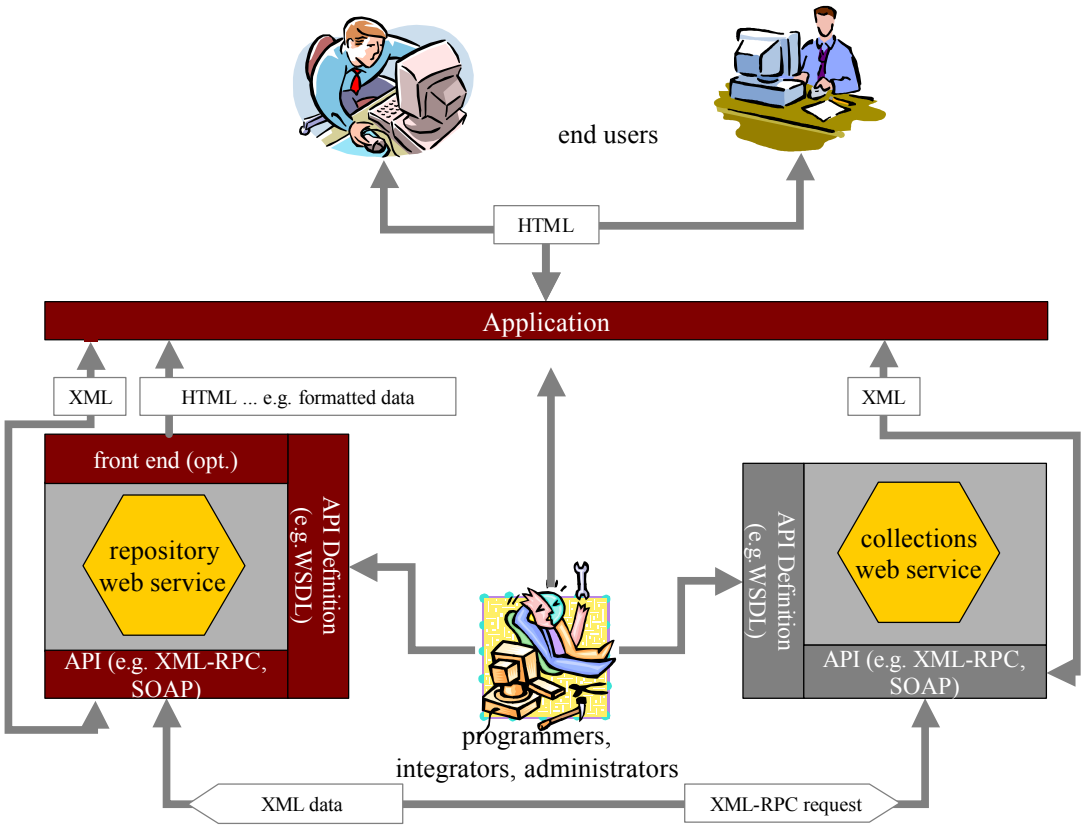

Figure 2: Detailed depiction of communication between services, applications and users.

The Figure above shows what happens in a typical scenario. Firstly, the communication is established by the programmers (centre of picture) who reads the API definitions of the two services and integrates them into, for example, a digital library application.

A digital library user sees an interesting paper in the repository and would like to see the details. The application asks the repository service to "show" the paper. It is displayed by an application, based on the data delivered (in XML or HTML) from the repository Web service. XML form is used if the application would like to do its own formatting of the data, HTML if the application is happy with the default way in which a service formats the data. The paper may be a part of a collection. To learn about this, the repository service asks the collections service using XML-RPC, if the paper in question is part of any collections. Collections service replies, in XML, with a list of collections in which the paper is present. This information is used by the repository service to generate XML or HTML data to return to the application. Finally, the application presents this to the end user inside the browser window.

1.4.1 Metadata harvesting

SciX architecture is compatible with the OAI-PMH. SciX uses an existing open source metadata harvesting software to feed 3rd party archives into SciX Repository service. SciX also provides an OAI-PMH adapter to the Repository Service, so that SciX Repositories could be harvestable by 3rd party harvesters.

2. SOPS

SciX Open Publishing Services (**SOPS**) is software that allows setting up various on-line scientific publishing media such as:

- personal archives,
- institutional archives,
- topic & society archives,
- electronic journals,
- electronic conference proceedings,
- workflow support for the above.

SOPS provides building blocks, such as repository, user management, discussions, ratings, reviews, review process support etc. out of which the above publications can be built.

SOPS is open in the sense that it provides:

- **WSDL** definitions of all available functions,
- Metadata harvesting according to the **OAI-PMH 2.0** standard.
- Compatibility with citation management software such as Reference Manager, Citation Manager and Endnote.
- Compatibility with Microsoft **Office 2003** Research Task Pane
- Really Simple Syndication (**RSS**) feeds and Office Smart Tags (comming soon).

SOPS is multilingual. It exists in English, German and Slovenian languages.

2.1 Installation overview

This section describes how SOPS is installed on a server.

2.1.1 Requirements

SOPS needs the following infrastructure:

- a server connected to the Internet with about 20 megabytes of free space + some 4 megabytes for each language version + space for data. Minimal hardware would be an Intel Pentium 133 Mhz class machine.
- an httpd server, such as Apache or Xitami.
- WODA Database and Web services generator.
- a person with some knowledge of Web Severs and Perl language.

The above infrastructure should be installed and tested. The WODA environment is a key requirement; it enables that the footprint of the individual services is very small and therefore manageable by the person installing and managing SOPS. WODA is documented at www.ddatabase.com. The reader should consult those documents on how to install it. The next section provides some essential information. Most of the applications and functions may be run whether on Unix, Linux or Windows operating systems meeting the requirements mentioned above. Solutions to troubleshooting to any WODA based web applications are available on the web and are continuously supported by the author of the engine.

2.2 The WODA Web Services Generator

WODA is a standalone database management system and Web Services generator for the World Wide Web. It consists of a Perl program e.g. woda-max.pl (also called the Woda engine) and several support programs (typically in /usr/local/woda), icons and javascripts (in /document-root/woda/). Each individual service is defined in the service definition file (*definition*). This file defines the data dictionary, customizes the user interface and then calls the engine to process the user request and generate the appropriate HTML or XML page. The definition is in two Perl hashes:

- Hash WBB defines features of a whole table, e.g. `$WBB{managerEmail}="joe@music.com"` defines email address of the manager of the database.
- Hash WBF defines features of table fields, e.g. `$WBF{name,help}="Enter your first name"` defines help text for the field 'name'.

A single engine therefore supports several services and is entirely service independent. All operating system dependent parameters are defined within the engine. The engines are language specific.

2.2.1 Layout of WODA files

WODA would install files to three different sections of the filesystem:

- /usr/local/woda holds WODA code in subdirectories named by the language version of Woda, for example /usr/local/woda/en/ would hold the English speaking Woda code.
- DOCUMENTDIR/woda/ holds WODA files that need to be visible by http, such as icons and JavaScript modules.
- CGIDIR/woda/ holds cgi scripts and sample applications, that come with WODA.

DOCUMENTDIR is the directory that holds documents delivered by http and CGI is the default CGI script directory of your Web Server.

Before installing SOPS it is imperative that the Woda installation is running faultlessly.

To prove that WODA is functioning properly please consult instructions at www.ddatabase.com

2.3 SOPS physical architecture

Before continuing the installation it would be appropriate to describe the physical layout of the SOPS and how it handles HTTP requests (Figure 3):

1) Web browser[1] makes a request to Web server to generate a page, for example listing papers with a certain keyword.

[1] Any http client can take this role, including OAI harvesters, Office applications, Citation Managers or other SOPS services.

2) The Web server maps the URL to a name of a file in the server's CGI directory, for example a script called works. This scrip has to assemble the declaration of a service from different sources so that it can pass the processing of the service to Woda.

3) The script first calls a file usually called common.pl. It includes the definition of the application of which a small part is the works service; it defines how various services (like works) fit together as well as common layout and appearance.

4) The common application definition includes the definitions that are common to all generic SOPS services.

5) The common application as well as all common SOPS settings are now defined.

6) The declaration of a generic SOPS works service is re-used.

7) At this point, the cgi/works may define how it is different from a plain vanilla works service; it may extend it in any way in which the Woda service generator or Perl language allow it to be done.

8) After the declaration of a service has been assembled it is passed to Woda for processing and …

9) The generation of a reply, that is first sent to the HTTP daemon that …

10) … passes the generated information to the client that made the original request.

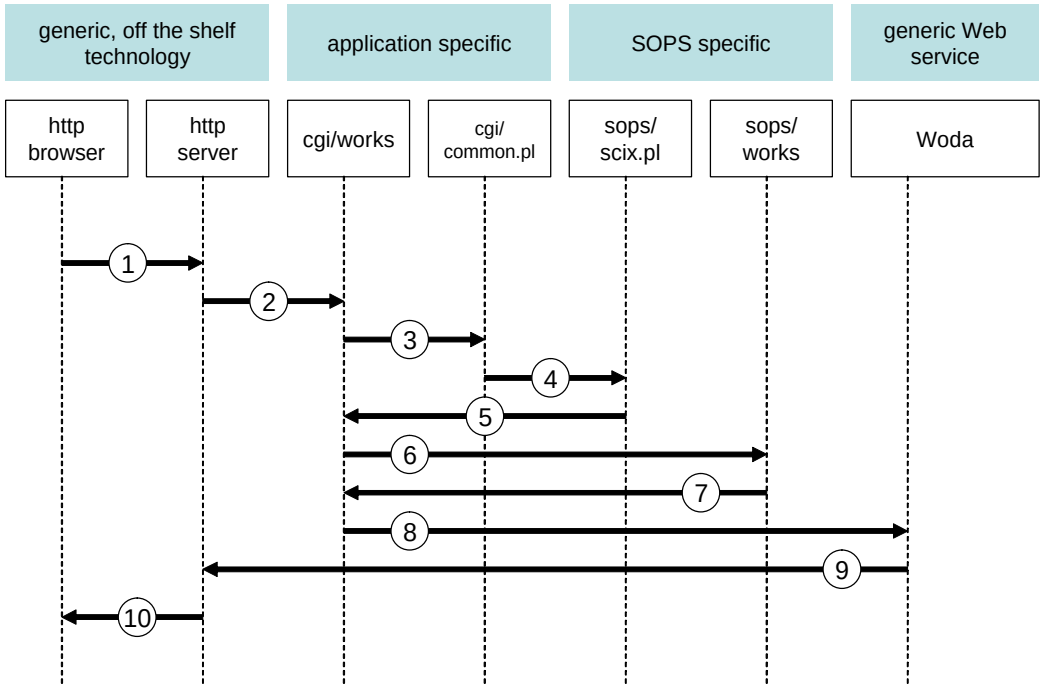

Figure 3: Serving an HTTP request by SOPS.

SOPS applications therefore have these main components:

component	role	example	typical size	reused
custom sops service script	assembles service definition, customizes generic SOPS service	cgi/work	100 bytes	not at all
application script	defines application and common application features	cgi/common.pl	10000 bytes	by 5-10 SOPS service scripts
SOPS script	defines common features of all SOPS services on this server	sops/scix.pl	500 bytes	by all generic sops scrips on this server
SOPS generic service definition	defines a SOPS service as discussed in Section 1.3.	sops/works.pl	10.000 bytes	by all custom sops service scripts on this server
WODA	interprets the service definition as assembled from the definitions above	woda/uk/woda.pl	500 kbytes	by SOPS and other services on this server

2.4 Installing SOPS

2.4.1 Automatic installation:

- untar the sops.tar.gz into an empty directory and run the command *perl install*

The installer will ask you a few questions and do what is described below under manual installation.

2.4.2 Manual installation:

- untar the sops.tar.gz into an empty directory;
- two directories will be created, sops and cgi; sops contains files related to SOPS specific layer, cgi contains application specific cgi scripts.
- copy all that is in the sops directory to /usr/local/woda/LL/lib/sops; make these files readable by the httpd process; replace LL with en, de or sl if you are installing English, DEutch or Slovenian version of SOPS.
- copy the files in cgi/ directory to the cgi directory of your server and make them readable and executable by the httpd process.

2.5 Configuring SOPS

2.5.1 Post install configuration of cgi scripts:

- make sure the 1st line points to perl interpreter correctly
- any other configuration can be done later.

2.5.2 Post install configuration of the cgi/common.pl file

1) Make sure that lines near the top like

require '/usr/local/woda/en/woda-max.pl';

actually do point to Woda.

2) Configure line like:

$DATADIRDIR ="/www/architektur-informatik.scix.net/htdocs/data";

to point to an existing directory where httpd process has read-write permissions and where all data of the application will be stored.

3) Configure the password for owner and administrator. Admin is the account for the computer expert that does technical maintenance of the application, owner is the account for the editor or custodian of the archive; he does content editing but is not a computer expert. Change lines like:

```
$WBB{'groups'} = <<EOM;
admin  .        passwordForAdminHere
owner  .        passwordForOwberHere
EOM
```

2.6 Try it out

Open your Web browser and open /cgi/sops/works. A menu page like the one below should appear.

If it did not, consult the troubleshooting section of Woda. Access to Web server's error log file is very valuable. Typically this is */var/log/access*.

2.7 Modifying the sops services

SOPS services can be modified by making changes to the:

- application specific files in the cgi path, in particular the common.pl and the cgi script files
- SOPS library files in the lib/sops directory
- WODA

The only intended modifications should be to the first set of files; in this case the user will be compatible with the future versions of SOPS and WODA.

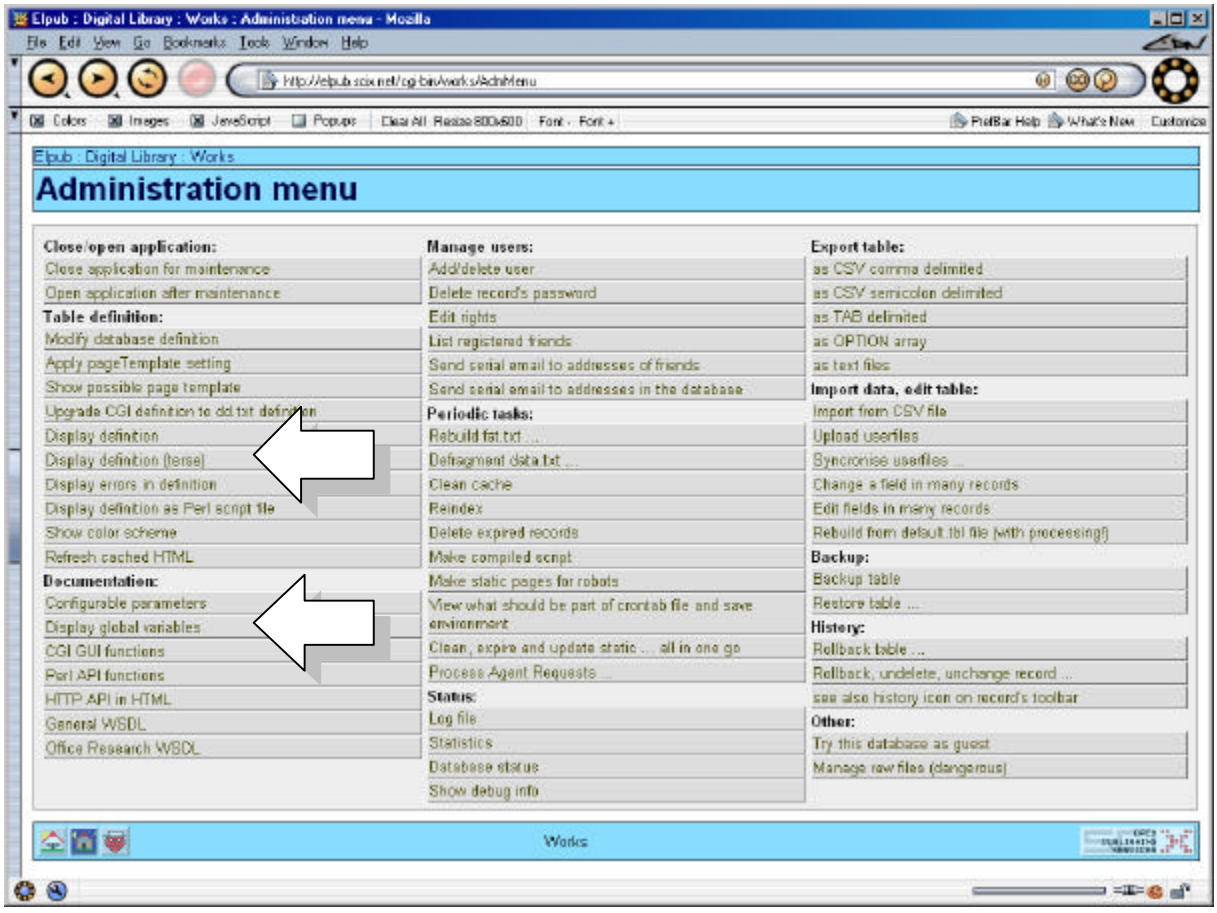

Figure 4: Service's administration menu.

The modifications are changes one of the dozens service parameters. The documentation about these parameters is given by each service form the service Administration Menu. Of interest at this point are the options:

1) Configurable parameters that displays the documentation of the configurable parameters.

2) Display definition that displays the current setting of the service definition. A sample output of this command is listed in Appendix D, Subsection 4.

2.8 Machine readable documentation (WSDL)

2.8.1 Traditional functionality

Web services are more and more emerging. Web service interfaces for already available traditional services, like searching the Web or querying the online bookstore are nowadays transformed to WSDL specification and accessible through SOAP clients. All SOPS service can describe themselves in WSDLs.

2.8.2 MS Office research task pane

Next novelty that is supported by SOPS services is Microsoft Research task pane, which was introduced recently and implemented in Office 2003.

This Open Software License (the "License") applies to any original work of authorship (the "Original Work") whose owner (the "Licensor") has placed the following notice immediately following the copyright notice for the Original Work: "**Licensed under the Open Software License version 1.0**"

License Terms

1) **Grant of Copyright License.** Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, non-sublicenseable license to do the following:

a) to reproduce the Original Work in copies;

b) to prepare derivative works ("Derivative Works") based upon the Original Work;

c) to distribute copies of the Original Work and Derivative Works to the public, with the proviso that copies of Original Work or Derivative Works that You distribute shall be licensed under the Open Software License;

d) to perform the Original Work publicly;

e) to display the Original Work publicly.

2) **Grant of Patent License.** Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, non-sublicenseable license, under patent claims owned or controlled by the Licensor that are embodied in the Original Work as furnished by the Licensor ("Licensed Claims") to make, use, sell and offer for sale the Original Work. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, non-sublicenseable license under the Licensed Claims to make, use, sell and offer for sale Derivative Works.

3) **Grant of Source Code License.** The term "Source Code" means the preferred form of the Original Work for making modifications to it and all available documentation describing how to access and modify the Original Work. Licensor hereby agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as Licensor continues to distribute the Original Work, and by publishing the address of that information repository in a notice immediately following the copyright notice that applies to the Original Work.

4) **Exclusions From License Grant.** Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectual property of Licensor except as expressly stated herein. No patent license is granted to make, use, sell or offer to sell embodiments of any patent claims other than the Licensed Claims defined in Section 2. No right is granted to the trademarks of Licensor even if such marks are included in

the Original Work. Nothing in this License shall be interpreted to prohibit Licensor from licensing under different terms from this License any Original Work that Licensor otherwise would have a right to license.

5) **External Deployment.** The term "External Deployment" means the use or distribution of the Original Work or Derivative Works in any way such that the Original Work or Derivative Works may be accessed or used by anyone other than You, whether the Original Work or Derivative Works are distributed to those persons, made available as an application intended for use over a computer network, or used to provide services or otherwise deliver content to anyone other than You. As an express condition for the grants of license hereunder, You agree that any External Deployment by You shall be deemed a distribution and shall be licensed to all under the terms of this License, as prescribed in section 1(c) herein.

6) **Warranty and Disclaimer of Warranty.** LICENSOR WARRANTS THAT THE COPYRIGHT IN AND TO THE ORIGINAL WORK IS OWNED BY THE LICENSOR OR THAT THE ORIGINAL WORK IS DISTRIBUTED BY LICENSOR UNDER A VALID CURRENT LICENSE FROM THE COPYRIGHT OWNER. EXCEPT AS EXPRESSLY STATED IN THE IMMEDIATELY PRECEEDING SENTENCE, THE ORIGINAL WORK IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THE WARRANTY OF NON-INFRINGEMENT AND WARRANTIES THAT THE ORIGINAL WORK IS MERCHANTABLE OR FIT FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL WORK IS WITH YOU. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO LICENSE TO ORIGINAL WORK IS GRANTED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

7) **Limitation of Liability.** UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL THE LICENSOR BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER ARISING AS A RESULT OF THIS LICENSE OR THE USE OF THE ORIGINAL WORK INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PERSON SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8) **Acceptance and Termination.** Nothing else but this License (or another written agreement between Licensor and You) grants You permission to create Derivative Works based upon the Original Work, and any attempt to do so except under the terms of this License (or another written agreement between Licensor and You) is expressly prohibited by U.S. copyright law, the equivalent laws of other countries, and by international treaty. Therefore, by exercising

any of the rights granted to You in Sections 1 and 2 herein, You indicate Your acceptance of this License and all of its terms and conditions. This license shall terminate immediately and you may no longer exercise any of the rights granted to You by this License upon Your failure to honor the proviso in Section 1(c) herein.

9) **Mutual Termination for Patent Action.** This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License if You file a lawsuit in any court alleging that any OSI Certified open source software that is licensed under any license containing this "Mutual Termination for Patent Action" clause infringes any patent claims that are essential to use that software.

10) **Jurisdiction, Venue and Governing Law.** You agree that any lawsuit arising under or relating to this License shall be maintained in the courts of the jurisdiction wherein the Licensor resides or in which Licensor conducts its primary business, and under the laws of that jurisdiction excluding its conflict-of-law provisions. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any use of the Original Work outside the scope of this License or after its termination shall be subject to the requirements and penalties of the U.S. Copyright Act, 17 U.S.C. Â§ 101 et seq., the equivalent laws of other countries, and international treaty. This section shall survive the termination of this License.

11) **Attorneys Fees.** In any action to enforce the terms of this License or seeking damages relating thereto, the prevailing party shall be entitled to recover its costs and expenses, including, without limitation, reasonable attorneys' fees and costs incurred in connection with such action, including any appeal of such action. This section shall survive the termination of this License.

12) **Miscellaneous.** This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

13) **Definition of "You" in This License.** "You" throughout this License, whether in upper or lower case, means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with you. For purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

This license is Copyright (C) 2002 Lawrence E. Rosen. All rights reserved. Permission is hereby granted to copy and distribute this license without modification. This license may not be modified without the express written permission of its copyright owner.

2.9 Changes of the License for SOPS

In clause 1 add:

f) to perform or display the Derived Work publicly with the proviso that Derivative Works that You perform or display shall be licensed under the Open Software License;

Clause 3 should read:

3) **Grant of Source Code License.** The term "Source Code" means the preferred form of the Original Work for making modifications to it and all available documentation describing how to access and modify the Original Work. Licensor hereby agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes, performs or displays publicly. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as Licensor continues to distribute the Original Work, and by publishing the address of that information repository in a notice immediately following the copyright notice that applies to the Original Work.