Using the SORASCS Prototype Web Portal

Bradley Schmerl, Michael W. Bigrigg,
David Garlan, Kathleen M. Carley

September, 2010
CMU-ISR-10-123

Institute for Software Research
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Center for the Computational Analysis of Social and Organizational Systems
CASOS technical report.

This work was supported in part by the Office of Naval Research - ONR -N000140811223. Additional support was provided by the center for Computational Analysis of Social and Organizational Systems (CASOS). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research, or the U.S. government.

Keywords: SORASCS, Services, Portal

Abstract

This document describes how to use the example portal distributed with SORASCS. The portal is meant as an example of how users may interact with services. It covers three things: 1) Navigating the portal; 2) Registering and managing services; and 3) Using the SORASCS Thick Client interface.

Table of Contents

1 Introduction

Over the past decade there have emerged a large variety of tools supporting analysis of heterogeneous information to understand complex real world relationships and trends, including natural language processing, network analysis, simulation, and what-if reasoning. Most analysis ensembles assume a relatively limited set of model types and input sources. They are wired together with special purpose, tool-specific and ad hoc integration code, making them brittle and requiring low-level system expertise to reconfigure in new ways or add new capabilities. The underlying problems stem from the lack of a coherent and flexible architectural framework that will allow tools and models to be seamlessly incorporated and composed by end users. Having such a framework would ideally allow information analysts to bring together tools from various vendors and research groups, dynamically compose their analyses in new ways, store and reuse workflows and settings, ingest new forms of information, and interactively fine tune analyses, simulations, and report generation through consistent and uniform forms of interaction.

SORASCS [1] is an architecture for the intelligence community that will enable these new capabilities. This architecture is based on service oriented architectures (SOA), which are typically used in the business community for integration and use of different business systems and processes developed by different companies at different times. The key aspects of SOA that enable this are:

- Modularity of components as services, providing simple operations that are loosely coupled. This allows applications to be easily composed out of existing services.
- Standards for communicating among distributed services over the web.
- Standards based support for discovery, orchestration, governance, security, etc.

The goal of SOA is to support the development of applications using web services, and therefore many of the features provided are general in nature, so that they can be used in many domains. SORASCS is layered upon these general features to provide features customized to the domain of intelligence analysis, and provides a framework upon which tool developers can integrate functionality with SORASCS

The purpose of this document is to serve as a user guide for the example web portal distributed with SORASCS. While the portal is not intended to be the main interaction between users and SORASCS (third party tools are intended to eventually take that place), it provides an example of how a website could be constructed for interacting with services, and serves as a test bed for testing the installation of SORASCS and a demonstration of some of the ideas we are investigating in SORASCS. The document first describes the context of the SORASCS Portal with respect to the overall SORASCS architecture. Following that, users are given instructions on how to use services. Next, the thick client user interface is described, which allows users to determine what SORASCS is doing, and register local programs as thick clients for use in SORASCS. Finally, users are taken through the process of how to register and manage new services.

2 SORASCS Architecture in a Nutshell

Work on SORASCS is developing an architecture that will be useful with both web-based and non web-based intelligence models and analysis tools. The aim of SORASCS is to be:

- Light: SORASCS should not involve providers completely rewriting their tools to conform to SORASCS. SORASCS should automate mundane tasks such as security, data use, etc., so that users and integrators are not burdened with these issues.
- Extensible: SORASCS should easily accommodate new kinds of tools, models, and services that are currently being developed in this thriving community.
- Community-based: Provisioned services will not be provided by SORASCS, but by tool providers in the intelligence community. The architecture therefore needs to be community based and open source.

In a nutshell, SORASCS will serve as the glue for linking a set of heterogeneous data sources and tools needed for modeling and analysis to support the needs of the intelligence community. Although we anticipate many third-party tools making use of the services, an initial proof of concept user interface is the SORASCS Portal, which provides access to individual services.

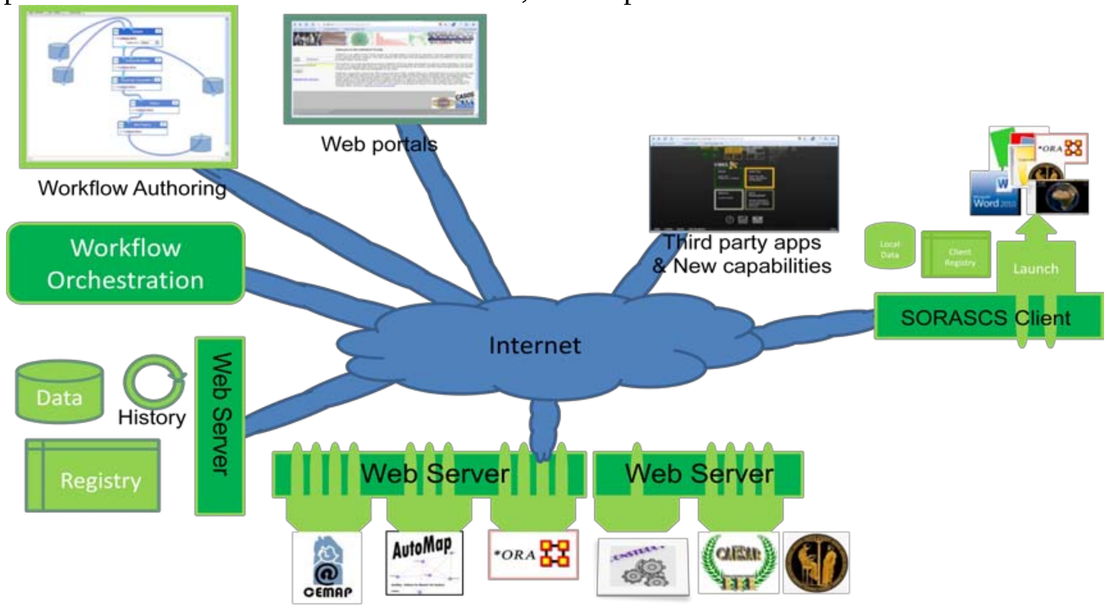


Figure 1. The SORASCS Architecture.

Figure 1 outlines the broad architecture for SORASCS; it consists of:

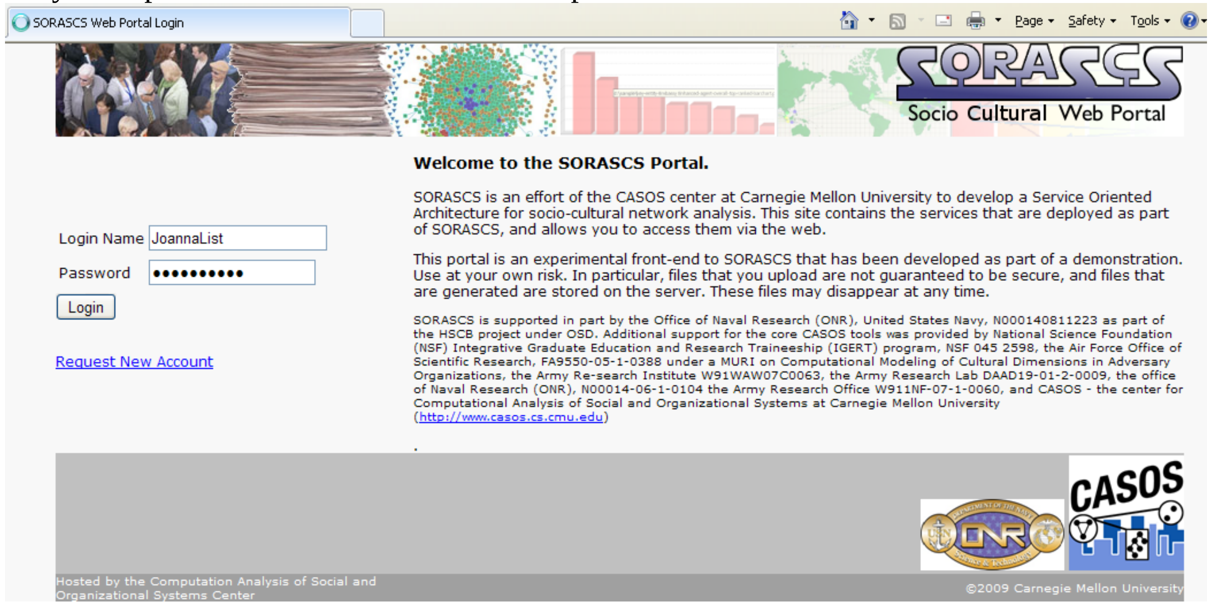
- Tools and components wrapped as web services and provided to the community via standard web service interfaces. Services are invoked by sending them messages (formatted as SOAP messages) through these interfaces. SORASCS defines a set of domain-specific APIs that SORASCS clients (web portals, workflow tools, and other custom tools) can use to interact with SORASCS services.
- A service registry, built on top of standard web service registry technologies, that provides additional information about services (currently parameterization, but other information such as quality of service, security related information, etc., is envisioned to go here).
- Data Services, that provide files and data required by services. References to data are passed to services and then resolved by SORASCS to reside on the local machine, for access by components and tools.
- Client-side services that provide SORASCS access to thick client tools that can be used on a user's local machine

- SORASCS level workflows, that are defined using a domain-specific workflow language that is designed to be accessible to data analysts. This workflow language allows analysts to capture and share common tasks with other analysts. The domain-specific language is translated into the technical language, BPEL, which is a standard SOA orchestration language and then executed on an open source execution platform.

This document is a user guide to the portal, which is one client that uses individual SORASCS services including extraction, text processing, and reporting.

3 Using the SORASCS Portal

When first browsing to the portal, the user must log in. If SORASCS has been installed locally, then the user can use the demonstration user account: JoannaList. However, if the user is using the public SORASCS Portal, they will need to enter their user id and password, or request a new account. Files in SORASCS are stored on the server, and so are protected by these user names and passwords. Users can only access and create files in their own user workspace on the server. They can upload files to the server and manipulate them there.



Login Name JoannaList
Password ••••••••••

Welcome to the SORASCS Portal.

SORASCS is an effort of the CASOS center at Carnegie Mellon University to develop a Service Oriented Architecture for socio-cultural network analysis. This site contains the services that are deployed as part of SORASCS, and allows you to access them via the web.

This portal is an experimental front-end to SORASCS that has been developed as part of a demonstration. Use at your own risk. In particular, files that you upload are not guaranteed to be secure, and files that are generated are stored on the server. These files may disappear at any time.

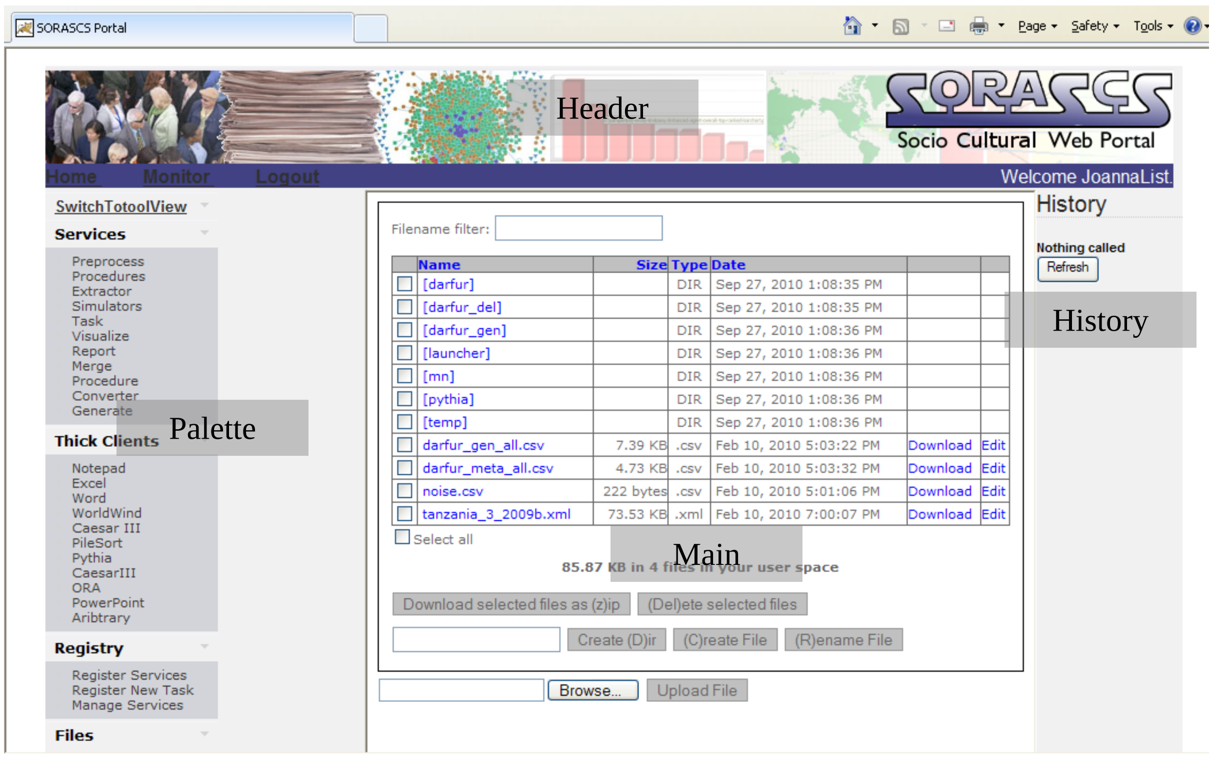
SORASCS is supported in part by the Office of Naval Research (ONR), United States Navy, N000140811223 as part of the HSCB project under OSD. Additional support for the core CASOS tools was provided by National Science Foundation (NSF) Integrative Graduate Education and Research Traineeship (IGERT) program, NSF 045 2598, the Air Force Office of Scientific Research, FA9550-05-1-0388 under a MURI on Computational Modeling of Cultural Dimensions in Adversary Organizations, the Army Re-search Institute W91WAW07C0063, the Army Research Lab DAAD19-01-2-0009, the office of Naval Research (ONR), N00014-06-1-0104 the Army Research Office W911NF-07-1-0060, and CASOS - the center for Computational Analysis of Social and Organizational Systems at Carnegie Mellon University (http://www.casos.cs.cmu.edu)

Hosted by the Computation Analysis of Social and Organizational Systems Center

©2009 Carnegie Mellon University

Once the user has logged on, they are presented with the SORASCS home page, which os divided into four regions:

1. The *header* region provide links for getting back to this home page from any page in the portal, accessing the monitoring part of SORASCS for seeing the progress of long-running services such as simulations, and logging out.
2. The *palette* region, which provides links to pages in the portal for invoking services, launching thick clients, managing services, etc.
3. The *main* region. Initially, this region displays the files that are available in the user's workspace, but it will also present forms to allow the user to enter details when invoking a service, and will display any results..
4. The *history* region, which displays a list of services that the user has invoked, with the most recently invoked service at the top of the list.



Name	Size	Type	Date		
[darfur]		DIR	Sep 27, 2010 1:08:35 PM		
[darfur_del]		DIR	Sep 27, 2010 1:08:35 PM		
[darfur_gen]		DIR	Sep 27, 2010 1:08:36 PM		
[launcher]		DIR	Sep 27, 2010 1:08:36 PM		
[mn]		DIR	Sep 27, 2010 1:08:36 PM		
[pythia]		DIR	Sep 27, 2010 1:08:36 PM		
[temp]		DIR	Sep 27, 2010 1:08:36 PM		
darfur_gen_all.csv	7.39 KB	.csv	Feb 10, 2010 5:03:22 PM	Download	Edit
darfur_meta_all.csv	4.73 KB	.csv	Feb 10, 2010 5:03:32 PM	Download	Edit
noise.csv	222 bytes	.csv	Feb 10, 2010 5:01:06 PM	Download	Edit
tanzania_3_2009b.xml	73.53 KB	.xml	Feb 10, 2010 7:00:07 PM	Download	Edit

85.87 KB in 4 files in your user space

3.1 The Main Region

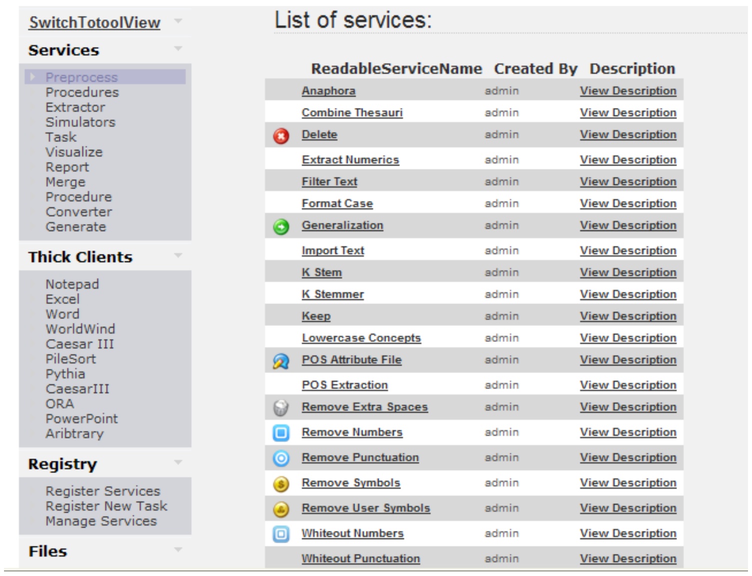
On the home page of SORASCS, the Main region shows the files owned by the user that have been uploaded to SORASCS. In this mode, there are several activities the user can perform:

1. *Clicking* on a file or folder. Names surrounded by […] are folders, or directories, containing other files. Clicking on those names will show the files that are contained in that folder. Clicking on files will cause them to be downloaded by the browser.
2. *Filename filter*. If there are a lot of uploaded files, the filename filter can be used to display only those filenames starting with the string typed into the corresponding box.
3. *Download*. Clicking the Download link on files will cause the browser to download the associated file.
4. *Edit*. This link opens the file in a simple text editor in the browser, allowing the contents to be edited.
5. Download selected files as (z)ip: This button combines the files selected (by their check boxes) into a zip file and causes the browser to download that zip file. Pressing ‘z’ in this window will invoke the same behavior.
6. (D)elete selected files. This causes the selected files or folders to be deleted from the server. Pressing ‘D’ will have the same behavior.
7. Create (D)ir. Typing a name into the text box to the left of this button and then pressing this button will create a new directory on the server.
8. (C)reate File: This will cause a new empty file to be created that has the name typed into the text box
9. (R)ename File: This will cause the selected file or folder to be given the new name typed into the text box

10. Upload File: This will allow files to be uploaded from the user's local machine to SORASCS. The user must first select the 'Browse…' button to locate the file, and then click the Upload File button to upload to SORASCS. To upload a directory, that directory must first be zipped on the local machine. When the file is uploaded, the right-most column in the file list will contain an 'Unzip' link, which will allow the zipped directory to be unzipped on the server.

3.2 The Palette Region

The palette region contains three main categories, and gives the user access to thin services, thick services, and service management facilities on SORASCS.



List of services:

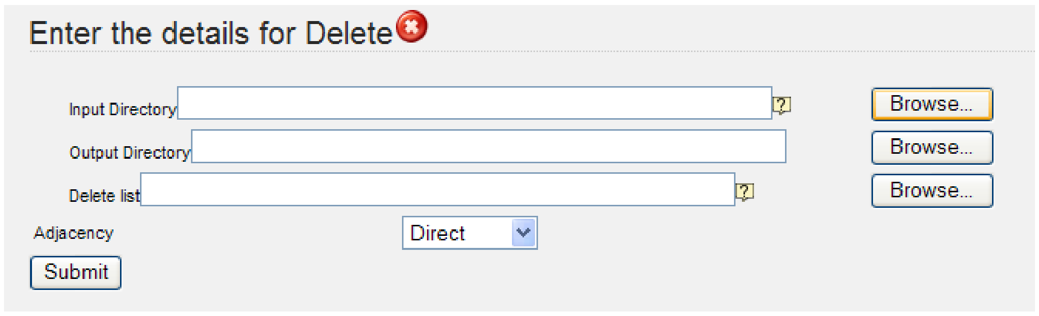
ReadableServiceName	Created By	Description
Anaphora	admin	View Description
Combine Thesauri	admin	View Description
Delete	admin	View Description
Extract Numerics	admin	View Description
Filter Text	admin	View Description
Format Case	admin	View Description
Generalization	admin	View Description
Import Text	admin	View Description
K Stem	admin	View Description
K Stemmer	admin	View Description
Keep	admin	View Description
Lowercase Concepts	admin	View Description
POS Attribute File	admin	View Description
POS Extraction	admin	View Description
Remove Extra Spaces	admin	View Description
Remove Numbers	admin	View Description
Remove Punctuation	admin	View Description
Remove Symbols	admin	View Description
Remove User Symbols	admin	View Description
Whiteout Numbers	admin	View Description
Whiteout Punctuation	admin	View Description

3.2.1 Services

For example, in the figure on the left, the user has selected the Preprocess item under the Services category. This presents them with a list of Preprocess services, with associated icons if they exist. The items and icons roughly follow the model from the CASOS Automap tool, and are broadly described as follows:

- *Preprocess*: Services that process plain text in some way, such as removing parts of it, applying thesauri, etc.
- *Procedures*: Services that enhance or merge existing files. For example, adding images to a network file, or sorting thesauri.
- *Extractor*: Extracts text files from non-text sources, such as blogs, twitter, facebook, etc.
- *Simulators*: Contains services that initiate SORASCS simulations.
- *Tasks*: Contains services that are combinations of other services, allowing for workflows to be accessed.
- *Visualize*: Services that visualize networks as images are covered by this category.
- *Report*: Services that generate reports on networks are contained in this category.
- *Merge*: Services that merge numerous files into one file (such as merging thesauri) are contained here.
- *Converter*: Converters convert one format of information to another. For example, services that convert DynetML files to files that can be read by Pythia are part of this category.
- *Generate*: Services that take some information and generate networks can be found here.

Clicking on the name of the service displays a form for the service that allows users to fill out more information for calling the service. For example, clicking 'Delete' in the figure above causes the form below to be displayed.

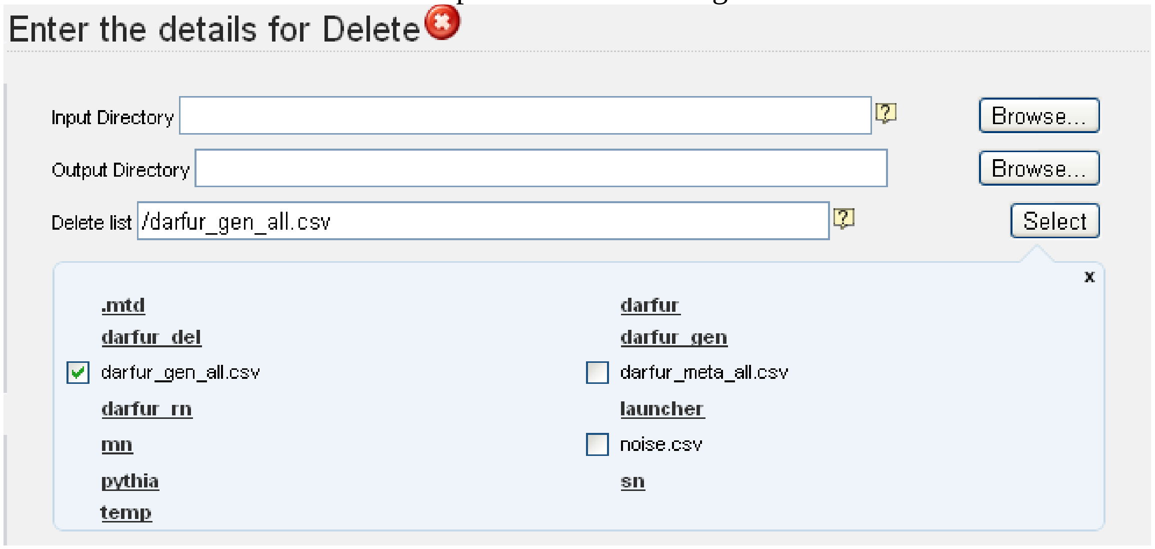


Enter the details for Delete

Input Directory		Browse...
Output Directory		Browse...
Delete list		Browse...
Adjacency	Direct	

Submit

Hovering over (when available) will display a description of the information to be filled in. Some forms are multiple pages, in which case the Submit button is replaced by a button labeled 'Next>' until last page is not displayed. If a parameter to be filled in is a file or a directory, a Browse… button is displayed to the right of the text box. Clicking this button will raise a file browser that will allow the user to select the type of file that is associated with that input. For example, in the image below displays a blue region wit the list of the appropriate files displayed. Selecting a file (by checking one of the checkboxes) will fill in the correspoding parameter. To hide the region, press the 'Select' button, which will commit the selection. Pressing the button labled 'x' in the top right of the file browser will cause the selection of the file to be canceled and the information restored to what it was prior to the file being selected.



Enter the details for Delete

Input Directory		Browse...
Output Directory		Browse...
Delete list	/darfur_gen_all.csv	Select

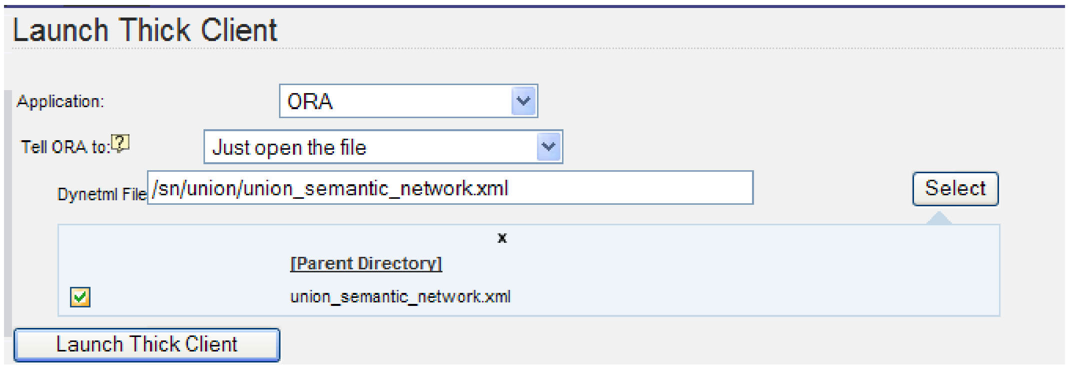
x

.mtd	darfur
darfur_del	darfur_gen
✓ darfur_gen_all.csv	☐ darfur_meta_all.csv
darfur_rn	launcher
mn	☐ noise.csv
pythia	sn
temp	

Finally, pressing the Submit button will cause the associated service to be invoked on SORASCS. In most cases, the user will be returned to the home page. If there is an error in calling the service, the error will appear in a red stripe above the File browser on the home page.

3.2.2 Thick clients

Thick clients are software programs registered by users with their local SORASCS client (see Section 4). They provide a means for SORASCS to launch programs on a user's machine, while still allowing the users to have control over which programs are available for SORASC to direct. The list of thick clients that are available to the portal are derived by querying the SORASCS client running on the users machine. If there is no client running on the user's machine, this list will be empty.



Launch Thick Client

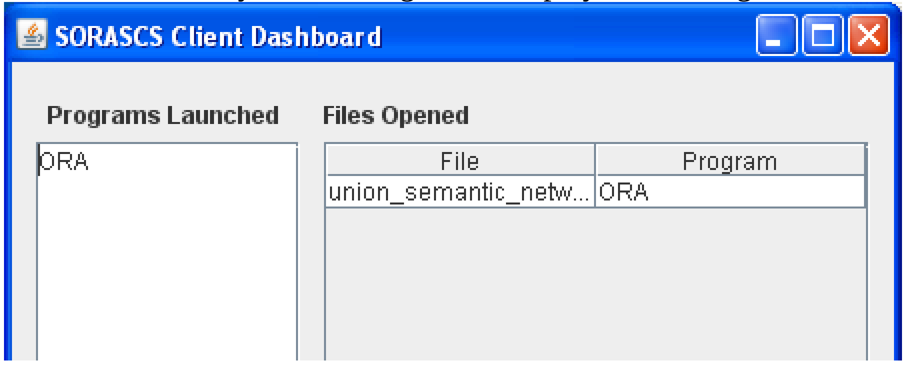
Application:	ORA
Tell ORA to:	Just open the file
Dynetml File	/sn/union/union_semantic_network.xml

✓ union_semantic_network.xml

The screenshot above shows an example of a form that might come up when selecting a thick client. This form allows the user to select files on the SORASCS server that can be used by programs running on their machine.

Pressing the "Launch Thick Client" button will cause the SORASCS client to receive a command telling it which program to launch. It will also download any files that are required from the SORASCS server.

The SORASCS client dashboard displays both the programs that have been launched by SORASCS and the files that they are working on, as displayed in the figure below.



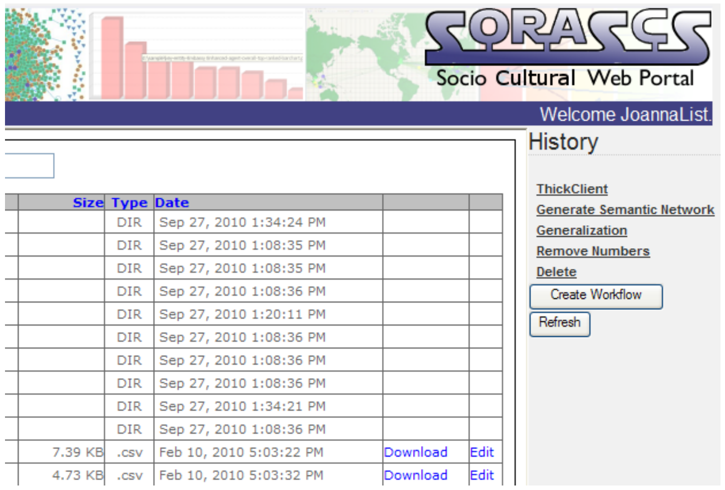
SORASCS Client Dashboard

Programs Launched: ORA

Files Opened

File	Program
union_semantic_netw...	ORA

When the thick client has finished, the entry will disappear. If the file has been changed by the program, it will be synchronized with the file on the server. Note that if the program creates additional files, they will need to be uploaded separately by the user to SORASCS.

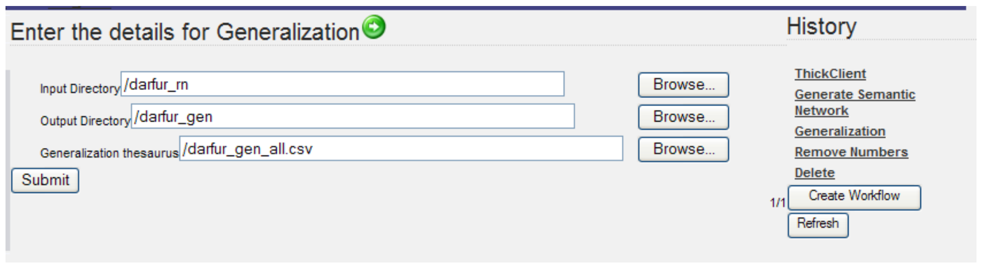


Size	Type	Date		
	DIR	Sep 27, 2010 1:34:24 PM		
	DIR	Sep 27, 2010 1:08:35 PM		
	DIR	Sep 27, 2010 1:08:35 PM		
	DIR	Sep 27, 2010 1:08:36 PM		
	DIR	Sep 27, 2010 1:20:11 PM		
	DIR	Sep 27, 2010 1:08:36 PM		
	DIR	Sep 27, 2010 1:08:36 PM		
	DIR	Sep 27, 2010 1:08:36 PM		
	DIR	Sep 27, 2010 1:34:21 PM		
	DIR	Sep 27, 2010 1:08:36 PM		
7.39 KB	.csv	Feb 10, 2010 5:03:22 PM	Download	Edit
4.73 KB	.csv	Feb 10, 2010 5:03:32 PM	Download	Edit

3.3 The History Region

The History region contains a list of all the services that the user has interacted with, the most recent service being at the top of the list. Only the 10 most recently used services are displayed. If more than 10 services have been used by the users, then a link is displayed that can be clicked on to view all services. If the user has invoked a workflow (i.e., a services that is a composition of other services), the workflow name will be displayed and the user will be able to expand the entry to display the services that were executed as part of the workflow.

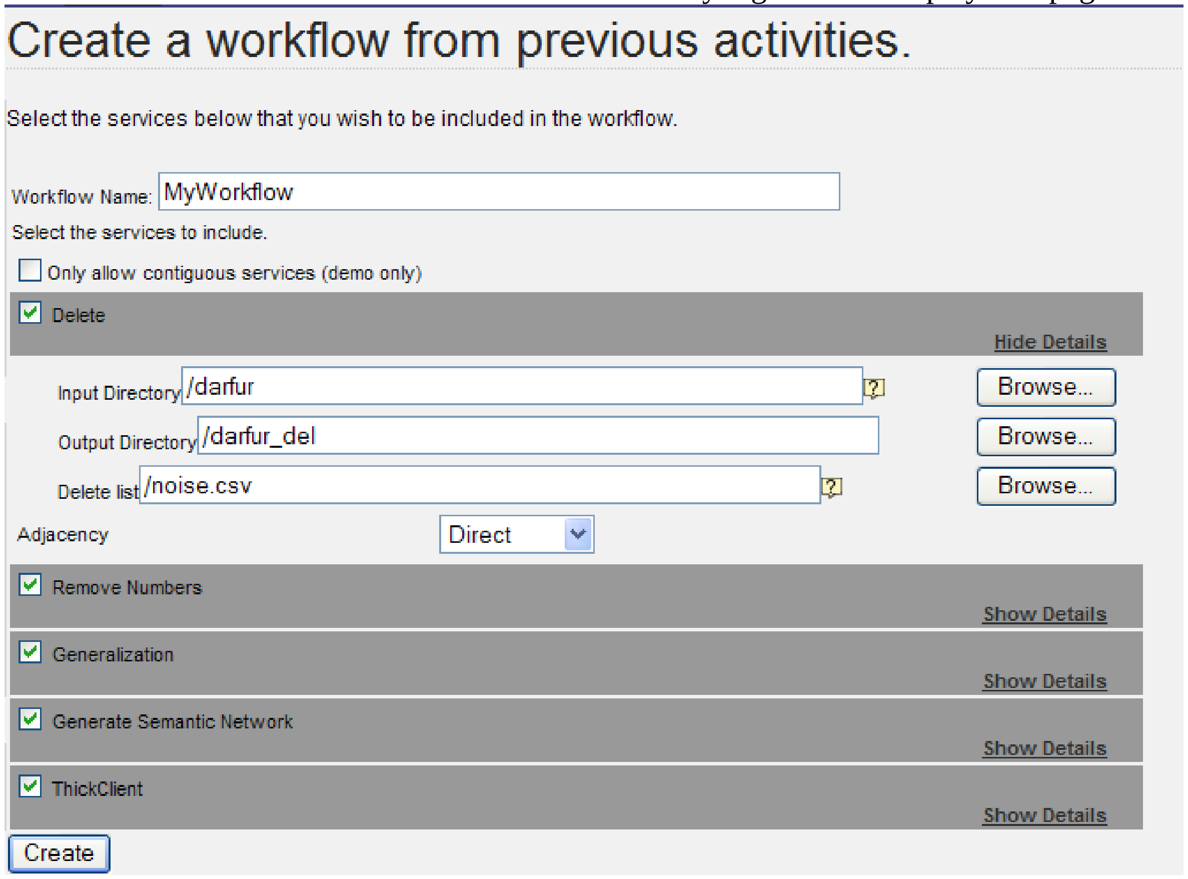
Clicking on any of the entries in the history list will cause the form for that service to be displayed, with the values that were passed to the service filled in.



Enter the details for Generalization

Input Directory	/darfur_rn	Browse...
Output Directory	/darfur_gen	Browse...
Generalization thesaurus	/darfur_gen_all.csv	Browse...

As a conceptual demonstration of constructing workflows, the history mechanism can also be used to package up the workflows that the user has invoked into one workflow. To do this, the user clicks on the "Create Workflow" button in the history region. This displays the page below.



Create a workflow from previous activities.

Select the services below that you wish to be included in the workflow.

Workflow Name: MyWorkflow

Select the services to include.

☐ Only allow contiguous services (demo only)

✓ Delete

Input Directory	/darfur	Browse...
Output Directory	/darfur_del	Browse...
Delete list	/noise.csv	Browse...
Adjacency	Direct	

✓ Remove Numbers

✓ Generalization

✓ Generate Semantic Network

✓ ThickClient

Each service has an entry in the list. The user can then see and change parameter values by selecting the "Show details" link. Furthermore, the user must select a number of contiguous services to include and provide a name for the workflow. After clicking create, the workflow will appear under the "Tasks" category of the palette.

4 Accessing SORASCS

SORASCS is an open source platform that can be downloaded from http://specialprojects.casos.cs.cmu.edu/bin/view/SORASCS/WebHome. While it may be installed locally by users and run on their machines, there is also a public site that provides the latest releases of the services and the portal. To access the portal at this site, go to http://sorascs.casos.cs.cmu.edu:8080/SoraSCSPortal. You will need a username and password to access both of these sites. To get these, contact Prof. Kathleen Carley (kathleen.carley@cs.cmu.edu). These sites will be maintained at least until 2013.

References

[1] D. Garlan, K. M. Carley, B. Schmerl, M. Bigrigg and O. Celiku. Using Service-Oriented Architectures for Socio-Cultural Analysis. In Proceedings of the 21st International Conference on Software Engineering and Knowledge Engineering (SEKE2009), Boston, USA, 1-3 July 2009.

[2] B. Schmerl, M. Bigrigg, D. Garlan, K.M. Carley. Integrating Components into SORASCS. Carnegie Mellon University, School of Computer Science, Institute for Software Research Technical Report CMU-ISR-10-122, August 2010.