

IST-2001-33127

SciX

Open, self organising repository for scientific
information exchange

D10: Implementation report

Responsible authors: Tomo Cerovšek, Žiga Turk, Gudni Gudnason,
Brian Clifton, Bob Martens, Grahame Cooper

Access: public
Version: 1.0
Date: March 31, 2004

Executive summary

This implementation report gives an overview of the implementation of the SciX platform. It is summarizing the architecture of the SciX platform and then documents the major components of the SciX platform – the SciX Open Publishing Platform (SOPS), the Advanced Services, the SciX Harvester and the Value Added Publications services.

The report is aimed at those who will be installing, maintaining and extending the SciX developments. Advanced services are a set of web services, tools and SOPS add-on utilities dealing with contents stored in SOPS services, integration, applications based on SOPS services, as well as contents generated through the use of SOPS services and extended applications, including user activities. Several services are presented in more detail including transaction matrix analyzer, clustering services, citation analysis, and scientific publishing services.

The three appendixes include: (1) License under which SOPS services may be distributed - called "Licensed under the Open Software License version 1.0"; (2) Typical SOPS files and settings which are read for implementers to be and (3) WSDL (Web Services Definition Language) documenting functionalities of the major components of SOPS.

Release history of this deliverable (D9):

date	changes
31.3.2004	Version 1.0 released
6.4.2004	Version 1.01 released including minor updates and corrections to the text

Table of Contents:

Executive summary.....	2
Table of Contents:.....	4
1. ARCHITECTURE SUMMARY.....	7
1.1 Introduction.....	7
1.2 Applications	8
1.2.1 3 rd Party Application 1.....	8
1.2.2 3 rd Party Application 2.....	9
1.2.3 Electronic journal	9
1.2.4 Conference support	9
1.2.5 Digital library	9
1.2.6 Web service administration	9
1.2.7 Value Added Publications (VAP).....	9
1.3 Overview of Services.....	10
1.4 Overview of the Protocols	12
1.4.1 Metadata harvesting.....	13
2. SOPS.....	14
2.1 Installation overview.....	14
2.1.1 Requirements.....	14
2.2 The WODA Web Services Generator	15
2.2.1 Layout of WODA files	15
2.3 SOPS physical architecture	15
2.4 Installing SOPS	17
2.4.1 Automatic installation:	17
2.4.2 Manual installation:.....	17
2.5 Configuring SOPS	18
2.5.1 Post install configuration of cgi scripts:.....	18
2.5.2 Post install configuration of the cgi/common.pl file	18
2.6 Try it out.....	18
2.7 Modifying the sops services.....	18
2.8 Machine readable documentation (WSDL)	19
2.8.1 Traditional functionality.....	19
2.8.2 MS Office research task pane.....	20
3. Advanced Services.....	22
3.1 Advanced services overview.....	22
3.1.1 Methodological aspects	22

3.2	Scientific conferences review and submission support	23
3.2.1	Requirements.....	23
3.2.2	Installation	23
3.2.3	Running the service.....	23
3.3	Transaction matrix analyzer.....	24
3.3.1	Requirements.....	24
3.3.2	Installation	24
3.3.3	Input data.....	24
3.3.4	Procedures and results	24
3.4	Collections statistics generator, similarity explorer	25
3.4.1	Requirements.....	25
3.4.2	Installation	25
3.5	Clustering services	25
3.5.1	Requirements.....	25
3.5.2	Input data.....	25
3.5.3	Procedure and results.....	25
3.6	Citation analysis	26
3.6.1	Input data.....	26
3.6.2	Pre-processing of raw data.....	27
3.6.3	Processing of references.....	29
3.6.4	Results	29
3.7	Quantitative analysis of referencing habits	29
3.7.1	Analysis on authors of cited material.....	32
3.7.2	Discussion on citation analysis	36
4.	SciX Harvester.....	38
4.1	Definitions	38
4.1.1	Harvester.....	38
4.1.2	Repository.....	38
4.1.3	Item	38
4.1.4	Unique Identifier.....	38
4.1.5	Record	39
4.2	SciX Harvester.....	40
4.2.1	SciX Harvester – OUT	40
4.2.2	SciX Harvester – IN.....	40
4.2.3	OAI-compliant Repositories.....	40
5.	VALUE-ADDED PUBLICATIONS.....	41
5.1	Introduction.....	41
5.2	Architecture overview.....	42
5.3	Deployment	44
5.4	Installation and configuration	45
5.4.1	Content Management System (CMS)	45
5.4.2	Syndication Server	47

5.4.3	Syndication Client	49
5.4.4	Installing servers as a Windows service	50
5.5	User's documentation overview	50
5.5.1	Content Management System.....	50
5.5.2	Syndication Server	51
5.5.3	Syndication Client	53
5.6	License	54
6.	Appendix A: SOPS license	55
6.1	Changes of the License for SOPS	57
7.	Appendix B: Some typical SOPS files and Settings	59
7.1	cgi/works	59
7.2	cgi/common.pl.....	59
7.3	lib/sops/ratings.pl.....	60
7.4	Compound definition of the Works services	61
7.4.1	Field settings	65
7.5	Scientific conferences review and submission support	67
8.	Appendix C: WSDL definitions	72
8.1	works Office 2003 WSDL	72
8.2	works WSDL.....	73

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

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 number 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 on the same server) or XML/SOAP if they are running on different servers.

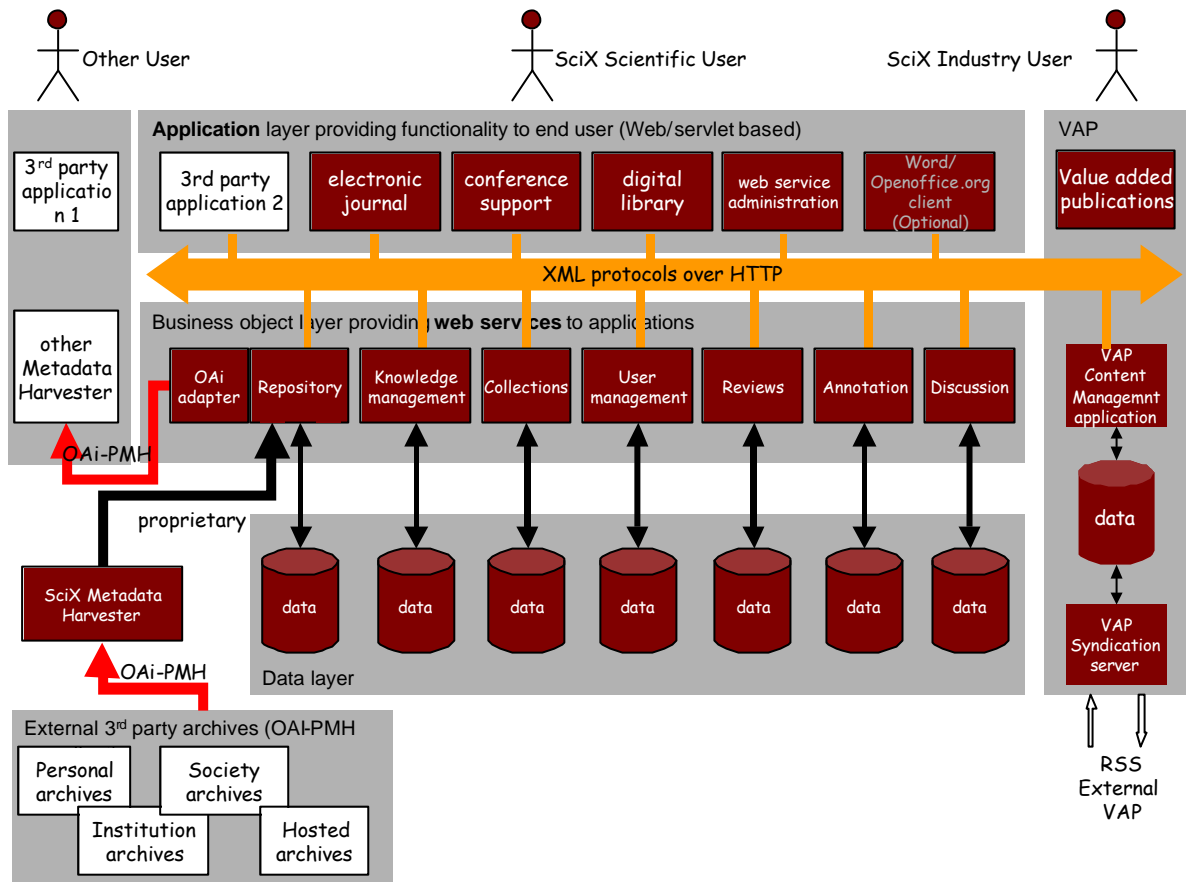


Figure 1: Logical architecture of the SciX pilot showing all major SciX and 3rd party components and ways of communication among them. SciX developments are shown as dark boxes with white text. 3rd party components are shown as white boxes with black text. XML over HTTP communication is denoted by orange,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

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 of 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 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.

service	name of software module	short description
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
SOAP server	SOAP.pl	Enable SOPS to 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 through 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.

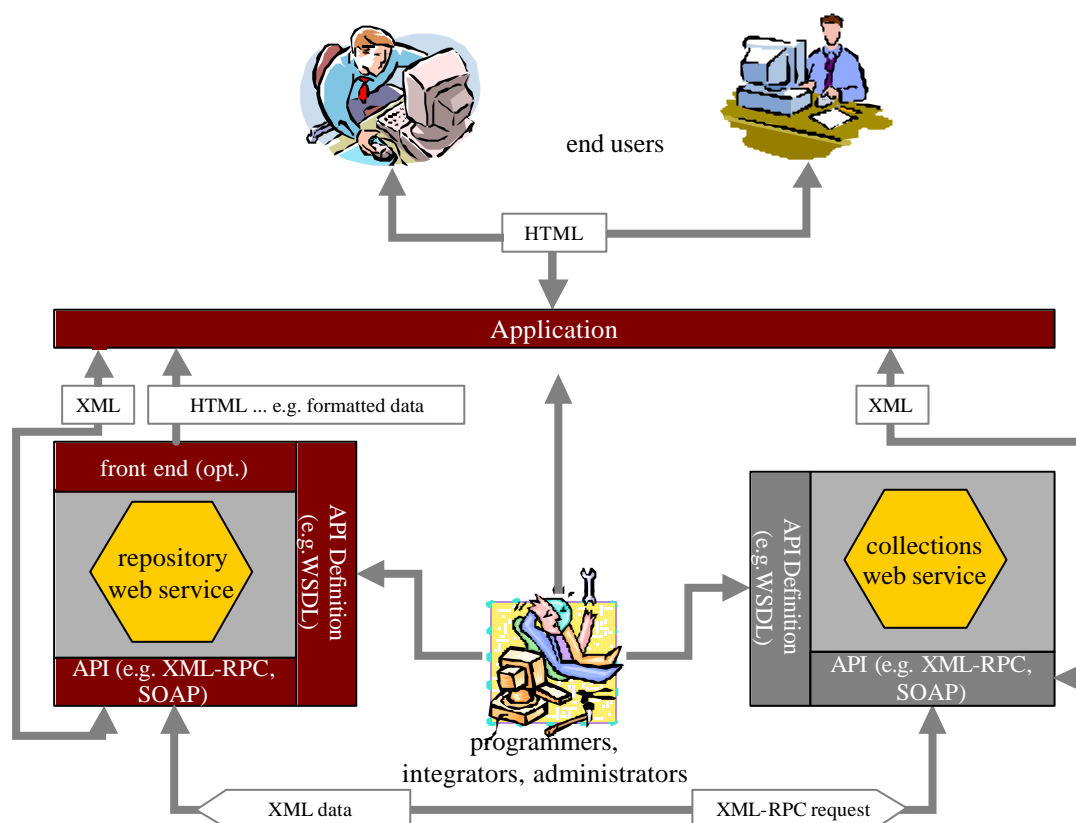


Figure 2: Detailed depiction of communication between services, applications and users.

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 (coming 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 Servers 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 browser¹ 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.

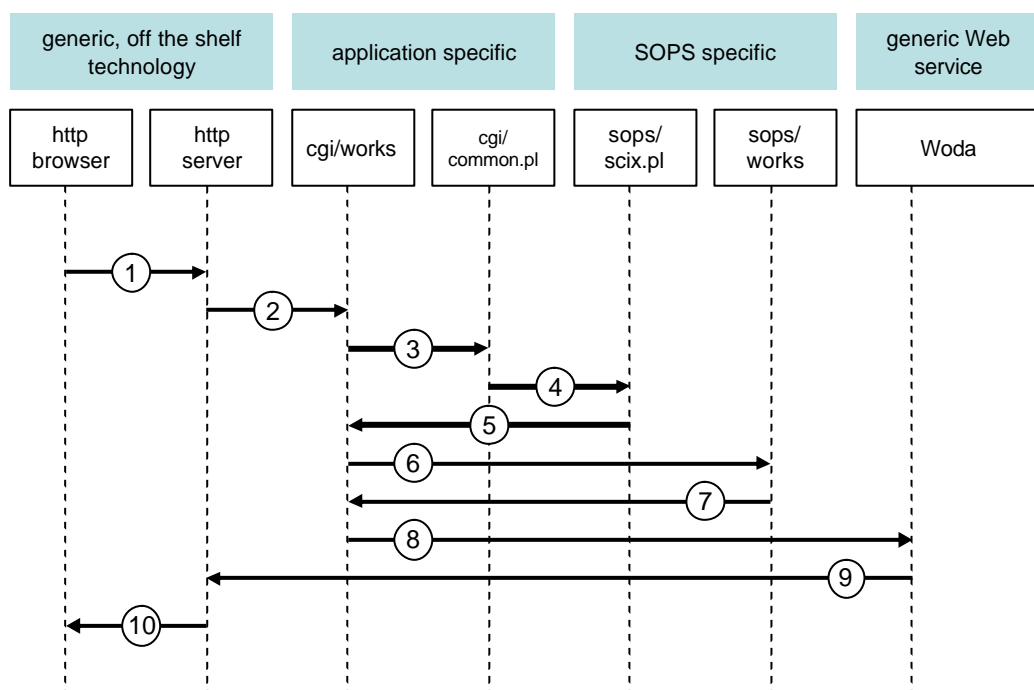


Figure 3: Serving an HTTP request by SOPS.

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.

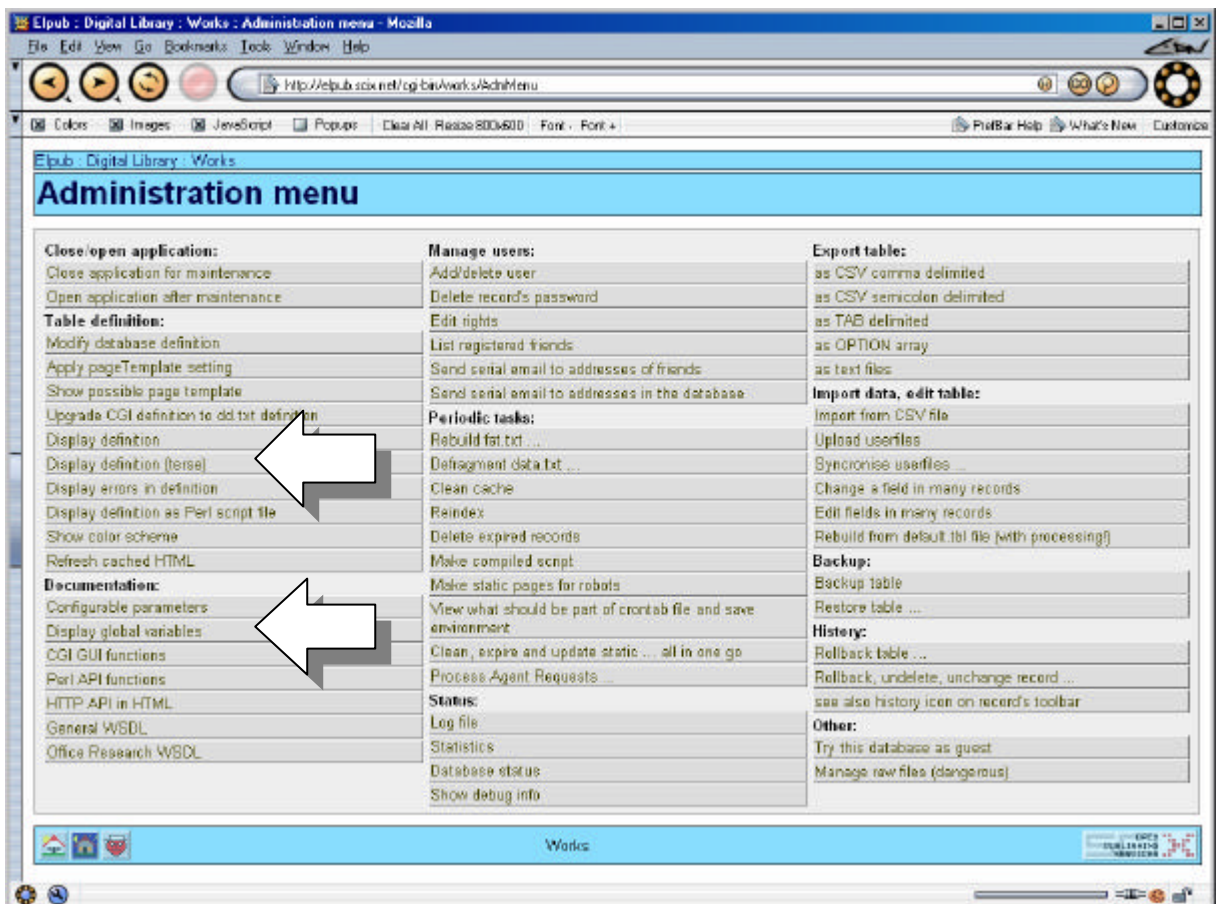


Figure 4: Service's administration menu.

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, which are listed in the appendix B.

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. The schema above shows the functionality.

```
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:s0="urn:Microsoft.Search" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" targetNamespace="urn:Microsoft.Search"
xmlns="http://schemas.xmlsoap.org/wsdl/">
<types>
<s:schema elementFormDefault="qualified" targetNamespace="urn:Microsoft.Search">
<s:element name="Registration">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="registrationXml" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
<s:element name="RegistrationResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="RegistrationResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
<s:element name="Query">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="queryXml" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
<s:element name="QueryResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="QueryResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
</s:schema>
</types>
<message name="RegistrationSoapIn">
<part name="parameters" element="s0:Registration" />
</message>
<message name="RegistrationSoapOut">
<part name="parameters" element="s0:RegistrationResponse" />
</message>
<message name="QuerySoapIn">
<part name="parameters" element="s0:Query" />
</message>
<message name="QuerySoapOut">
<part name="parameters" element="s0:QueryResponse" />
```

```

</message>
<portType name="ResearchSoap">
<operation name="Registration">
<documentation>Registration for the Woda Search Service.</documentation>
<input message="s0:RegistrationSoapIn" />
<output message="s0:RegistrationSoapOut" />
</operation>
<operation name="Query">
<documentation>Runs a search in Woda.</documentation>
<input message="s0:QuerySoapIn" />
<output message="s0:QuerySoapOut" />
</operation>
</portType>
<binding name="ResearchSoap" type="s0:ResearchSoap">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
<operation name="Registration">
<soap:operation soapAction="urn:Microsoft.Search/Registration" style="document" />
<input>
<soap:body use="literal" />
</input>
<output>
<soap:body use="literal" />
</output>
</operation>
<operation name="Query">
<soap:operation soapAction="urn:Microsoft.Search/Query" style="document" />
<input>
<soap:body use="literal" />
</input>
<output>
<soap:body use="literal" />
</output>
</operation>
</binding>
<service name="Research">
<documentation>Woda Research Web Service</documentation>
<port name="ResearchSoap" binding="s0:ResearchSoap">
<soap:address location="http://itc.scix.net/cgi-bin/series/Office" />
</port>
</service>
</definitions>
  
```

3. ADVANCED SERVICES

Advanced services are a set of web services, tools and SOPS add-on utilities dealing with:

- contents stored in SOPS services
- integration and applications based on SOPS services
- contents generated through the use of SOPS services and extended applications, including user activities

This Section contains description of some of the services. An analysis produced by one such service is presented in more detail as an illustration.

3.1 Advanced services overview

Advanced services include:

- web services providing API for any third party WSDL aware software
- transaction matrix manipulation utility
- web-log analysis,
- personal profile services,
- collections statistics generator, similarity explorer,
- integration services,
- clustering services,
- data visualization utility web services,
- citation analysis,
- and scientific conference submission and review process support.

A special type of services that utilizes the power of SOPS and as well as advanced services is web conferences managing services.

3.1.1 Methodological aspects

The work presented is guided by following basic principles:

- *Gradual development.* The goal was to enable gradual development of tools and immediate use of results. The results should be organized in a way appropriate for further use, or complementary analysis.
- *Paradigms.* The analysis was influenced by several paradigms: process-oriented model, graphical models, IR (information retrieval) concept of bag of stem words, as well as systemic view to similarity – defined by canonical form. Terms citation and reference are used interchangeably.
- *Types of analysis.* Three main types of analysis have been used to produce results presented in this paper: (1) Text mining, (2) Statistical analysis of citations (3) Graphical models as a way of interpretation of relationships between elements subjected to analysis.
- *Use of multiple techniques.* Since the methods used in the analysis do not offer sufficient, or one-and-only interpretation of the contents and relationships, multiple techniques might be applied.
- *Use of voting mechanisms.* If there are alternative methods available, which sometimes deliver different results – a mechanical voting mechanism may be used to automatically select the best result.

3.2 Scientific conferences review and submission support

There are two independent distributions of the services. First uses SOPS services and requires full installation of WODA, and second, that stand alone service that requires only Perl and nothing else.

3.2.1 Requirements

Following SOPS services are used if fully SOPS compatible solutions is installed

Repository service	works.pl	Basic metadata about works
User management	users.pl	User managements
Conference reviews	reviews-conf.pl	Service handles peer reviews of conference papers.
Technical reviews	reviews-tech.pl	Service handles technical review of the work, such as formatting, citation styles etc.

In appendix d one can find parameters that might be changed for typical conference.

3.2.2 Installation

- untar the conf.tar.gz into an empty directory that is accessible through web browser
- five scripts and file files will be generated. Make sure that scripts are executable.
- make shure that file are executable (check server settings – endings on windows, chmod on unix)
- make sure that folders are allowed to be written
- prepare template to use as a web desing and
- edit conf.config file and adjust settings (name of the template to use for desing, number of reviews, etc, admin password, etc) or run config.pl through web browser and you can set all the settings.

Next generation of conference support for review and submission services that is under the development will allow enable users of the repository to quickly create web application on the repositories' hosted servers.

3.2.3 Running the service

Simply go the server alias and you should see the scripts:if everything is installed properly you should be able to run the conferences support service.

3.3 Transaction matrix analyzer

Basic principle of the transaction matrix is very close to bag of words analysis, but can be used in several contexts. It was demonstrated in d14 that it can be successfully used in web-log analysis, but the principles are equally useful in the context of citation analysis or on any other data.

Session	Paper 1	Paper 2	...	Paper i	...	Paper n-1	Paper n
1	0	1		1		0	1
2	1	1		0		0	0
...							
J	0	0		1		1	0
...							
s-1	1	1		0		1	0
S	0	0		1		0	0

Figure 5: Transaction matrix

3.3.1 Requirements

There are no additional requirements except DBI Perl support is required.

3.3.2 Installation

Transaction matrix analyzer maybe run as command line utility or may be used as a web services (according to published WSDL). In both cases one must specify storage folders, etc. simply copy the file into library folder.

3.3.3 Input data

Any data that can be view as transaction matrix described above. This input data might be transformed into the following form. The input data must be prepared in a form similar to sparse matrix definitions for MatLab, details are given in software documentation. Utilities that operate on web logs generated by are tailored for transaction matrix analyzer.

3.3.4 Procedures and results

Basic functions that are supported by the software are:

- Store row (sparse vector)
- Find first n similar rows to a given row (RowId, n)
- Find first m similar columns to a given column (ColumnId, m)
- Create different clusters based on rows or columns
- Probability prediction (in the development)

Results are:

- Successful storage (Boolean)
- Ids of rows' headings (Array)
- Ids of columns' headings (Array)

3.4 Collections statistics generator, similarity explorer

Collection statistics generator produces basic information on the data that is available in the database or digital library. Similarity explorer is a part of utilities that are needed in the context of citation analysis of the identification of two same or similar entities, for example, two articles, or persons, or reviews. Similarity measures can be determined on several reasonable: based on the angle between documents (typical angle-based measure is cosine between vectors), distance based similarity measures (usually Euclidian distance), correlation factors (such as Jaccardi) or probabilistic dependencies, Levinstein distance, etc.

3.4.1 Requirements

The selection of similarity measure procedure depends on the user that must install appropriate packages and use appropriate settings. Similarity explorer is an Add-on for SOPS services and might be used for any services. The purpose is twofold: first, it may assist in removing duplicate or almost identical records or field values, which can be used in the next step (clustering).

3.4.2 Installation

Simply copy the files into SOPS library folder and you will see additional options in the admin interface.

3.5 Clustering services

Clustering service is implemented as a web service. Since clustering procedure might take quite a long time several mechanisms still should be developed to enable real-time use of the service. As described in D14 the purpose of clustering is to: Group similar papers into groups, for example to defined sessions of a conference, provide navigation and browsing through the collection, establish relations between humans and papers, for example for the purpose of assigning reviewers. There are two main sources of data, which may be used to perform clustering techniques: “objective” metadata about papers and textual content as well but on “subjective” user’s use of the papers.

3.5.1 Requirements

There are no specific requirements of the data that is stored in SOPS services since clustering service can easily. No installation is required since the service operates as web service.

3.5.2 Input data

Depending on algorithm user may select different parameters, such as number of clusters user wishes to produce (K- means clustering) or number of levels in the case of hierarchical clustering algorithms.

3.5.3 Procedure and results

Results are returned in HTML form if the services are accessed through browser interface.

3.6 Citation analysis

In the framework of the SciX-project a substantial extension of full text e-prints could be realized for the CUMINCAD-repository. The extraction of references from these e-prints and storage in another digital library - called CUMINCAD.REFS - defined a next working stage. The citation data was divided into four parts and equipped with the CUMINCAD-ID of the specific publication, in order to provide a link to the display. Therefore, a database-user can see how the authorship references on the present publication. A substantial number of references could be extracted for the archived e-prints.

In this section we first give a brief description of the input data that was used for further analysis. Afterwards, methods and tasks that were performed are presented more in detail.

3.6.1 Input data

The data is represented in figures relevant to IR (Information Retrieval). As shown in table 1, there was a significant growth of records and vocabulary in the CUMINCAD database in the period 2002-2004 after the SciX project was initiated (more than 3000 new records were added). As indicated in average term vector length – these newly added records do have more elaborate descriptions; on the other hand the standard deviation indicates that the length of descriptions varies very much.

Table 1. Collection of statistics in the CUMINCAD database

Year	Total number of records (in whole collection)	Number of stemmed terms (in whole collection)	Average term usage (in whole collection)	Average term vector length	Standard deviation of vector length	Average freq. of terms in vector	% of terms with freq. 1
CUMINCAD repository							
2002	3042	11762	14.76	57.39	30.32	1.39	78.66
2004	6399	21255	18.41	61.32	57.00	1.40	78.47
CUMINCAD references							
2004	23988	8550	14.36	5.12	2.22	1.02	98.30

Term vectors are much shorter because CUMINCAD references do not include abstracts, which were taken into account in the case of CUMINCAD records. The next two figures show a number chronological distribution of records (or records per year of publishing) in both databases.

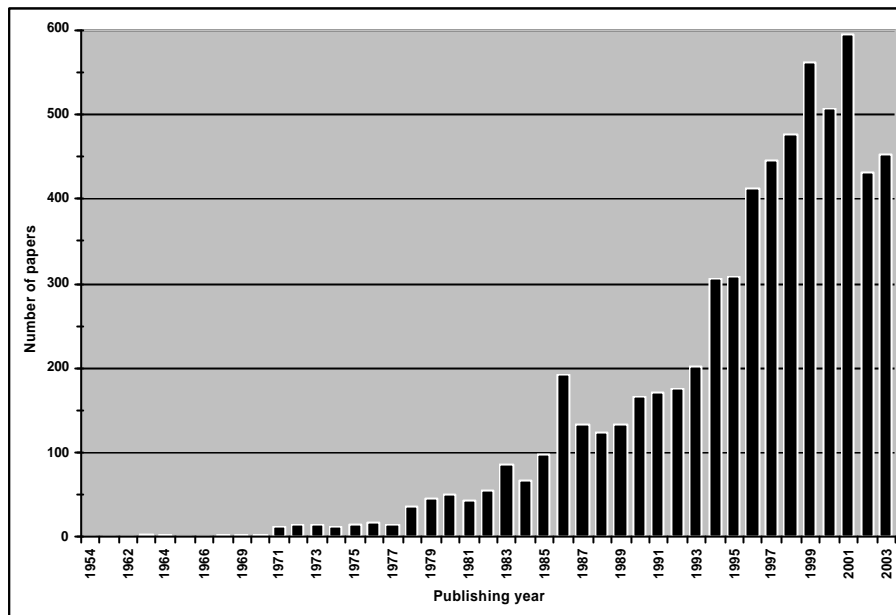


Figure 6: Chronological distribution of CUMINCAD papers

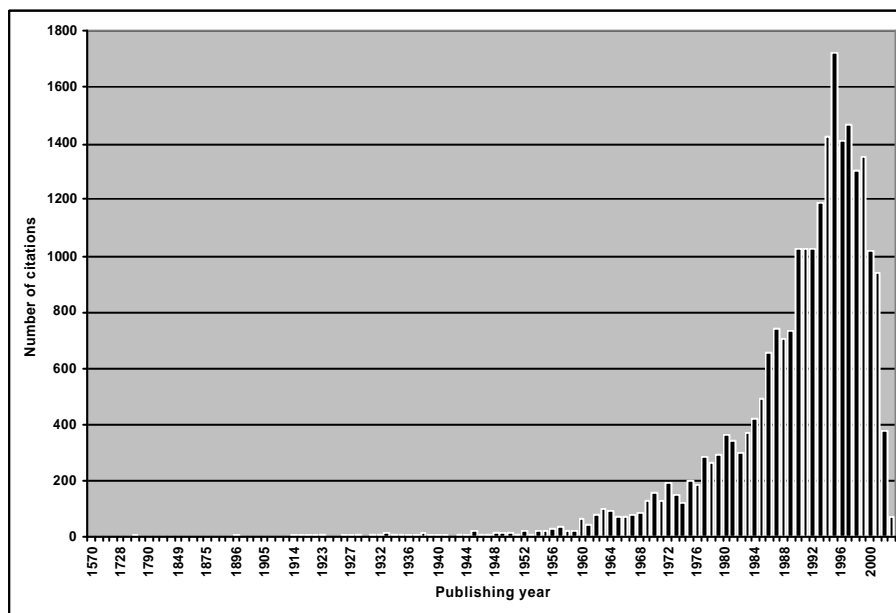


Figure 7: Chronological distribution of CUMINCAD references

3.6.2 Pre-processing of raw data

In order to provide relevant results, several issues had to be solved first. If the input data are not correct and/or properly prepared, or if it is analysed under wrong assumptions, the analysis will not be relevant. Having in mind that the same items (articles) have several records, and/or have different descriptions there were typed by authors who used the references, following two issues were the main concern:

- (1) How to identify identical parts of the fields, whole fields, and whole records that are (or should be) identical in one of the two databases?
- (2) How to identify typing errors that are not spelling errors, and how to identify, add, or/and change missing or wrong data automatically?

Both issues are somehow related and require similar approach: (1) they cannot be solved with just one single solution in one step – the process is iterative; (2) more than one measure of similarity must be used. The first step in the analysis procedure was creation of similarity matrix based on different length measures of field values, to Euclidian distance, and angle based distances, to asymmetrical Levenstein or edit distance.

The problem of identifying errors is more complex since it includes more details, and it depends on human factors. Most of the problems were related to the database of references, which does include duplicates of records that have quite different descriptions. In order to be able to decipher paths through references it is extremely important to map and assign the same identifiers to the entries in the database of references as well as to the entries in the main CUMINCAD database.

The selection of canonical form that would uniquely identify authors was quite straightforward:

- For example, variations of descriptions: "Kalay, Yehuda E.", "Kalay Y E ", and "Kalay, Y.E. " are identified as Kalay.Y.E.
- Multiple authors like "Zilles, S.N., Lucas, P., Linden, T.M., Lotspiech, J.B. and Harbury, A.R." would be transformed into array with five elements: (1) "Zilles.S.N."; (2) "Lucas.P."; (3) "Linden.T.M."; (4) "Lotspiech.J.B."; (5) "Harbury.A.R.".
- Missing or incomplete author data was automatically detected from related records based on associated fields. Identified inconsistent use of authors' names was additionally improved through mapping of names to main (most frequent) identifier.

Next important issue in the context of citation analysis of sources is the identification of two same or similar articles. For example, two articles may be considered the same or similar if: (1) title, year and author(s) and other are exactly the same; (2) author(s), title, etc. are almost the same; (3) they have the same textual content; (4) they have the same citations; (5) they have the same usage. Based on preliminary analysis and available metadata following combinations were used to determine paper identifier.

- Canonical form of the author(s) + year + similar title, different distance measures were used to determine similarity;
- Publishing year + extracted words from the title, extraction was based on simple elimination of stop words;
- Canonical form of the author(s) + extracted words from the title
- Missing or incomplete title data was replaced with data from related records if at least first author and year matched and if the normalized distance measure was relatively high (at least 80 %).

3.6.3 Processing of references

One of the main purposes of automated analysis is to detect similarities. The similarity depends on the definition of equality – that is – with the definition of canonical form. The aim was to create such canonical form that can serve at the same time as identifier. The processing of data was primarily focussed on the relationship between references, as well as preparation of the results in the form, which would be useful for further analysis. Therefore, following major steps were carried out:

1. Select analysis parameters based on preliminary statistical data;
2. Eliminate sources of errors and assure that assumptions are satisfied;
3. Establish relationships between papers;
4. Evaluate relationships;
5. Visualize the results in the form of graphical models.

3.6.4 Results

In this section results of the analysis will be presented. First, we give an overview of quantitative analysis where we address two interesting questions about referencing habits: (1) How many references do author use, and (2) How "old" are the references. Second, some qualitative results are covering specific relations between chronological developments, with regard to topics and authors illustrating the growth of scientific community and specific citation related topics.

3.7 Quantitative analysis of referencing habits

The paper with the highest number of references (70 - in words: seventy) is (Bermudez et. al., 2000). However, the average number of references per paper is around 10.4. A Distribution of number of papers related to the corresponding number of references is presented on the figure below.

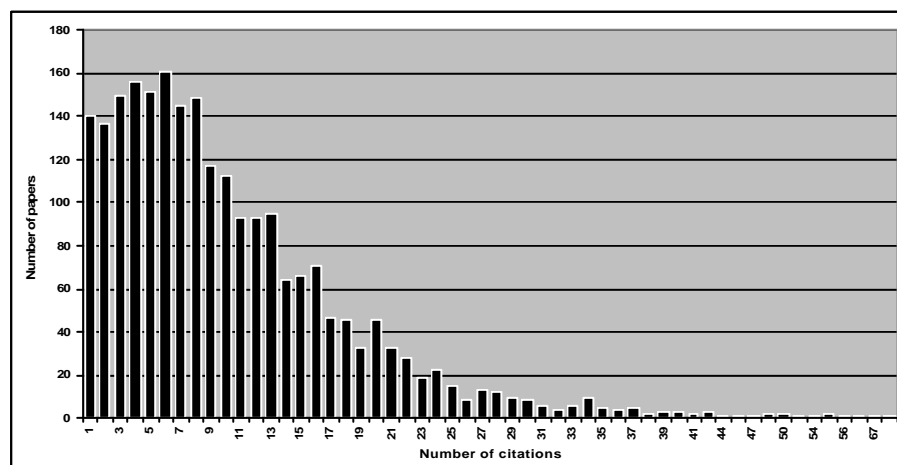


Figure 8: Frequency of citations

The analysis shows that 60% of the papers provide 1-10 references; around 30 % of the papers have 10-20 references; a bit more than 7 % have 20-30 references and less than 3%

attach more than 30 references. A distribution of citations in Figure 9 is presented in a way to be comparable with analysis presented at (Open Citations project, 2002).

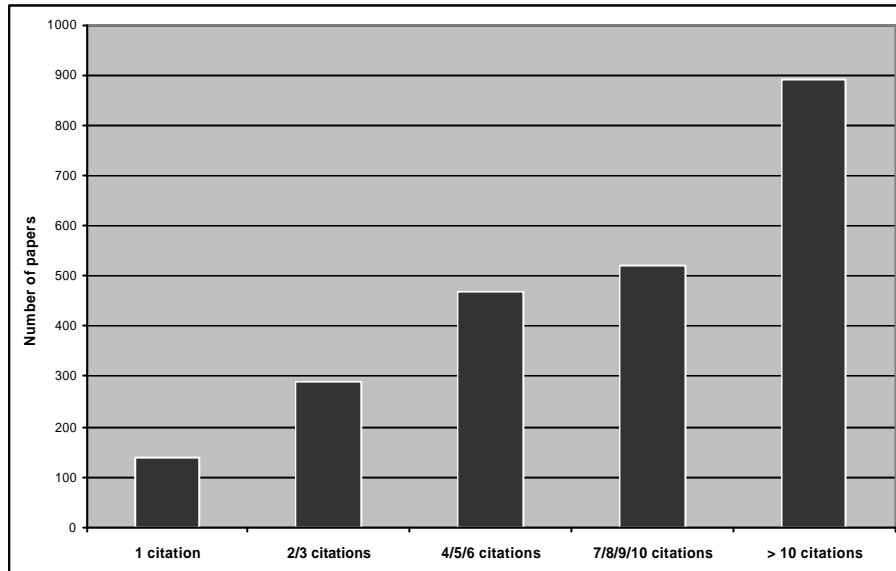


Figure 9: Distribution of citations

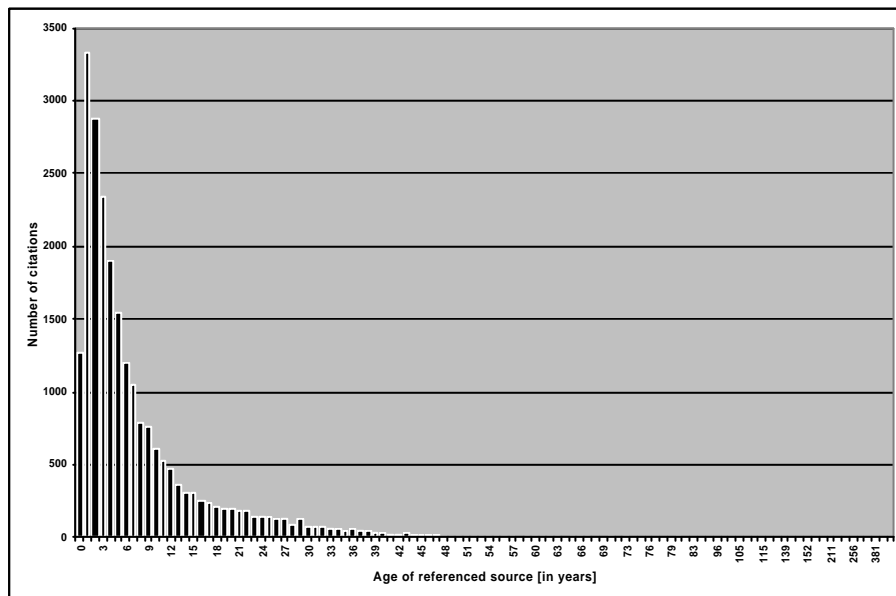


Figure 10: Age of citations

The age of citation may show the growth of interest in a particular topic, or in general – the speed of knowledge exchange. The relationships between citations in the chronological sense may be represented in the form of year-to-year citation networks. An example of such network representing citations younger than seven years is presented in figure 7.

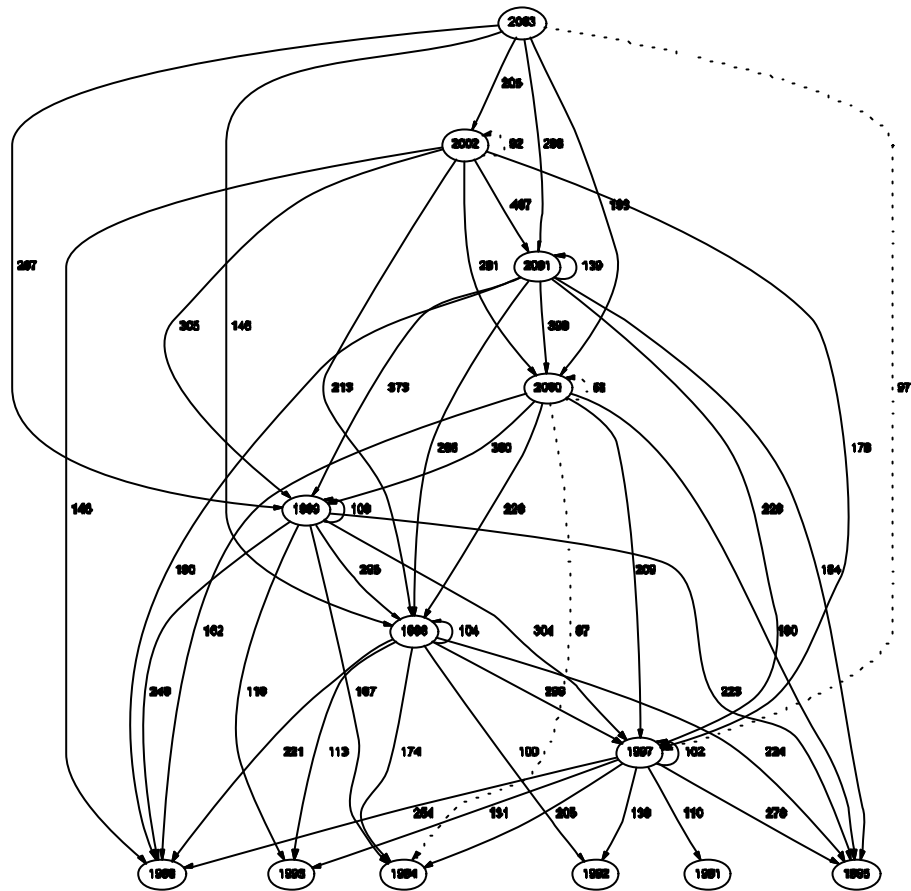


Figure 11: Year-to-Year relationships between citations

The vertices in the graph (Figure 11) represent all papers that were published in a certain year. Directed edges from vertex i to vertex j denote that papers published in year labelled on vertex i have referenced papers that were published in year labelled on vertex j . These edges are visualised in three styles and reflect a number of year-to-year citations (dotted < 100 , thin > 100 , and thick lines > 200). Labels on edges represent the exact number of references.

A normalized view on the age of references is shown in the figure below.

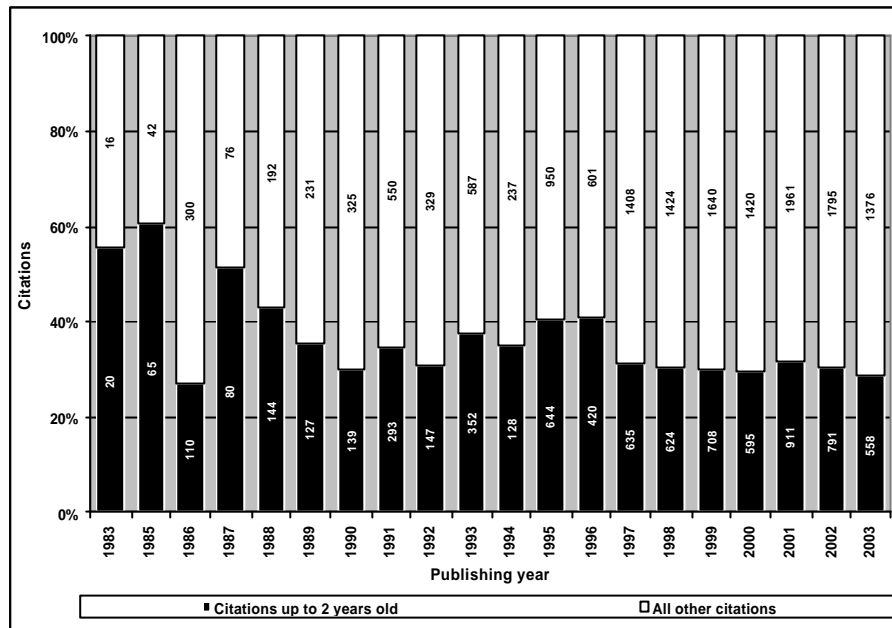


Figure 12: Distribution of citations used in papers published in the last two decades

3.7.1 Analysis on authors of cited material

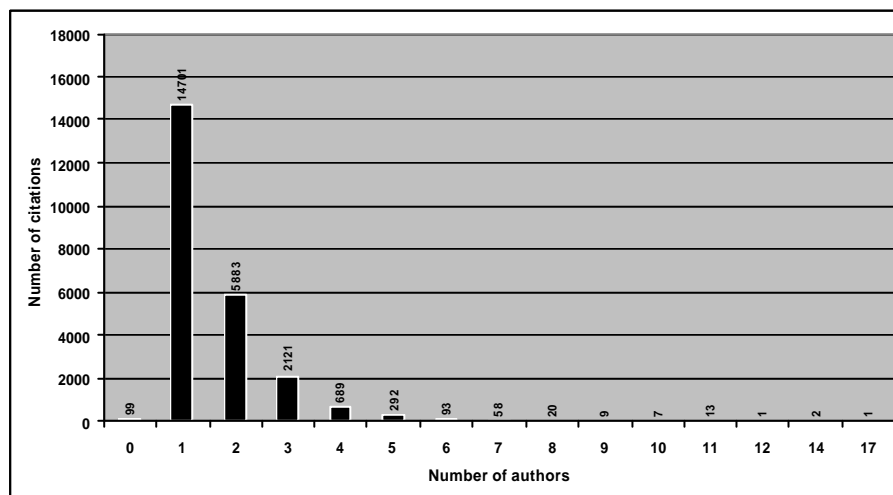


Figure 13: Distribution of citations per number of authors

There is just a small number of unknown sources (99); from the remaining part most of the references (60 % or 147001) were written by a single author, 25 % of the referenced papers were written by 2 authors, a bit less than 10 % by three, and 4 or more persons wrote less than 5 % of cited papers. The cited paper with the highest number of authors (17) is (Akin et al, 1997).

About 12500 different individual authors were authors of 24000 cited sources, precisely: in average each author was cited 1.92 times. How frequently a graphical representation of number of citations of twenty-five most popular authors is shown on Figure 13. These authors cover about 20 % of citations in the CUMINCAD database of references. About half of most

cited papers were written individually, and equal shares (25 %) represent number of papers that were written by most cited authors whether as first authors or as co-authors.

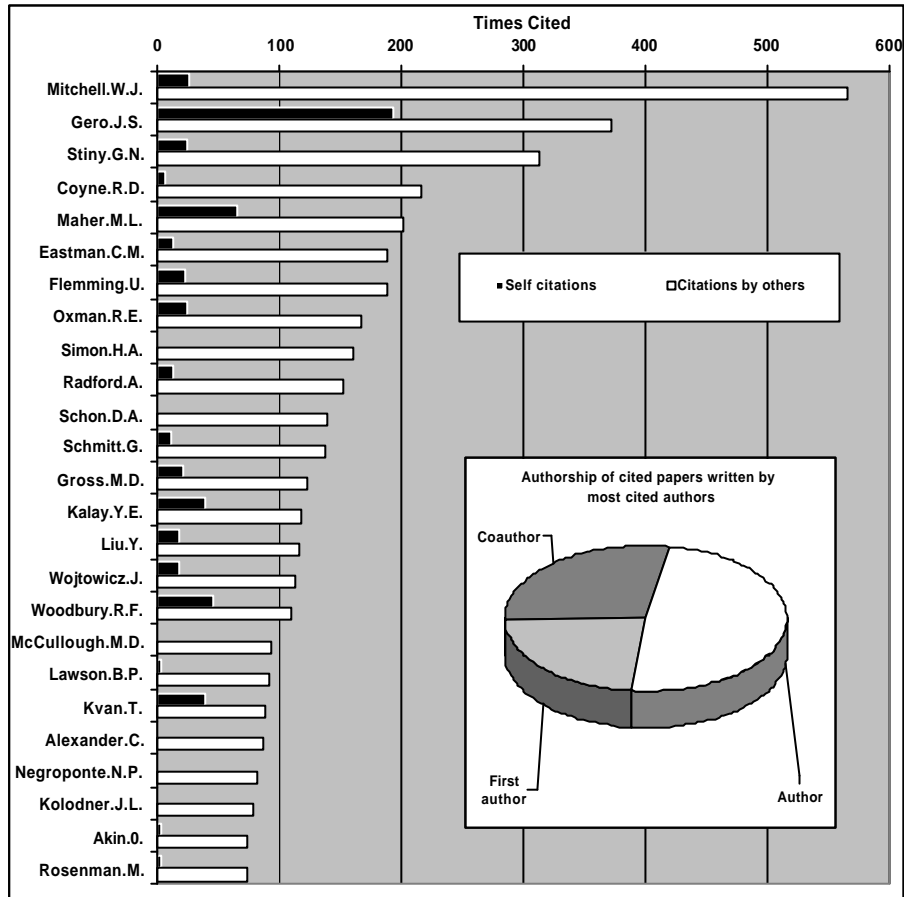


Figure 14: Top 25 most cited authors

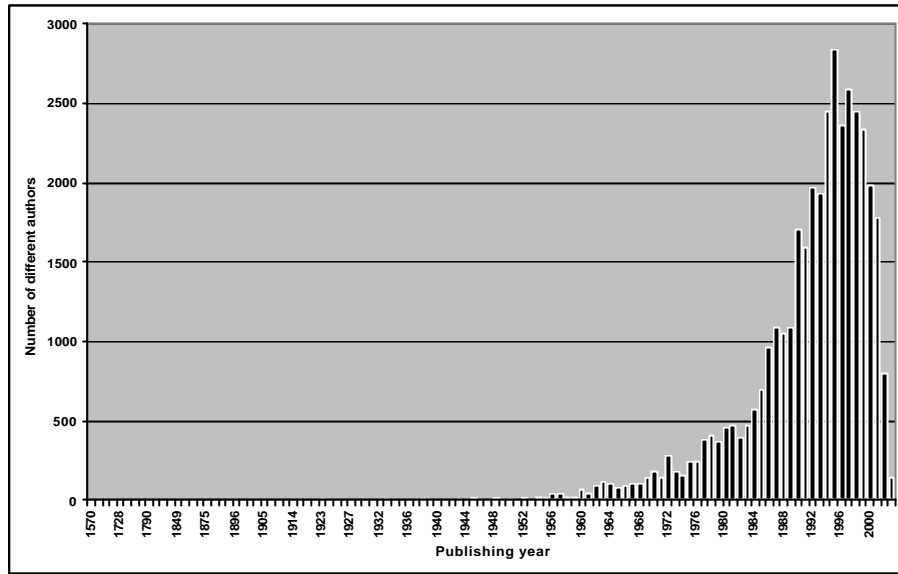


Figure 15: Number of authors per year

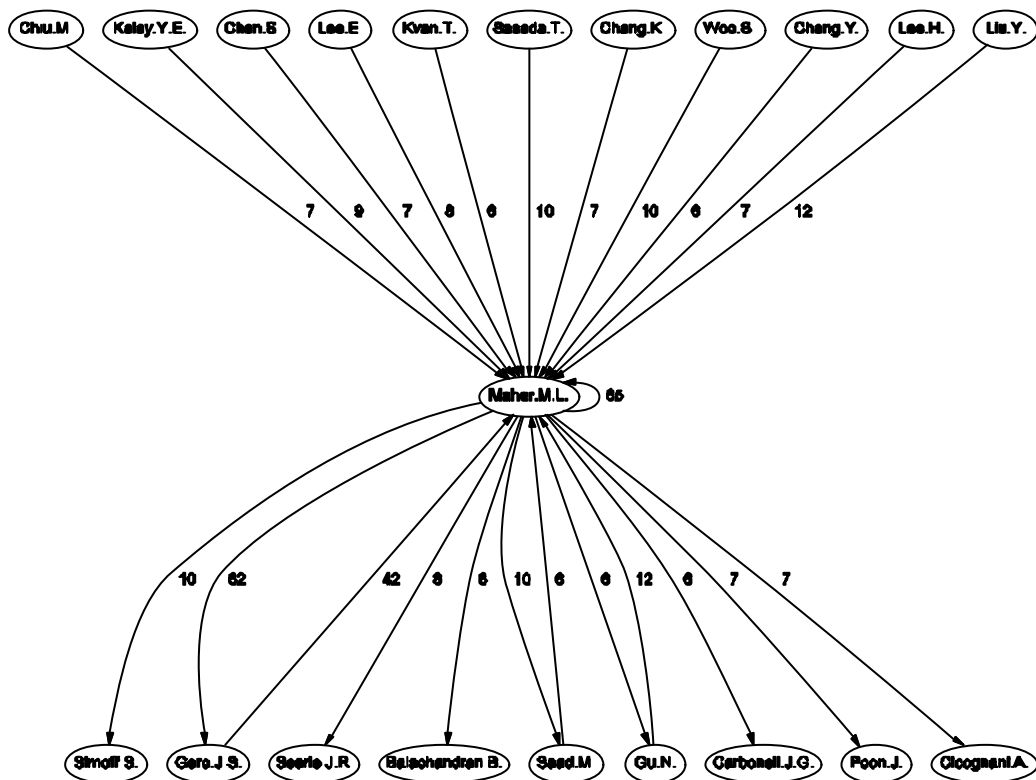


Figure 16: Example of citation-network

The following two graphs illustrate related scientific efforts, and collaboration, as a further exploration of authors' activities that were included in the network visualized in Figure 16.

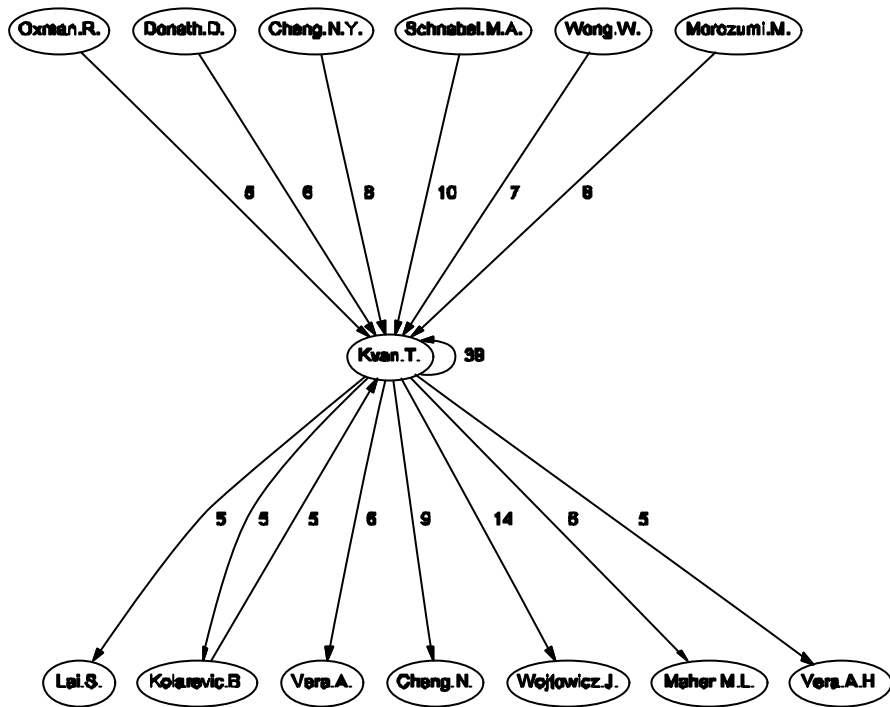


Figure 17: Example of author-citations network

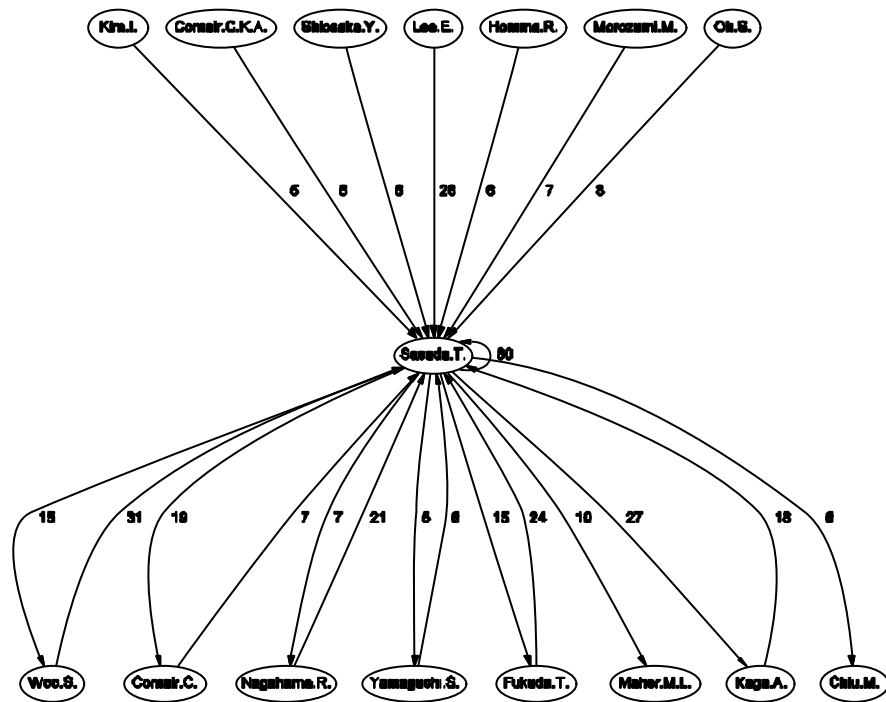


Figure 18: An example of a well established network

The results could be integrated into CUMINCAD profile (see Figure 19).

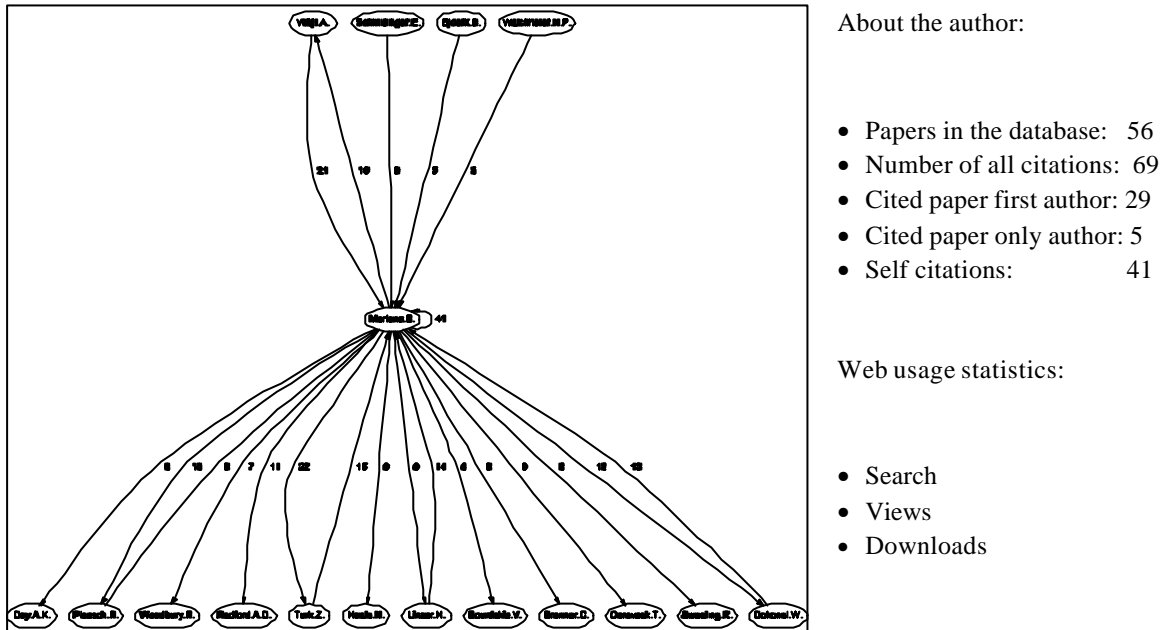


Figure 19: An example of authors' profile details repository feature

3.7.2 Discussion on citation analysis

Citing habits. Paper gives an insight into citing habits illustrated on *Figure 8: Frequency of citations* and *Figure 9: Distribution of citations*. A comparative analysis of citing habits clearly indicates that CAAD community uses many more references compared to other fields presented in (Open Citation project, 2002). The conclusions could be twofold: CAAD research is much broader - multidisciplinary, and that it therefore requires a broader overview on studied topics - since citing reflects literature study. A complementary analysis on web usage statistics would offer additional information on specific articles. Namely, citations only provide information about the articles that were cited and not about the content that wasn't. That is made possible through the use of digital libraries, since we can gain more information on papers' usage – i.e.: they have been read, abstracted viewed, or their full-texts have been downloaded but they were never cited. Embedding such relationships into the digital libraries' functionality would give an added value and completely new view to the impact of scientific works.

Chronological view of scientific information exchange. - From *Figure 5: Age of citations*, we can conclude that most scientific ideas evolve in the period of 2 years. Although the specimen might be considered too small, observation of *Figure 11: Year-to-Year relationships between citations*, and *Figure 12: Distribution of citations used in papers published in the last two decades*, as well as additional analysis of papers containing citations, we can claim that importance of digital libraries is evident and that they highly influence the evolution and directions of research topics, and their transfer.

Scientific collaboration networks. From the *Figure 14: Top 25 most cited authors* it is evident that the most influential authors usually write papers solely or at least they are the first author. This paper describes how networks between papers (and especially their authors!) in the area of Computer Aided Architectural Design can be identified with the help of graph theory. Often two or more authors are connected as illustrated on *Figure 16*, and *Figure 17* (people that are in focus of the networks, or have connections happens to be on the list of most cited authors), or they build self-sustained networks as shown on *Figure 18*. From *Figure 11: Number of authors per year*, one could claim that number of authors that are cited is constantly falling since 1996, but it is not so – the number depends on age of cited papers (*Figure 10*).

Exploitation of the results. First of all, an "enabler" for scientific collaboration has to be mentioned as the developed mechanisms heavily support the task of finding right persons (as potential collaborators in the future). research by means of a well developed Citation Index and measurement of Impact Analysis would work out well for CAAD as well. Repeating research already undertaken by others and the failure to evaluate is nothing new. And last, but not least, it is important to emphasise that possible use of the results in the form illustrated on *Figure 19* would significantly improve the scientific digital-library end-user experience. With the help of smart user interfaces it would enable easier literature study as well as establishing of social networks. This will enhance representations (keywording system, classification of papers, for optimal access, browsing and search routines), which will be more easily generated with the help of text mining, clustering and usage analysis techniques. The results could significantly improve the precision of the search results

4. SCIX HARVERSTER

SciX is an OAI-compliant repository and as such allows the metadata within its repository to be extracted or harvested by external sites. This section describes the harvester that harvests OAI compliant repositories and feeds the data in to the SciX 'works' service.

4.1 Definitions

The following definitions are found on the Open Archives Initiative for Metadata Harvesting (OAI-PMH) site.

4.1.1 Harvester

A harvester is a client application that issues OAI-PMH requests. A harvester is operated by a service provider as a means of collecting metadata from [repositories](#).

4.1.2 Repository

A repository is a network accessible server that can process the 6 OAI-PMH requests. A repository is managed by a data provider to expose metadata to [harvesters](#). To allow various repository configurations, the OAI-PMH distinguishes between three distinct entities related to the metadata made accessible by the OAI-PMH.

- *resource* - A resource is the object or "stuff" that metadata is "about". The nature of a resource, whether it is physical or digital, or whether it is stored in the repository or is a constituent of another database, is outside the scope of the OAI-PMH.
- *item* - An [item](#) is a constituent of a repository from which metadata about a resource can be disseminated. That metadata may be disseminated on-the-fly from the associated resource, cross-walked from some canonical form, actually stored in the repository, etc.
- *record* - A [record](#) is metadata in a specific [metadata format](#). A record is returned as an XML-encoded byte stream in response to a protocol request to disseminate a specific metadata format from a constituent item.

4.1.3 Item

An item is a constituent of a repository from which metadata about a resource can be disseminated. An item is conceptually a container that stores or dynamically generates metadata about a single resource in multiple formats, each of which can be harvested as [records](#) via the OAI-PMH. Each item has an [identifier](#) that is unique within the scope of the repository of which it is a constituent.

4.1.4 Unique Identifier

A unique identifier unambiguously identifies an item within a repository; the unique identifier is used in OAI-PMH requests for extracting metadata from the item. Items

may contain metadata in [multiple formats](#) . The unique identifier maps to the item, and all possible [records](#) available from a single item share the same unique identifier.

The format of the unique identifier **must** correspond to that of the [URI \(Uniform Resource Identifier\)](#) syntax. Individual communities **may** develop community-specific [URI](#) schemes for coordinated use across repositories. The scheme component of the unique identifiers **must not** correspond to that of a recognized URI scheme unless the identifiers conform to that scheme.

Unique identifiers play two roles in the protocol:

1. *Response*: Identifiers are returned by both the [ListIdentifiers](#) and [ListRecords](#) requests.
2. *Request*: An identifier, in combination with a [metadataPrefix](#) , is used in the [GetRecord](#) request as a means of requesting a [record](#) in a specific metadata format from an item.

Note that the identifier described here is *not* that of a *resource*. The nature of a resource identifier is outside the scope of the OAI-PMH. To facilitate access to the resource associated with harvested metadata, repositories **should** use an element in metadata records to establish a linkage between the record (and the identifier of its item) and the identifier (URL, URN, DOI, etc.) of the associated resource. The mandatory Dublin Core format provides the `identifier` element that **should** be used for this purpose.

4.1.5 Record

A record is metadata expressed in a single format. A record is returned in an XML-encoded byte stream in response to an OAI-PMH request for metadata from an item. A record is identified unambiguously by the combination of the [unique identifier](#) of the item from which the record is available, the `metadataPrefix` identifying the metadata format of the record, and the [timestamp](#) of the record. The XML-encoding of records is organized into the following parts:

- *header* -- contains the unique identifier of the item and properties necessary for selective harvesting. The header consists of the following parts:
 - the [unique identifier](#) -- the unique identifier of an item in a repository;
 - the [timestamp](#) -- the date of creation, modification or deletion of the record for the purpose of [selective harvesting](#).
 - zero or more [setSpec](#) elements -- the [set](#) membership of the item for the purpose of [selective harvesting](#).
 - an **optional** `status` attribute with a value of "deleted" -- indicates the withdrawal of availability of the specified metadata format for the item, dependent on the repository support for [deletions](#).

- *metadata* -- a single manifestation of the metadata from an item. The OAI-PMH supports items with multiple manifestations (formats) of metadata. At a minimum, repositories **must** be able to return records with metadata expressed in the [Dublin Core](#) format, without any [qualification](#). Optionally, a repository **may** also disseminate other formats of metadata. The specific metadata format of the record to be disseminated is specified by means of an argument -- the [metadataPrefix](#) -- in the [GetRecord](#) or [ListRecords](#) request that produces the record. The [ListMetadataFormats](#) request returns the list of all metadata formats available from a repository, or for a specific item (which can be specified as an argument to the [ListMetadataFormats](#) request).

4.2 SciX Harvester

4.2.1 SciX Harvester - OUT

As explained in the previous section SciX is OAI-compliant and its metadata is harvestable. The software has been tested using routines based upon the offering from OCLC. This consists of invoking a routine with the base-url of the SciX repository as parameter. The metadata is extracted on-line or can be written to a file. A typical command is :-

```
Java -classpath <CLASSPATH> <java routine> -out <OUTFILE> http://<Base URL of SciX repository>
```

4.2.2 SciX Harvester - IN

SciX is capable of assimilating metadata harvested from OAI-compliant repositories or indeed from other sources. SciX will accept a comma-delimited file of data and post that file to its own repository. An example of a file is :-

```
Id;authors;year;title;source;summary;series
```

```
8202008;Katz,Sheldon;2003;Rational Curves...;Math Journal;a summary of the article;USAL1
```

```
8202009;an author,2004;a title; a source;a summary;USAL1
```

Interfaces to the OCLC software have been developed in order to harvest data from a selected repository based upon the following:-

Topic eg Maths, Physics

Date . from and to date range.

4.2.3 OAI-compliant Repositories

All repositories which are OAI-compliant have a baseURL which the harvester specifies to gain access to the data. A list of registered OAI repositories is held at <http://www.openarchives.org/Register/BrowseSites.pl>

5. VALUE-ADDED PUBLICATIONS

5.1 Introduction

Value Added Publication (VAP) is a SciX application that demonstrates how SciX content may be used in vertical markets or subject domains (figure. 20). The aim of a VAP application is to provide the normal industry user with access to industry specific articles that are created by VAP operators (e.g. editors and publishers) based on and edited from SciX digital archive content. These articles in form of digests, reviews and summaries are more suited to industry specific requirements and thereby extend the reader base of scientific publications. The section below outlines some of the main characteristics of a SciX VAP application:

- The Operators of a VAP application are typically industry organizations, federations and associations, commercial information providers or special interests groups
- VAP moderated articles are targeted to industry practitioners while content in SciX Digital Libraries are more specific to the scientific and academic community

VAP applications can cooperate in a peer-to-peer network to disseminate, aggregate or re-use content created by one VAP in another VAP. The logical architecture of the VAP handles content creation and content presentation as separate concerns. For example a content author (an expert) that writes about a specific domain of knowledge can make his content available to several VAP content consumers for presentation. Similarly a content consumer (e.g. an association) can aggregate and re-publish domain specific content from several independent VAP publishers to a single audience group.

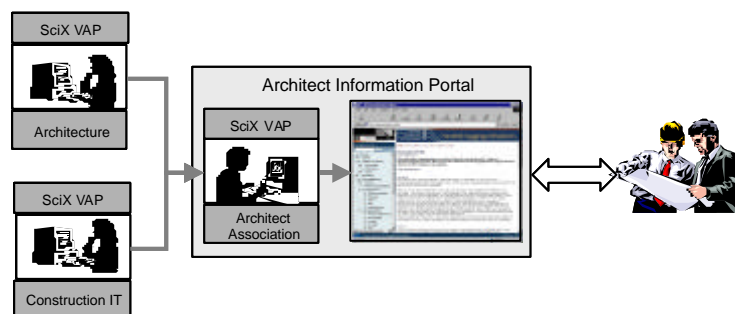


Figure 20: SciX VAP content aggregation

Figure 20. Illustrates an example of the VAP publishing model. The Architect Information Portal (AIP) is subscribing to content from two independent subject specific services, the architecture subject service and the construction informatics subject service. The content is

aggregated by the AIP's own SciX VAP and made available to the portal software for publication to end-users.

5.2 Architecture overview

An overview of the VAP architecture is shown in figure 21. The SciX VAP architecture is designed to promote flexible use case scenarios and a widely accessible system in a peer-to-peer arrangement suitable for handling varied needs of industry community's and end-user requirements.

Components of the VAP application are shown in coloured boxes (figure 21.) while external applications are shown white. The architecture is based on the principles of separation of concerns; a) content creation and content presentation are handled by external applications and b) content management and content delivery are handled by VAP applications. In other words the VAP application makes no assumptions about the tools used for creating a VAP article or the format in which it is stored, nor the method or technology by which it will be presented to the end user as demonstrated in Figure 21.

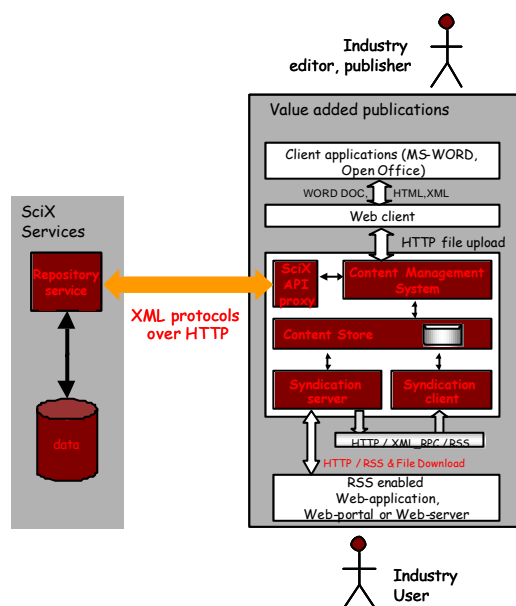


Figure 21: The SciX VAP software

The main components of the VAP architecture are three types of applications, a Content Management system, Syndication Server and Syndication Client and an interface proxy, which provides access to the SciX digital archives.

SciX Interface proxy (SIP). The SIP handles the interaction of a VAP application with the SciX repository service. The repository service provides access to metadata and full text of scientific papers in a Digital archive. The SIP uses the repository Dublin Core compatibility to provide meta-data about scientific papers.

VAP editors use the SIP to search the digital archives, to read and upload papers, produce citations and bibliographic notes. More specifically the SIP provides functionality for

VAP editors to:

- Do keyword search or build detailed searches for scientific publications based on numerous parameters
- Download scientific publication full text and metadata for building citations and bibliographic notes
- Bookmark repository scientific publications

The SIP contains a registry with digital archives from a range of available sites that is available to users to select from.

Content Management System (CMS). The CMS provides the support for maintaining edited articles, collaborative authoring and versioning and publication of articles to end-users.

The CMS provides two user interfaces (UI), one for authors and the other for managing the CMS. The author UI is called “Workspace Manager” (WM). In the WM the authors manage their private article repository, user access permissions and publication of articles. Similarly the CMS manager UI manages CMS system central resources e.g. users, roles, access privileges and the public repository that can be access by all users.

The CMS includes a logical repository structure consisting of nested collections and files similar to OS file systems, locking mechanism to implement checkout/checkin functionality for collaborative authoring and centrally managed access control to namespaces, collections and files. The VAP repository consists of three types of namespaces, an owner private namespace where users store their personal articles and documents, a project namespace where sharable articles are stored and a public namespace where syndicated content feeds and downloaded public documents are stored.

Syndication Server. The Syndication server consists off a Manager Web application and an HTTP XML-RPC server application. The Manger application handles the management of content delivery. The Content delivery is subscription based and automated by a syndication schedule (e.g. when, where, what). The Syndication server constantly scans the CMS repository for published updates and new articles. When such updates are available the Syndication server sends the designated client a notification of an available update in the form of an XML-RPC request. Notifications of updates are formatted as RSS channel headlines. Upon receiving a request to download the clients downloads the published RSS channels containing the information about the new and updated articles and optionally uploads the actual full text articles for publication to end-users or further processing.

Syndication Client. The Syndication Client is HTTP server application in the P2P network that listens for XML-RPC requests. When updates are available the syndication server sends the client a notification in an XML-RPC request. The client responds by calling an XML-RPC method on the Syndication server to upload a RSS channel that contains information on updates. The syndication client can then, depending on the VAP configuration, store the RSS channel in the CMS repository and/or store the RSS channel to a file where it can be picked

up by any RSS enabled application for presentation. Depending on preferences the client can optionally retrieve the actual articles being advertised in the RSS channel from the Syndication server by successive XML-RPC method calls and persist them in the CMS public repository namespace.

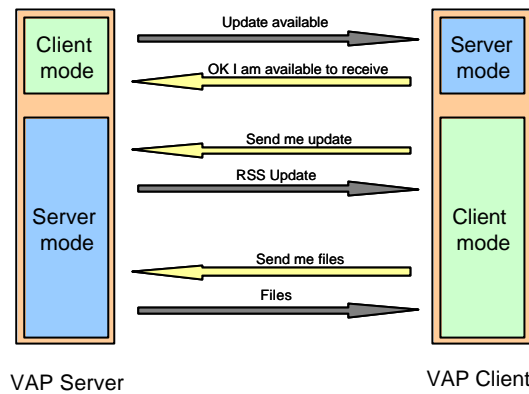


Figure 22: The SciX XML-RPC protocol

5.3 Deployment

The SciX VAP framework is based on a syndication model. The SciX VAP exist in a peer-to-peer network where the content delivery mechanism allows communication of content between independent SciX VAP applications and publication applications like Web servers.

The VAP applications have been designed to allow greatest flexibility in deployment. The table below shows how applications can be deployed on the same host to provide different level of functionality in a publishing environment.

Deployment / application	CMS	SS	SC	Remarks
Content Management	X			The CMS is a standalone content management system and may be used in this configuration to create a web enabled central file repository and collaborative authoring environment.
Publisher	X	X		Using the Syndication server in conjunction with the CMS enables syndication of RSS headlines of updated and new content to any Aggregator on the network.
Headline Aggregator			X	The Syndication client can be configured in a standalone configuration to receive RSS headlines form any publisher. When in this configuration, RSS channels and optionally downloaded files are normally saved into the file space of a web application, which makes it available to the end-user
Content	X		X	In this scenario the received RSS channels can be collected

Aggregator				in the central CMS along with downloaded files. A management interface enables the site manager to control more precisely how, when and what content is eventually made available the end-user.
Publisher / Aggregator	X	X	X	This scenario uses one central CMS for aggregating content form VAP publisher, authoring and publishing content to other Aggregators. Aggregated content can be republished, used for creating new content by authors or displayed to the end-user.

CMS: Content management system, SS: Syndication server, SC: Syndication client

Figure 23. demonstrates a typical implementation scenario of two VAP applications syndicating content on separate systems.

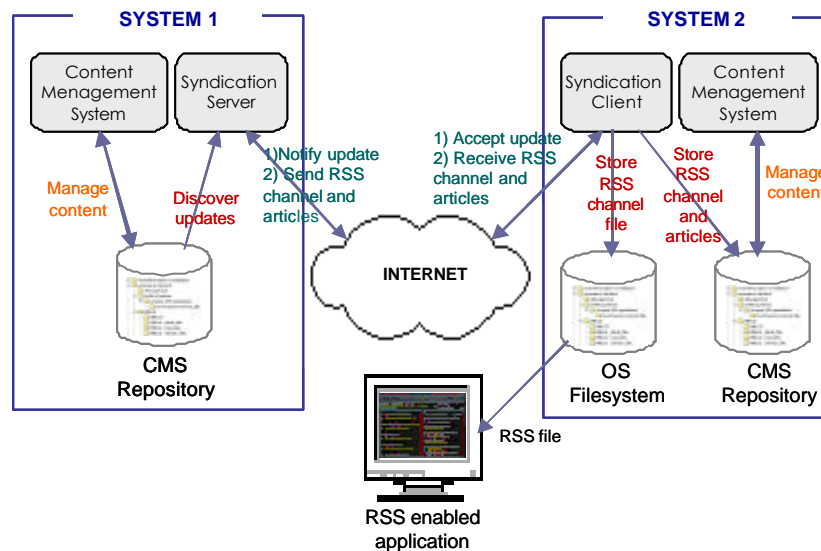


Figure 23: VAP application cooperation

5.4 Installation and configuration

5.4.1 Content Management System (CMS)

5.4.1.1 Installation requirements

The CMS requires two applications to be installed on a server before installation.

- Web application server e.g. Apache Tomcat or equivalent
- MySQL RDBMS or an equivalent database system e.g. Oracle or MS SQL

5.4.1.2 Installation

The CMS package is distributed as a single war file (web application archive). The following installation instruction applies to the Apache Tomcat servlet container. Please refer to the documentation supplied with your application server (e.g. JRUN, BOSS etc) for installing a War file.

The simplest way to install the web application is to drop the war file into the webapps directory of the Tomcat directory tree and restart the server. If you can't bring down the server you can use the administrative interface to install the web file. Use the war parameter to specify a URL for either a directory or a web application archive (WAR) file to install a web application context.

Example: if the ".war" file exists in the /path/to/scixvap.war on the Tomcat server it will be installed as the web application context named /scixvap. Notice that there is no path parameter so the context path defaults to the name of the web application archive file without the ".war" extension. If you need to use a different context name add the path parameter e.g. ...?path=/footoo&war= ... to the URL.

```
***http://localhost:8080/manager/install?war=jar:file:/path/to/ scixvap.war!/***
```

After Tomcat installs the CMS application, check the web application configuration file and make any necessary adjustments.

```
*** {Tomcat_home}/webapps/scixvap/web-inf/web.xml
```

5.4.1.3 Configuration

The CMS distribution comes pre-configured. Depending on your requirements you will need to make changes to the distribution configuration files in the directory {Tomcat_home}/webapps/scixvap/WEB-INF/conf as explained below.

- scixvap.properties, specifies the operational context of the CMS and will adjust the user interface accordingly. e.g. if the operational context is together with a syndication server this property will add the syndication server management interface to the CMS menu structure and similarly for the syndication client.
- torque.properties, specifies the underlying database system used by the CMS (please refer to the <http://db.apache.org/torque/> web site for supported RDBMS and how to configure torque) The CMS distribution comes pre-configured for mySQL and contains empty databases that can be dropped into the mySQL RDBMS without any additional configuration. SQL template schemas are available in the directory {Tomcat_home}/webapps/scixvap/WEB-INF/db to create database tables and initialize the essential database tables. These are compatible with the mySQL RDBMS and might need some minor modification to work with other database systems.
- CmsResources.properties will always have to be modified as it contains the physical path to where the CMS document root is in the hard disk directory tree. Similarly it contains the physical path to the Domain.xml file below that will need to be modified as well.

- Domain.xml; the domain file describes the structure of the Slide CMS which is the underlying software for the CMS. This file should normally not need any modification.

5.4.2 Syndication Server

5.4.2.1 Installation requirements

Before attempting to start the server the CMS needs to have been installed before.

5.4.2.2 Installation

- Download the distribution from the web site to a empty directory
- Unpack the ZIP file

Depending on the distribution downloaded, binary or full source the following directory structure is created that contains the full distribution.

/bin – Wrapper and scripts for (un)installing the server as a Windows service

/build – Ant build scripts, configuration files and byte code classes

/conf - configuration files

/lib - contains all jar files used in project

/logs – log files generated by the server

/src - the source file tree

/torque – SQL schemas used for building and initializing database tables

/db – Synserver MySQL database

The Syndication server is a Java application and works on server platforms that have the Java Virtual Machine.

Apache java-based build tool, Ant is required to run the server from the command prompt as well as to set up the server databases.

The Syndication server may be installed on NT, Win2000 and Win2003 servers, as a Windows Service. For more information see section 6.3.4

5.4.2.2.1 *Testing the installation*

Starting the application form the command prompt does testing if the installation was successful.

Navigate to {app_home} directory

Make sure you already have Ant installed. Start the server by: > ant runServer. This will start the Syndication Server in the command window. Following is the expected output.

If the output that appears on the screen ends like:

```
runClient:  
  
[JAVA] Torque init done!!  
  
[JAVA] SyndicationServer initialized  
  
[JAVA] Listening on port {port_number}
```

The server was successfully started, otherwise if any errors were reported during the startup of the server re-check the installation and configuration again and make sure all paths are correct.

5.4.2.3 Configuration

For the server to work properly some application file physical paths need to set in relation to the installation directory. The configuration files can be found in the {app_home}/conf directory.

SyndicationServer.properties

syndicationserver.port={port number}, should only be modified if the default port number isn't available.

torque.properties

Refer to section 6.2.3.2 for information to set properties for Torque databases.

SynServer_wrapper.conf

wrapper.java.command={physical path to the java application}, e.g. c:\j2sdk\bin\java

5.4.2.4 Getting started

Now that the server is installed, but before getting it started, a database must first be set-up that the Syndication server uses to maintain information about syndication clients. The distribution comes with a ready MySQL database, which is the default, contained in the {app_home}/tdb directory. If MySQL hasn't already been installed, download the MySQL database from <http://www.mysql.com> and install according to the instructions on the site. If you need to rebuild the database or prefer to use a different RDBMS proceed as follows:

Before proceeding, make sure you have set the torque.properties correctly (see configuration)

Navigate to the {app_home}/torque directory

```
run >ant create-db
```

To add in some test data to the database,

Run >ant create-init-server

If all steps have been taken, the Syndication Server should be up and running.

5.4.3 Syndication Client

5.4.3.1 Installation

The same installation instructions apply for the Syndication client as for the Syndication server. Please refer to section 6.3.3.1. and for testing installation of the client application use > ant runClient.

It the Syndication client as a standalone server or in conjunction with a Vap Content Management System. When in a standalone mode (Headline Aggregator) the client is only capable of saving syndicates RSS headline to the OS filesystem, whereas in conjunction with a CMS (Content Aggregator) can both save the syndicated content in the OS filesystem and/or in the CMS. Please see the configuration section, property “rsschoice” on how to configure the server in different modes.

5.4.3.2 Configuration

Before starting the client application it needs to be configured to operate in the VAP publishing environment. Configuration is done by setting properties in the SyndicationClient.properties file in the {app_home}/conf directory.

syndicationclient.port	The Syndication client IP port number. Must be above 1024
syndicationclient.username	The username a client uses in communication with ser
syndicationclient.password	The password a client uses in communication with server
syndicationclient.confirmValue	The confirmation value used to authenticate the client as a peer in the VAP publishing environment
syndicationclient.downloadFolder	Directory path to which downloaded articles are saved to.
syndicationclient.rsschoice	Setting of rsschoice property determines how the received RSS content is saved. The available options are: <ul style="list-style-type: none"> • none, inactive mode, does no attempt to save received content • db, saves the received content in the CMS only (this option is only available when operated in Content Aggregation configuration)

	<ul style="list-style-type: none"> file, saves the received content to the OS file system in the directory path specified by the and downloadFolder and rssDownloadFolder respectively both, sets the option to db and file
syndicationclient.rssDownloadFolder	Directory path to which the RSS channel file is saved to.
torque.properties	The physical path to the torque properties file, default is {app_home}/conf/torque.properties

Other configuration files are described in sections 6.4.2.3 Installing the Server as a Windows service and 6.2.3.2 configuring database properties (torque.properties) which only applies when the Syndication server is configured as a Content Aggregator.

5.4.4 Installing servers as a Windows service

To install the Syndication server and client as a Windows service do:

- Check the service wrapper, this is done by running the {app_home}/bin/SynServer.bat or SynClient.bat files. If no error messages are displayed then everything should work fine
- To install the service run {app_home}/bin/InstallSS-NT.bat for the Server and /InstallSC-NT.bat for the Client
- Make the service automatic (optional). Open the Service panel from the Administrative Tools and locate the Syndication Server or Syndication Client services. Right click and choose properties. In the Startup Type box choose “automatic”. The service will now automatically start when the operating system is restart without any intervention by the user.

5.5 User’s documentation overview

5.5.1 Content Management System

The VAP CMS provides two user interfaces to its users. An administrator interface and a CMS user interface.

The administrator interface depending of the configuration of the VAP publishing environment will provide interface for administering the CMS, and additionally interfaces for administration of the Syndication server and Client, which are detail in their respective sections.

The CMS admin interface manages users, roles and permissions. The standard version of the CMS defines three roles for users, cmsAdmin, cmsUser and cmsGuest.

When a new cmsUser is created a user workspace is also created in the CMS. The user workspace gives the user access to three types of sub-workspaces

- Private workspace, only accessible by the user who owns the workspace. The owner has read,write,delete privileges
- Project workspace, shared workspace with other users. Authorized CMS users can be given access to project file space with read, write and file delete privileges. Only the workspace owner can share project folders with other users.
- Public workspace, shared by all users of the CMS, to the normal CMS user the public workspace is read only. Only the cmsAdmin higher privileges.

The CMS user interface gives the user access to his workspaces. Managing workspaces is very similar to managing the Windows file system with the Windows Explorer application. Hieratical folder structures can be created and files uploaded. The most noticeable difference is file checkin/checkout. When a users checks out a file for editing the file is locked for other users who have access rights to it. The file is unlocked when the user checks it in again or when filelock timeout is rehashed.

The second part of the user interface is the SciX digital Library interface. CMS users can search registered SciX digital libraries, read and download papers, set bookmarks to them and retrieve bibliographic information.

The digital library registry, which is a part of this interface, contains connection information about libraries that you have access to. The information should be retrieved from the the library provides e.g. cit.scix.net, CUMINCAD, ELPUB etc.

A part of the SciX digital Library interface will enable the user to manage the downloaded information very easily. Users can as well recall bookmarked papers for reading at convenient times.

When downloading papers users can choose if they want them stored in their private folder, project folders or the public folder. It should however be noted the in the current version of the CMS users can't remove files from the public folder,. This may change in future releases of the CMS.

5.5.2 Syndication Server

The syndication server provides a user interface for registering syndication clients.

The syndication client registry is used to provide connection information and describe the existing relationship between the two. The registry entry can be better described in the figure 24.

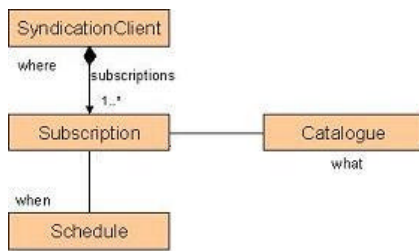


Figure 24: Syndication client registry entry

The Syndication client entry contains the following values

Host Name	The host name assigned to the Syndication client
Login Name	The login name used to connect the Syndication client
Host Password	The password associated with the login name
Host Confirm Value	A value used by server and client to confirm the identity of client and server when connecting
Host URL	The URL of the Syndication Client
Host Port	The port number of the Syndication Client
Host Authentication	A value for host authentication of the client

Any Syndication client can have many subscriptions for content delivery. The subscription specifies the details of the syndication process and contains the following values.

Label	A name assigned to the subscription
Description	A short description of what is subscribed to
Delivery Method	The only available delivery method is RSS
Refresh Updates	How often is the subscription to be updated
RSS Version	Version of RSS used. Supported RSS versions are: 0.91, 0.92, 1, 2, 3
Catalogue	The Name of the catalogue describing the content subscribed to

Catalogues are instruments to specify the content being subscribed to in an abstract way. Catalogues are re-useable items and may exist in many subscriptions.

Label	A name assigned to the Catalogue
Description	A short description of the Catalogue
ContentMap	A XML file describing the content the client wishes to receive. This is created manually by the server admin.

ContentMap format

Documents in the CMS are classified in subject specific hierarchical categories in the subject field of the document metadata with the following structure “/sub1/sub1.1/sub1.1.1”

The content map can specify content in two ways 1) by specifying a subject or 2) by specifying matching search parameters for documents.

Example content map

```

.....
<contentmap>
<subject name="subjectname"/>
...more subject elements
<matchdocument>
<param fieldname="name" machvalue="value"/>
...more param elements
</matchdocument>
....more matchdocument elements
</contentmap>

```

5.5.3 Syndication Client

In a standalone configuration the Syndication client has no user interface. Administration of the application is done by setting properties in the configuration file.

When used with the VAP CMS and item is added to the menu structure of the CMS “RssAdmin”. The option enables the CMS administrator to manage and edit RSS channels stored in the CMS. By using cut and paste new RSS channels can be created to better suit the end point.

5.6 License

The VAP applications are distributed under the license of the Apache Software Foundation (ASF). <http://www.apache.org/licenses/>

6. APPENDIX A: SOPS LICENSE

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 PRECEDING 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 MERCHANTABILITY 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.

6.1 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.

7. APPENDIX B: SOME TYPICAL SOPS FILES AND SETTINGS

This appendix lists, for illustration purposes, some of the typical SOPS files.

7.1 cgi/works

The 'works' script from ELPUB. It refers to the application (common.pl) and standard SOPS library (lib/sops/works.pl), redefines the options of the series field and submits the whole definition to Woda for processing.

```
#!/usr/local/bin/perl

require "common.pl";
require "lib/sops/works.pl";

$WBF{series,options} = '
ELPUB:1997
ELPUB:1998
ELPUB:1999
ELPUB:2000
ELPUB:2001
ELPUB:2002
ELPUB:2003
ELPUB:2004
ELPUB:2005
other:conference
other:journal
other:report
other:symposium
other:workshop
thesis:BSc
thesis:MSc
thesis:PhD
';

do main();
```

7.2 cgi/common.pl

The common.pl file of the ELPUB application.

```
require '/usr/local/woda/en/woda.pl';
require 'lib/sops/scix.pl';

$LIBRARY = "ELPUB Digital Library: ";
$DATADIRDIR = "/www/elpub.scix.net/htdocs/data";
$WBB{'apTitle'}=<<EOM;
Elpub : <A HREF=/>Digital Library</A> :
EOM

#
# --- generic setting for all digital libraries
#
```

```

$WBB{'dfiltHits'} = '20';
$WBB{'maxHits'} = '100';
$WBB{'language'} = 'UK';
$WBB{'intlCharset'} = 'ISO-8859-1';
$WBB{'userScope'}='/' ;
$WBB{'userTable'}='http://www.scix.net/cgi/users';

#
# --- special settings
#

$WBB{'groups'} = <<EOM;
admin .      deleted
owner .      deleted
EOM

$WBB{'ownerGroup'} = 'owner';
$WBB{'manager'} = 'Prof. Bob Martens';
$WBB{'managerEmail'} = 'b.martens@tuwien.ac.at';
1;

```

7.3 lib/sops/ratings.pl

Ratings service is a very primitive service but it has the features of a service. Note the two blocks defining the WBB (service wide) and WBF (field specific) settings.

```

$WBB{'dataDir'} = "$DATADIRDIR/ratings";
$WBB{'dbTitle'} = 'Ratings'; #UK
$WBB{'about'} = <<EOM; #UK
Here are the ratings related to the $LIBRARY. They show how much the users like the
individual items in the library.
EOM

$WBB{'recordLock'}='user';
$WBB{'ownerField'}='from';
$WBB{'rights'} = '
allow owner Add|Edit|Adm|Delete
allow user Add|Edit
deny . Add
deny . DisplayStructure|BrowseRing|Register
';

$WBB{'sort;DEFAULT'} = '$rec{rating}';
$WBB{'key'}="$rec{paper}.$rec{from}";
$WBB{'format;DEFAULT'} = <<'EOM';
&PIC();
"$pic{from} rated $pic{paper} as <B>$pic{rating}</B>"
EOM

#
# Field definitions:
#

$i=1000;          # counter

```

```

$x='from';      # -----
$WBB{$x,'head'}='from'; #UK
$WBF{$x,srt}=$i--;
$WBF{$x} = '1;';
$WBF{$x,'type'} = 'INPUT';
$WBF{$x,'d'} = "$User";
$WBF{$x,'typePar'} = 'HIDDEN';

$x='paper';      # -----
$WBB{$x,'paper'}=""; #UK
$WBF{$x,srt}=$i--;
$WBF{$x} = '&FLD("works",$_, "title") ne ""';
$WBF{$x,picture} = '&LINK("works",$_,&FLD("works",$_, "title"))';
$WBF{$x, help} = 'Should be a valid ID of a paper'; #UK
$WBF{$x,'type'} = 'INPUT';

$x='rating';      # -----
$WBB{$x,'head'}='rating'; #UK
$WBF{$x,srt}=$i--;
$WBF{$x} = '1;';
$WBF{$x,'type'} = 'OPTION';
$WBF{$x,'d'} = '3';
$WBF{$x,'picture'} = '&DECODE()';
$WBF{$x,'options'} = <<EOM;  #UK
1      poor
2      average
3      good
4      very good
5      excellent
EOM

$x='comment'; # -----
$WBB{$x,'head'}='comment'; #UK
$WBF{$x,srt}=$i--;
$WBF{$x} = '1;';
$WBF{$x,'type'} = 'HTMLAREA';

1;

```

7.4 Compound definition of the Works services

Essentials

dbTitle Works

apTitle Elpub : Digital Library :

about <P>This is the ELPUB Digital Library: .

<P>

<P>The use is free of charge.

Access to advanced features (article upload, creation of collections, discussion forums, ratings, reviews, etc.)

is subject to user registration and login.

<P>The ELPUB Digital Library: is managed by Prof. Bob Martens. Contact details below under 'Database management'

<P>This application is demonstrating the SciX technology.

recordTitle publication
manager Prof. Bob Martens
managerEmail b.martens@tuwien.ac.at
dataDir /www/elpub.scix.net/htdocs/data/works
htmlDir /www/elpub.scix.net/htdocs/data/works
htmlURL /data/works
sort;DEFAULT "\$rec{authors},\$rec{year}"
key \$rec{id}

Icons

iconOpen paper

Add to toolbars

toolbar Basket "share this selection"

Add to menus on the Home page

userMenuAdd Contribute a publication,/cgi-bin/contrib/Add
 Add a user,/cgi-bin/users/Add

userMenuOther OAI Harvesting,OAIHelp

Change the look of the Detail page

detailTitle "Paper \$rec{_id}:
\$rec{title}"

Speed up searches

grepBigger 300000

User defined search results format 'DEFAULT'

formatNameDEFAULT Default

formatHeadDEFAULT <TABLE ALIGN=CENTER CELLSPACING=1>
 <TR VALIGN=TOP>
 <TH>#</TH>
 <TH>details</TH>
 <TH>full text</TH>
 <TH>citation</TH>
 <TH>check to select</TH>
 </TR>

formatRowDEFAULT

```
if ($rec{content} && -s "$DATADIR/att/$rec{content}") {
    $x = qq(<IMG $ICONPAR SRC="$ICONURL/pdf.gif">);
} elsif ( $rec{more} ) {
    $x = qq(<IMG $ICONPAR SRC="$ICONURL/fforward.gif">);
} else {
    $x = "&nbsp;";
}

"
<TR VALIGN=TOP>
<TD>$ThisRecordNo.</TD>
<TD ALIGN=CENTER VALIGN=TOP>$ThisRecordIcons</TD>
<TD ALIGN=CENTER VALIGN=TOP>$x</TD>
<TD WIDTH=100%>
$rec{authors} ($rec{year}) <A
  HREF=$ThisRecordURL><B>$rec{title}</B></A>, $rec{source}
</TD>
<TD WIDTH=20 VALIGN=TOP>
$ThisRecordBasket
</TD></TR>
"
```

formatFootDEFAULT </TABLE><P>

User defined search results format '1'

formatName1 Harward style references

formatHead1

formatRow1 "

\$rec{authors} (\$rec{year}) \$rec{title} \$rec{source} ,
http://itc.scix.net/id.cgi/\$rec{id}.

"

formatFoot1

User defined search results format '2'

formatName2 BibTeX references

formatHttpHead2 Content-type: text/plain

formatRow2 \$i = \$rec{authors};

\$i =~ s/[A-Z]\W+//g;

\$i =~ s/\s//g;

\$s = \$rec{source};

\$a = \$rec{abstract};

\$s =~ s/\n/ /g;

"

\@misc {\$i\$rec{year},
author = {\$rec{authors}},
title = {\$rec{title}},
year = {\$rec{year}},
publisher = {\$s},
abstract = {\$a}
URL = {http://itc.scix.net/id.cgi/\$rec{id}}

}

"

User defined search results format '3'

formatName3 Endnote (refer)

formatHttpHead3 Content-type: application/x-endnote-refer

formatRow3 \$i = \$rec{authors};

\$i =~ s/[A-Z]\W+//g;

\$i =~ s/\s//g;

\$s = \$rec{source};

\$s =~ s/\n/ /g;

"

%A \$rec{authors}
%T \$rec{title}
%D \$rec{year}
%B \$s
%U http://itc.scix.net/id.cgi/\$rec{id}
%X \$rec{summary}

"

Custom browse options

AZindex authors authors Author index
titles title Index of words in titles
abstrac summary,title Index of words in titles and abstracts

AZindexMethod authors ASCII,TITLE,SPACE

AZnot the
of
and
to

in
 is
 for
 this
 are
 that
 on
 it
 be
 as
 an
 with
 by
 which
 will
 from
 has
 can
 these
 have
 been
 used
 use
 or
 at
 such
 new
 we
 not
 was
 there
 also
 its
 within
 other
 more
 their
 one
 all
 so
 any
 than
 what
 may
 if

Shopping basket

basketName favorite papers

Group based security

groups	admin	.	zebra
	owner	.	plenica
rights	allow	owner	^Add ^Edit ^Adm ^Delete
	deny	.	^Add ^Edit ^Delete
	deny	.	^DisplayStructure ^BrowseRing ^Register
	deny	guest	^AdvancedSearch
	deny	.	^Search ^Basket
			format=XML format=CSV format=TAB format=PARA

userScope /

User based security

userTable users
ownerGroup owner

Limiting number of found records

dfltHits 20
maxHits 100

Support for non-english languages

language UK
intlCharset ISO-8859-1

7.4.1 Field settings

id

typePar SIZE=10
p Unique ID of the paper
cond Must be non-empty
srt 1000

authors

p All authors of the paper
help Enter all authors of the paper. Suggested formats are like "Turk Z, Martens B" or "Turk, Ziga; Martens, Bob"
cond Letters, dashes, spaces, semicolons. No dots!
meta DC.creator
srt 999

year

typePar SIZE=4
p Year when this was published
cond 4 digits.
meta DC.date
srt 998

title

p Full title of the publication or paper. Sentence case is suggested.
cond Starts with a capital and followed by up to 100 characters.
change \$_ =~ tr/[a-z]/[A-Z]/; return \$_;
meta DC.title
srt 997

source

type HTMLAREA
typePar ROWS=2 COLS=50
p Where was it published
help Enter all other publication data such as journal name, volume, number, page numbers etc.
sticky 1
meta DC.source

srt 996

summary

type HTMLAREA

p Brief summary

meta DC.description

srt 995

keywords

meta DC.subject

srt 994

series

type OPTION

p If this was published in a series (or group), select one:

options,

ELPUB:1997,
ELPUB:1998,
ELPUB:1999,
ELPUB:2000,
ELPUB:2001,
ELPUB:2002,
ELPUB:2003,
ELPUB:2004,
ELPUB:2005,
other:conference,
other:journal,
other:report,
other:symposium,
other:workshop,
thesis:BSc,
thesis:MSc,
thesis:PhD

sticky 1

srt 993

type

type OPTION

p Is this document of a special type:

options normal paper,
keynote paper,
short paper,
extended abstract,
poster,
paper session,
symposium,
workshop

sticky 1

srt 992

email

type EMAIL

p Email address of the contact author
sticky/
srt 991

more

type URL
p URL of the full text
help Enter a URL where more information (perhaps the full text, multimedia content) of the publication is available on the Internet.
srt 990

content

type USERFILE
srt 989

discussion

type COMPUTE
picture \$replyto="works-\$rec{_id}"; \$x =
&ROWS("forum","_id","replyTo:\$replyto"); \$replyto =
&htEscape(\$replyto); \$x = \$x ~ tr/\n/\n/; if (\$x) { \$o = "<A
CLASS=BUTTON HREF=/cgi-
bin/forum/Search?search=replyTo:\$replyto>\$x
discussions
\n"; #UK } else { \$o = "No discussions.\n";
#UK \$s = \$rec{title}; \$s = &htEscape(\$s); \$o .= "<A
CLASS=BUTTON HREF=/cgi-
bin/forum/Add?replyTo=\$replyto&subject=\$s>Post discussion
...\n"; #UK }
srt 988

ratings

type COMPUTE
picture \$o=""; \$paper="\$rec{_id}"; \$x =
&ROWS("ratings","rating","paper:\$paper"); if (\$x) { \$o = "<A
HREF=/cgi-
bin/ratings/Search?search=paper:\$paper>Ratings:
\$x
\n"; } \$o .=<<EOT; <FORM ACTION="/cgi-
bin/ratings/Changed"> <INPUT TYPE=HIDDEN NAME=from
VALUE="\$User"> <INPUT TYPE=HIDDEN NAME=paper
VALUE="\$rec{_id}"> <SELECT NAME=rating> <OPTION
VALUE="1">poor(1) <OPTION VALUE="2">average(2)
<OPTION SELECTED VALUE="3">good(3) <OPTION
VALUE="4">very good(4) <OPTION VALUE="5">excellent(5)
</SELECT> <INPUT TYPE=SUBMIT VALUE="Rate it!">
</FORM> EOT
srt 987

7.5 Scientific conferences review and submission support

Below are the extensions of the people.pl service required for the conference support.

```

$x='topics'; # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = '1;';
$WBF{$x,cond}='Select between 1 and 5';
$WBF{$x,'type'} = 'OPTION';
$WBF{$x,'typePar'} = 'CHECKBOX COLS=1 NOID';
$WBF{$x,'options'} = <<EOM;
topics1ID      topic1description
...
..
.
EOM
$WBF{$x,picture}      = <<'EOM';
$v=&htEscape($_);
$WBF{'topics','options'} =~ m/$v[^\d]*/;
$m = $1;
"<A HREF=\"\${SCRIPT}/BrowseTree?field=topics&value=$v\">$_ $m</A><BR>";
EOM
$WBF{$x,'head'} = 'Topics';

if ($status eq "yes") {

  $x='breakAbs'; # -----
  $WBF{$x,srt}=$i--;
  $WBF{$x} = 'RELATED TO THE CONFERENCE';
  $WBF{$x,'type'} = 'BREAK';

  $x = 'intention'; #####
  $WBF{$x,'srt'}=$i--;
  $WBF{$x}='1;';
  $WBF{$x,'type'}='OPTION';
  $WBF{$x,'options'}='I will not attend
  I am not comming, stop sending me any email
  I will attend. Save me a seat and a copy of the proceedings.
  ';
}

$x='papersSubmitted';
$WBF{$x} = '1';
$WBF{$x,srt}=$i--;
$WBF{$x,'type'} = 'COMPUTE';
$WBF{$x,'picture'} = '&RELATED("papers","title","=contact:$ID)';

$x='reviewsAssigned';
$WBF{$x} = '1';
$WBF{$x,srt}=$i--;
$WBF{$x,'type'} = 'COMPUTE';
$WBF{$x,'picture'} = <<'EOM';

  return if # $User ne $rec{username} &&
  $Group ne 'admin';

  local (@x,$x,$out,$t,$a,$o);

  @x = split( /\n/, &ROWS("revass","paper","/reviewers:$ID"));
  $out=";

```

```

    foreach $x (@x) {
        $t = &FLD("papers",$x,"title");
        $a = &FLD("papers",$x,"authors");
        $o .= qq(<A HREF="/cgi-bin/papers/Show?$x">$t</A> $a <BR>\n);
    }
    $o;
EOM

$x='reviewsDone';
$WBF{$x} = '1';
$WBF{$x,srt}=$i--;
$WBF{$x,'type'} = 'COMPUTE';
$WBF{$x,'picture'} = '

        return if $User ne $rec{username} && $Group ne "admin";
';

$x='breakAbs'; # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = 'ADMINISTRATIVE';
$WBF{$x} = ' --- ';
$WBF{$x,'type'} = 'BREAK';
$WBF{$x,'sees'} = 'admin';
$WBF{$x,'modifies'} = 'admin';

$x='committee'; # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = '1;';
$WBF{$x,p} = 'In which committees is this person involved';
$WBF{$x,'type'} = 'OPTION';
$WBF{$x,'typePar'} = 'CHECKBOX';
$WBF{$x,'modifies'} = 'admin';
$WBF{$x,'options'}=<<EOM;
ORGANISING
PROGRAMME
EOM

$x='adminComment'; # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = '1;';
$WBF{$x,p} = 'Any other administrative comment';
$WBF{$x,'type'} = 'TEXTAREA';
$WBF{$x,'modifies'}='admin';
$WBF{$x,'sees'} = 'admin';

if ($status eq "yes") {

$x='preregistered'; # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = '1;';
$WBF{$x,p} = 'Was this person pre-registered';
$WBF{$x,'type'} = 'OPTION';
$WBF{$x,'typePar'} = 'RADIO';

```

```

$WBF{$x,'modifies'} = 'admin';
$WBF{$x,'sees'} = 'admin';
$WBF{$x,'options'}=<<EOM;
NO
YES
EOM

$x='chairs';      # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = '1;';
$WBF{$x,p} = 'Which session this person chairs';
$WBF{$x,'type'} = 'INPUT';
$WBF{$x,'modifies'} = 'admin';
$WBF{$x,'sees'} = 'admin';

$x='paymentAmount'; # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = '1;';
$WBF{$x,p} = 'Amount paid in Euro (unless otherwise stated)';
$WBF{$x,'type'} = 'INPUT';
$WBF{$x,'modifies'}='admin';
$WBF{$x,'typePar'} = 'SIZE=8';
$WBF{$x,'sees'} = 'admin';

$x='paymentDate'; # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = '1;';
$WBF{$x,p} = 'When was the payment made';
$WBF{$x,'type'} = 'DATE';
$WBF{$x,'modifies'}='admin';
$WBF{$x,'sees'} = 'admin';

$x='spouse'; # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = '1;';
$WBF{$x,p} = 'Comming with a spouse?';
$WBF{$x,'type'} = 'OPTION';
#$WBF{$x,'typePar'} = 'RADIO';
$WBF{$x,'modifies'} = 'admin';
$WBF{$x,'sees'} = 'admin';
$WBF{$x,'options'}=<<EOM;
NO
YES
YES,REGISTERED
EOM

$x='breakTravel'; # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = 'TRAVEL DETAILS';
$WBF{$x,'type'} = 'BREAK';

$x='arrivalDate'; # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = '1;';
$WBF{$x,p} = 'Date of arrival.';
$WBF{$x,'type'} = 'DATE';

```

```

$x='arrivalTime';      # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = 'm^d\d:\d\d|^$';
$WBF{$x,cond}='like 09:35 or 22:45';
$WBF{$x,p} = 'Time of landing (on the arrival date).';
$WBF{$x,'type'} = 'INPUT';
$WBF{$x,'typePar'} = 'SIZE=5';

$x='transportReservation';      # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = '1;';
$WBF{$x,p} = 'Did you make a transport reservation by the hotel';
$WBF{$x,'type'} = 'OPTION';
$WBF{$x,'options'}='no|yes';

$x='departureDate';      # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = '1;';
$WBF{$x,p} = 'Date of arrival to Portoroz.';
$WBF{$x,'type'} = 'DATE';

$x='departureTime';      # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = 'm^d\d:\d\d|^$';
$WBF{$x,cond}='like 09:35 or 22:45';
$WBF{$x,p} = 'Time of departure (on the arrival date).';
$WBF{$x,'type'} = 'INPUT';
$WBF{$x,'typePar'} = 'SIZE=5';

$x='foodPreference';      # -----
$WBF{$x,srt}=$i--;
$WBF{$x} = '1;';
$WBF{$x,p} = 'Any special food preferences';
$WBF{$x,'type'} = 'OPTION';
$WBF{$x,'options'}='NONE|NO BEEF|NO PORK|NO RED MEAT|VEGETARIAN';
}

```

B. APPENDIX C: WSDL DEFINITIONS

Such definitions are available for all SOPS services. Listed are only the samples for the “works” service.

B.1 works Office 2003 WSDL

```
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:s0="urn:Microsoft.Search" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" targetNamespace="urn:Microsoft.Search"
xmlns="http://schemas.xmlsoap.org/wsdl/">
<types>
<s:schema elementFormDefault="qualified" targetNamespace="urn:Microsoft.Search">
<s:element name="Registration">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="registrationXml" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
<s:element name="RegistrationResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="RegistrationResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
<s:element name="Query">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="queryXml" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
<s:element name="QueryResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="QueryResult" type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
</s:schema>
</types>
<message name="RegistrationSoapIn">
<part name="parameters" element="s0:Registration" />
</message>
<message name="RegistrationSoapOut">
<part name="parameters" element="s0:RegistrationResponse" />
</message>
<message name="QuerySoapIn">
<part name="parameters" element="s0:Query" />
</message>
```



```

<message name="QuerySoapOut">
<part name="parameters" element="s0:QueryResponse" />
</message>
<portType name="ResearchSoap">
<operation name="Registration">
<documentation>Registration for the Woda Search Service.</documentation>
<input message="s0:RegistrationSoapIn" />
<output message="s0:RegistrationSoapOut" />
</operation>
<operation name="Query">
<documentation>Runs a search in Woda.</documentation>
<input message="s0:QuerySoapIn" />
<output message="s0:QuerySoapOut" />
</operation>
</portType>
<binding name="ResearchSoap" type="s0:ResearchSoap">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
<operation name="Registration">
<soap:operation soapAction="urn:Microsoft.Search/Registration" style="document" />
<input>
<soap:body use="literal" />
</input>
<output>
<soap:body use="literal" />
</output>
</operation>
<operation name="Query">
<soap:operation soapAction="urn:Microsoft.Search/Query" style="document" />
<input>
<soap:body use="literal" />
</input>
<output>
<soap:body use="literal" />
</output>
</operation>
</binding>
<service name="Research">
<documentation>Woda Research Web Service</documentation>
<port name="ResearchSoap" binding="s0:ResearchSoap">
<soap:address location="http://iaps.scix.net/cgi-bin/works/Office" />
</port>
</service>
</definitions>

```

8.2 WORKS WSDL

```

<?xml version="1.0" encoding="utf-8"?>
<definitions
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://elpub.scix.net/cgi-bin/works"
  targetNamespace="http://elpub.scix.net/cgi-bin/works">

```

```

<types>
  <xsd:schema elementFormDefault="qualified" targetNamespace="http://elpub.scix.net/cgi-
bin/works">
    <xsd:complexType name="rec">
      <xsd:sequence>
        <xsd:element name="_id" type="xsd:string"/>
        <xsd:element name="id" type="xsd:string"/>
        <xsd:element name="authors" type="xsd:string"/>
        <xsd:element name="year" type="xsd:string"/>
        <xsd:element name="title" type="xsd:string"/>
        <xsd:element name="source" type="xsd:string"/>
        <xsd:element name="summary" type="xsd:string"/>
        <xsd:element name="keywords" type="xsd:string"/>
        <xsd:element name="series" type="xsd:string"/>
        <xsd:element name="type" type="xsd:string"/>
        <xsd:element name="email" type="xsd:string"/>
        <xsd:element name="more" type="xsd:string"/>
        <xsd:element name="content" type="xsd:string"/>
        <xsd:element name="discussion" type="xsd:string"/>
        <xsd:element name="ratings" type="xsd:string"/>
        <xsd:element name="_from" type="xsd:string"/>
        <xsd:element name="_at" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="records">
      <xsd:sequence>
        <xsd:element name="recs" type="tns:rec"/>
      </xsd:sequence>
    </xsd:complexType>

  </xsd:schema>
</types>

<message name="SearchInputSoap">
  <part name="search" type="xsd:string"/>
  <part name="format" type="xsd:string"/>
  <part name="fields" type="xsd:string"/>
</message>
<message name="SearchOutputSoap">
  <part name="return" element="tns:records"/>
</message>
<message name="ShowInputSoap">
  <part name="format" type="xsd:string"/>
  <part name="id" type="xsd:string"/>
  <part name="fields" type="xsd:string"/>
</message>
<message name="ShowOutputSoap">
  <part name="return" element="tns:rec"/>
</message>
<message name="ShowInput"/>
<message name="ShowOutput">
  <part name="Body" element="tns:string"/>
</message>
<message name="SearchInput"/>
<message name="SearchOutput">

```

```

        <part name="Body" element="tns:string"/>
    </message>
    <message name="XputInput"/>
    <message name="XputOutput">
        <part name="Body" element="tns:string"/>
    </message>
    <message name="XputManyInput"/>
    <message name="XputManyOutput">
        <part name="Body" element="tns:string"/>
    </message>

    <portType name="WodaSoap">
        <operation name="Show">
            <input message="tns:ShowInputSoap"/>
            <output message="tns:ShowOutputSoap"/>
        </operation>
        <operation name="Search">
            <input message="tns:SearchInputSoap"/>
            <output message="tns:SearchOutputSoap"/>
        </operation>
        <operation name="Xput">
            <input message="tns:XputInputSoap"/>
            <output message="tns:XputOutputSoap"/>
        </operation>
        <operation name="XputMany">
            <input message="tns:XputManyInputSoap"/>
            <output message="tns:XputManyOutputSoap"/>
        </operation>
    </portType>
    <portType name="WodaHttpGet">
        <operation name="Show">
            <input message="tns:ShowInput"/>
            <output message="tns:ShowOutput"/>
        </operation>
        <operation name="Search">
            <input message="tns:SearchInput"/>
            <output message="tns:SearchOutput"/>
        </operation>
        <operation name="Xput">
            <input message="tns:XputInput"/>
            <output message="tns:XputOutput"/>
        </operation>
        <operation name="XputMany">
            <input message="tns:XputManyInput"/>
            <output message="tns:XputManyOutput"/>
        </operation>
    </portType>
    <portType name="WodaHttpPost">
        <operation name="Show">
            <input message="tns:ShowInput"/>
            <output message="tns:ShowOutput"/>
        </operation>
        <operation name="Search">
            <input message="tns:SearchInput"/>
            <output message="tns:SearchOutput"/>
        </operation>
        <operation name="Xput">

```

```

                <input message="tns:XputInput"/>
                <output message="tns:XputOutput"/>
            </operation>
            <operation name="XputMany">
                <input message="tns:XputManyInput"/>
                <output message="tns:XputManyOutput"/>
            </operation>
        </portType>

        <binding name="WodaSoap" type="tns:WodaSoap">
            <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>

            <operation name="Show">
                <soap:operation soapAction="http://elpub.scix.net/cgi-bin/works/Show"
style="document"/>
                <input>
                    <soap:body use="literal"/>
                </input>
                <output>
                    <soap:body use="literal"/>
                </output>
            </operation>
            <operation name="Search">
                <soap:operation soapAction="http://elpub.scix.net/cgi-bin/works/Search"
style="document"/>
                <input>
                    <soap:body use="literal"/>
                </input>
                <output>
                    <soap:body use="literal"/>
                </output>
            </operation>
            <operation name="Xput">
                <soap:operation soapAction="http://elpub.scix.net/cgi-bin/works/Xput"
style="document"/>
                <input>
                    <soap:body use="literal"/>
                </input>
                <output>
                    <soap:body use="literal"/>
                </output>
            </operation>
            <operation name="XputMany">
                <soap:operation soapAction="http://elpub.scix.net/cgi-bin/works/XputMany"
style="document"/>
                <input>
                    <soap:body use="literal"/>
                </input>
                <output>
                    <soap:body use="literal"/>
                </output>
            </operation>
        </binding>
        <binding name="WodaHttpGet" type="tns:WodaHttpGet">
            <http:binding verb="GET"/>
            <operation name="">
                <http:operation location="http://elpub.scix.net/cgi-bin/works/Show"/>
            </operation>
        </binding>
    
```

```

        <input>
            <http:urlEncoded/>
        </input>
        <output>
            <mime:mimeType part="Body"/>
        </output>
    </operation>
    <operation name="">
        <http:operation location="http://elpub.scix.net/cgi-bin/works/Search"/>
        <input>
            <http:urlEncoded/>
        </input>
        <output>
            <mime:mimeType part="Body"/>
        </output>
    </operation>
    <operation name="">
        <http:operation location="http://elpub.scix.net/cgi-bin/works/Xput"/>
        <input>
            <http:urlEncoded/>
        </input>
        <output>
            <mime:mimeType part="Body"/>
        </output>
    </operation>
    <operation name="">
        <http:operation location="http://elpub.scix.net/cgi-bin/works/XputMany"/>
        <input>
            <http:urlEncoded/>
        </input>
        <output>
            <mime:mimeType part="Body"/>
        </output>
    </operation>
</binding>
<binding name="WodaHttpPost" type="tns:WodaHttpPost">
    <http:binding verb="POST"/>
    <operation name="Show">
        <http:operation location="http://elpub.scix.net/cgi-bin/works/Show"/>
        <input>
            <mime:contentType="application/x-www-form-urlencoded"/>
        </input>
        <output>
            <mime:mimeType part="Body"/>
        </output>
    </operation>
    <operation name="Search">
        <http:operation location="http://elpub.scix.net/cgi-bin/works/Search"/>
        <input>
            <mime:contentType="application/x-www-form-urlencoded"/>
        </input>
        <output>
            <mime:mimeType part="Body"/>
        </output>
    </operation>
    <operation name="Xput">
        <http:operation location="http://elpub.scix.net/cgi-bin/works/Xput"/>

```

```

        <input>
            <mime:content type="application/x-www-form-urlencoded"/>
        </input>
        <output>
            <mime:mimeXml part="Body"/>
        </output>
    </operation>
    <operation name="XputMany">
        <http:operation location="http://elpub.scix.net/cgi-bin/works/XputMany"/>
        <input>
            <mime:content type="application/x-www-form-urlencoded"/>
        </input>
        <output>
            <mime:mimeXml part="Body"/>
        </output>
    </operation>
</binding>

<service name="WodaService">
    <port name="WodaSoap" binding="tns:WodaSoap">
        <soap:address location="http://elpub.scix.net/cgi-bin/works/SOAP"/>
    </port>
    <port name="WodaHttpGet" binding="tns:WodaHttpGet">
        <http:address location="http://elpub.scix.net/cgi-bin/works"/>
    </port>
    <port name="WodaHttpPost" binding="tns:WodaHttpPost">
        <http:address location="http://elpub.scix.net/cgi-bin/works"/>
    </port>
</service>

```