How to Use SWEET

(Extended Version)

Table of Contents

1. Introduction

SWEET is a web application-based MicroWSMO editor, which is implemented in two versions. The first version takes the form of a vertical widget displayed within a Web browser. It is very lightweight and can be used to directly annotate RESTful service descriptions visualized in the browser window. The second implementation is part of a fully-fledged dashboard application developed within the SOA4All[1] project, which supports users in performing different tasks related to the service lifecycle. It has some additional functionalities and is more stable. Both SWEET implementations share common main functionalities:

- Insertion of hRESTS microformat tags in the HTML service descriptions in order to mark service properties (operations, address, HTTP method, input, output and labels).
- Integrated ontology search for linking semantic information to service properties.
- Insertion of MicroWSMO model reference tags, pointing to the associated semantic meaning of the service properties.
- Saving of semantically annotated HTML RESTful service description.
- Extraction of MicroWSMO service descriptions based on the annotated HTML.

Each of SWEET's implementations addresses different user needs and supports different use cases. The browser-based lightweight version is suitable for users who are browsing for RESTful services and want to directly annotate the currently displayed description. On the other hand, the extended version of SWEET, is suitable in cases where the user performs multiple tasks, and needs some additional functionalities which make the annotation process easier.

Even though, both implementations share the same core functionalities, the extended SWEET version has some additional features. It includes a three-based view, which visualizes the annotated service and enables the deleting and editing of annotations. In addition, the insertion of hRESTS tags is made easier for the users by enabling only the tags, which can be used in the current status of the application, and disabling all other tags. In this way, the user is intuitively supported in making correct service annotations. The domain ontology functionalities are also improved, by implementing a newer version of Watson's API[2] and using paginated requests, which reduces the waiting times. Finally, the semantic service property annotations can be deleted, which is not possible in the lightweight version of SWEET.

[1] http://www.soa4all.eu/

[2] http://watson.kmi.open.ac.uk/

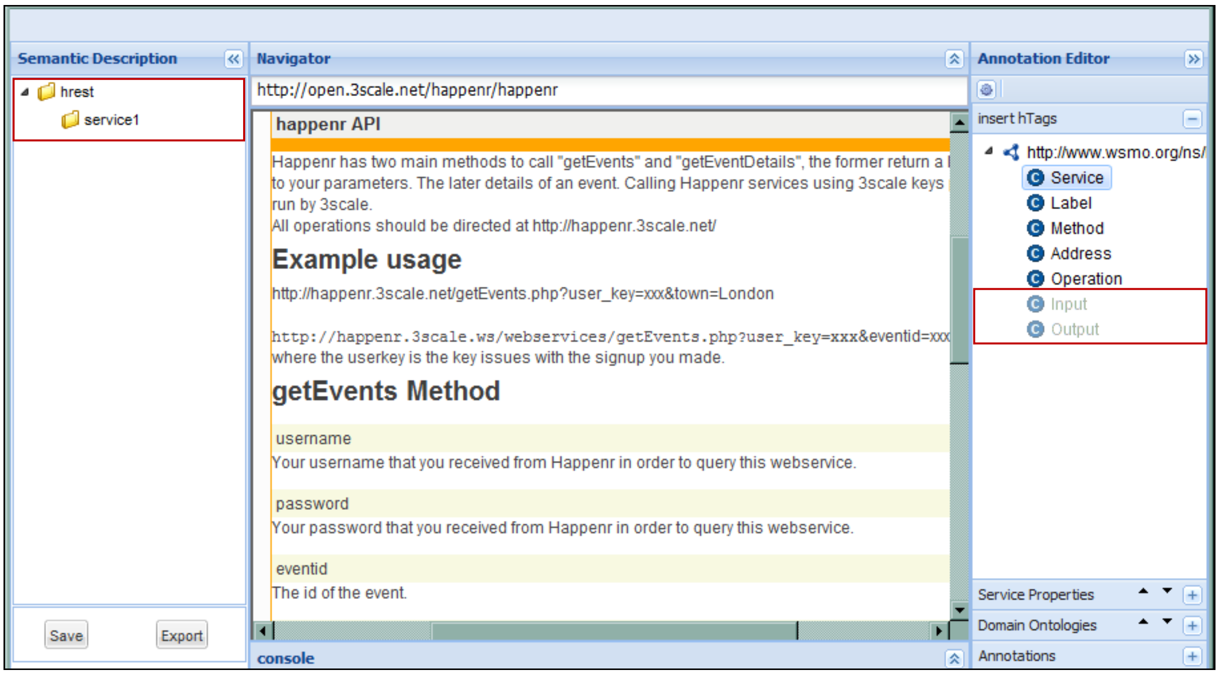
2. Documentation

2.1 Installation and Configuration

The here presented extended version of SWEET requires no installation or configuration. It can be started in a Web browser and is directly ready for use. However, it is recommended to use it with Firefox[3] browser.

2.1.1 How to use SWEET

SWEET is a web application-based MicroWSMO editor. The extended version of SWEET is based on the ExtGWT[4] technology and has more visualization components than the lightweight implementation. Figure 1 visualizes the GUI of SWEET. It consists of three main panels, including the *Semantic Description* panel, the *Navigator* panel and the *Annotation Editor* panel. It provides the same common functionalities, including: indentifying of service properties by inserting hRESTS tags in the HTML service description; searching for domain ontologies suitable for annotating the service properties; annotating service properties with semantic information and saving or exporting the annotated RESTful service. However, it provides a number of additional features, which make the using of the tool more user-friendly and the operation more stable.



http://open.3scale.net/happenr/happenr

happenr API

Happenr has two main methods to call "getEvents" and "getEventDetails", the former return a l
to your parameters. The later details of an event. Calling Happenr services using 3scale keys
run by 3scale.
All operations should be directed at http://happenr.3scale.net/

Example usage

http://happenr.3scale.net/getEvents.php?user_key=xxx&town=London

```
http://happenr.3scale.ws/webservices/getEvents.php?user_key=xxx&eventid=xxx
```
where the userkey is the key issues with the signup you made.

getEvents Method

username
Your username that you received from Happenr in order to query this webservice.

password
Your password that you received from Happenr in order to query this webservice.

eventid
The id of the event.

Figure 1: Extended Version of SWEET

Indentifying service properties by inserting hRESTS tags

The main difference between the lightweight version and this one is that here the service description is visualized within the editor (in the *Navigator* window) instead of in the browser itself. The insertion of hRESTS tags is done in the same way, by selecting the text describing the service property and clicking on the corresponding node in the hRESTS tags tree. However, there are a number of main important additional features that need to be

[3] http://www.mozilla-europe.org/en/firefox/

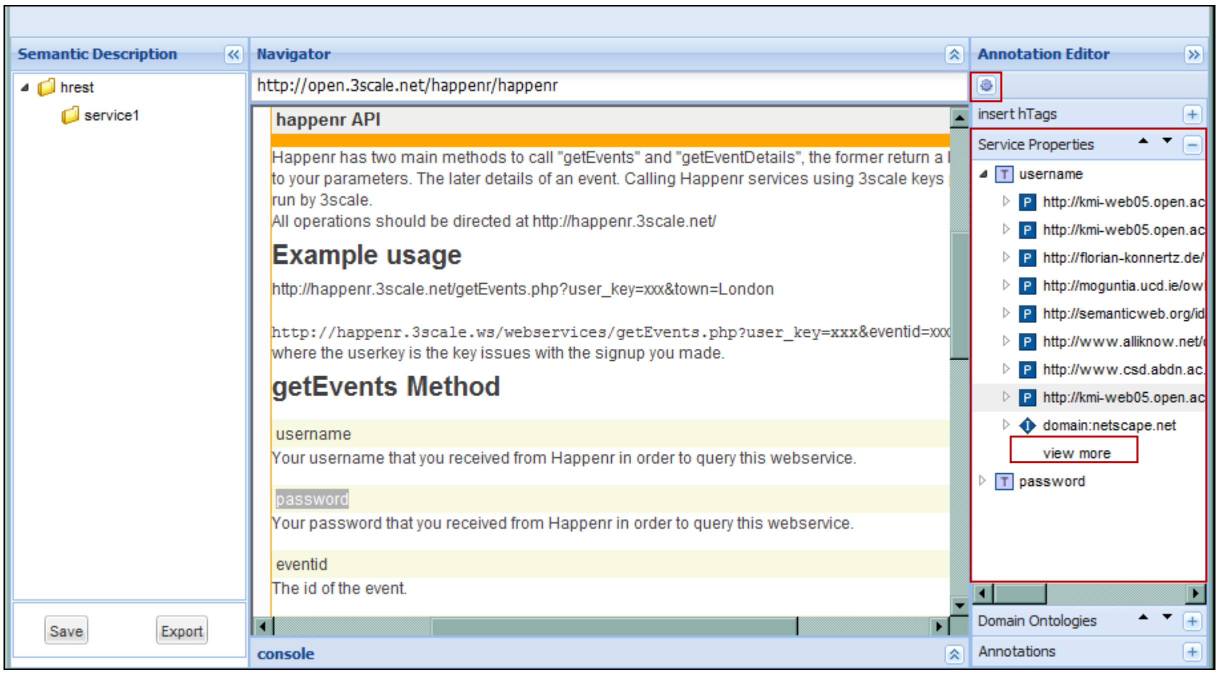
[4] http://extjs.com/products/gxt/

mentioned. First, each of the tag nodes are enabled and disabled depending on whether the user can select them in the current status of the service annotation or not. For example, the Input node is disabled, if there are no marked operations in the service description. In this way, the user is guided through the process of marking service properties and it is ensured that the resulting service structure is correct.

Second, the inserted hRESTS tags are represented in a tree structure in the *Semantic Description* panel, which tracks the user's annotation actions. The tree shows, which service properties are already marked and if the user recognizes a mistake, he/she can delete the corresponding hRESTS node and remark it.

Searching for suitable domain ontologies

The domain ontologies search is based on Watson, however, this implementation uses a newer and a faster version of the API. The user can search for suitable domain ontologies for a given service property by selecting it and clicking on the search icon (Figure 2). The results are displayed in the *Service Properties* panel. This panel has a number of additional useful features. First, the retrieving of ontologies matches is faster and is paginated. If the first set of ontology results is insufficient, the user can search for more results by clicking on "view more". Second, the search is session based and the user preserves his/her ontology search while annotating different service descriptions. Third, all search results can be collapsed or expanded, by clicking on the up and down arrows on top of the panel. This makes browsing of the ontology matches easier. Finally, the search is restricted with a timeout so that if there are problems with Watson's API or the response is taking too long, the application will not be blocked.



http://open.3scale.net/happenr/happenr

happenr API

Happenr has two main methods to call "getEvents" and "getEventDetails", the former return a
to your parameters. The later details of an event. Calling Happenr services using 3scale keys
run by 3scale.
All operations should be directed at http://happenr.3scale.net/

Example usage

http://happenr.3scale.net/getEvents.php?user_key=xxx&town=London

http://happenr.3scale.ws/webservices/getEvents.php?user_key=xxx&eventid=xxx
where the userkey is the key issues with the signup you made.

getEvents Method

username
Your username that you received from Happenr in order to query this webservice.

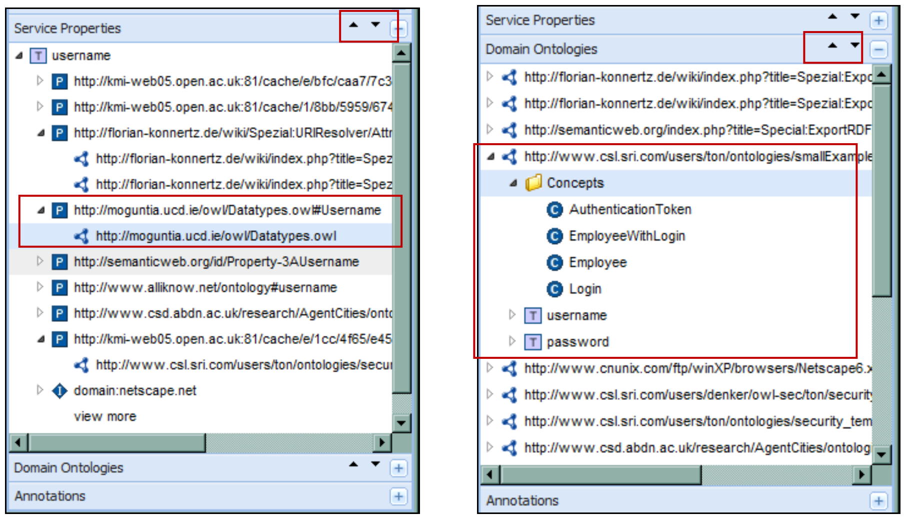
password
Your password that you received from Happenr in order to query this webservice.

eventid
The id of the event.

http://kmi-web05.open.ac
http://kmi-web05.open.ac
http://florian-konnertz.de/
http://moguntia.ucd.ie/ow
http://semanticweb.org/id
http://www.allknow.net/
http://www.csd.abdn.ac.
http://kmi-web05.open.ac

Figure 2: Service Properties Panel

The implementation of the *Service Properties* and *Domain Ontologies* panels supports the user in choosing a suitable ontology for annotating the individual service properties of the complete RESTful service. These supporting functionalities are visualized in Figure 3. The user can view the URI of each of the matching concepts, properties or instances and the corresponding ontology. Additional information is available in the *Domain Ontologies* panel, which shows all service properties that can be annotated with one particular ontology as well as a list of all concepts. The entries of both panels can be expanded or collapsed in order to ease the navigation.



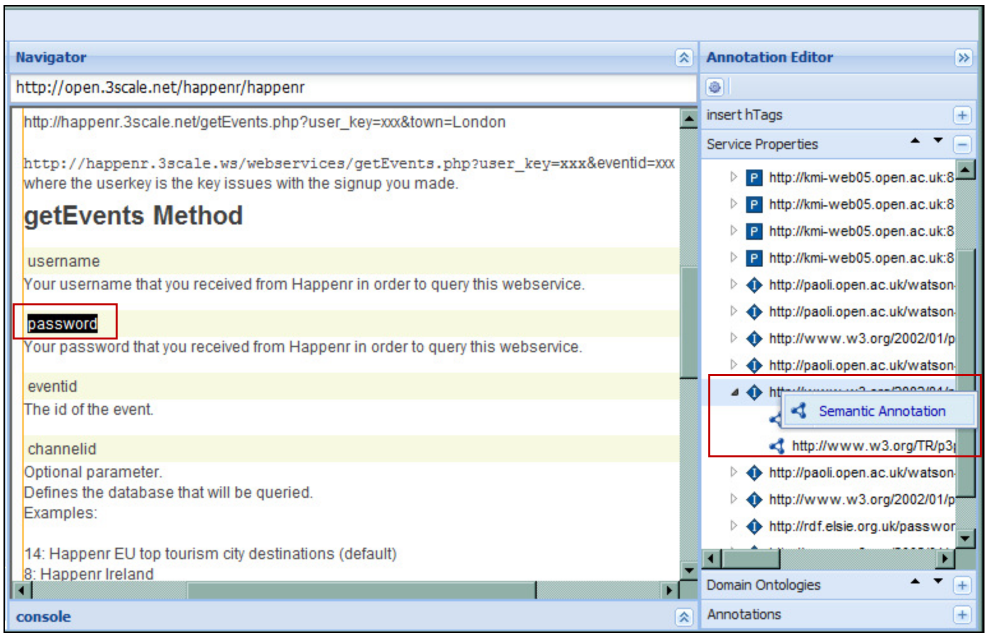
Service Properties
username
http://kmi-web05.open.ac.uk:81/cache/e/bfc/caa7/7c3
http://kmi-web05.open.ac.uk:81/cache/1/8bb/5959/674
http://florian-konnertz.de/wiki/Spezial:URIResolver/Attr
http://florian-konnertz.de/wiki/index.php?title=Spez
http://florian-konnertz.de/wiki/index.php?title=Spez
http://moguntia.ucd.ie/owl/Datatypes.owl#Username
http://moguntia.ucd.ie/owl/Datatypes.owl
http://semanticweb.org/id/Property-3AUsername
http://www.allknow.net/ontology#username
http://www.csd.abdn.ac.uk/research/AgentCities/onto
http://kmi-web05.open.ac.uk:81/cache/e/1cc/4f65/e45
http://www.csl.sri.com/users/ton/ontologies/secu
domain:netscape.net
view more
Domain Ontologies
Annotations

Service Properties
Domain Ontologies
http://florian-konnertz.de/wiki/index.php?title=Spezial:Expo
http://florian-konnertz.de/wiki/index.php?title=Spezial:Expo
http://semanticweb.org/index.php?title=Special:ExportRDF
http://www.csl.sri.com/users/ton/ontologies/smallExample
Concepts
AuthenticationToken
EmployeeWithLogin
Employee
Login
username
password
http://www.cnunix.com/ftp/winXP/browsers/Netscape6.x
http://www.csl.sri.com/users/denker/owl-sec/ton/security
http://www.csl.sri.com/users/ton/ontologies/security_tem
http://www.csd.abdn.ac.uk/research/AgentCities/ontolog
Annotations

Figure 3: Exploring Domain Ontologies

Annotating service properties

Once the user has decided, which ontology to use for the service property annotation, he/she can do an annotation by selecting the part of the service and clicking on "Semantic Annotation" in the *Service Properties* context menu. An example annotation is visualized in Figure 4. The result of the annotation is the same as when using the lightweight version of SWEET: the model and href tags are inserted in the HTML to mark the association of the particular HTML element with the semantic concept.

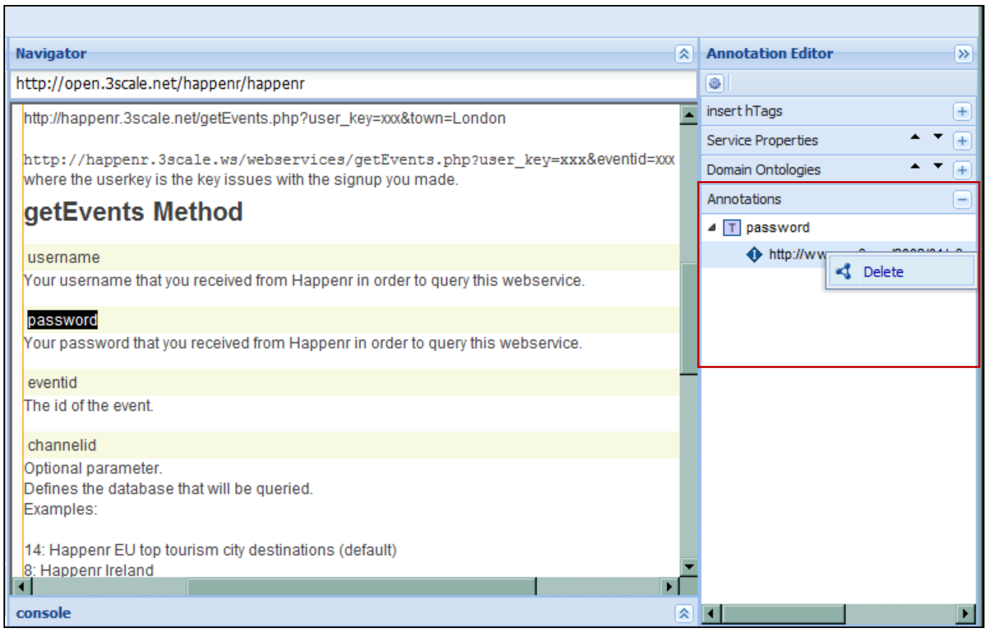


Navigator
http://open.3scale.net/happenr/happenr
http://happenr.3scale.net/getEvents.php?user_key=xxx&town=London
http://happenr.3scale.ws/webservices/getEvents.php?user_key=xxx&eventid=xxx
where the userkey is the key issues with the signup you made.
getEvents Method
username
Your username that you received from Happenr in order to query this webservice.
password
Your password that you received from Happenr in order to query this webservice.
eventid
The id of the event.
channelid
Optional parameter.
Defines the database that will be queried.
Examples:
14: Happenr EU top tourism city destinations (default)
8: Happenr Ireland
console

Annotation Editor
insert hTags
Service Properties
http://kmi-web05.open.ac.uk:8
http://kmi-web05.open.ac.uk:8
http://kmi-web05.open.ac.uk:8
http://kmi-web05.open.ac.uk:8
http://paoli.open.ac.uk/watson
http://paoli.open.ac.uk/watson
http://www.w3.org/2002/01/p
http://paoli.open.ac.uk/watson
Semantic Annotation
http://www.w3.org/TR/p3
http://paoli.open.ac.uk/watson
http://www.w3.org/2002/01/p
http://rdf.elsie.org.uk/passwor
Domain Ontologies
Annotations

Figure 4: Inserting Semantic Annotations

A summary of the already made annotations it given in the *Annotations* panel. An additional functionality of this panel is that annotations can be removed by choosing "Delete" (Figure 5) from the context menu. In this way, the user can remove incorrect annotations and substitute them with new ones without having to reload the tool and start the annotation process from the very beginning.



http://open.3scale.net/happenr/happenr

http://happenr.3scale.net/getEvents.php?user_key=xxx&town=London

http://happenr.3scale.ws/webservices/getEvents.php?user_key=xxx&eventid=xxx
where the userkey is the key issues with the signup you made.

getEvents Method

username
Your username that you received from Happenr in order to query this webservice.

password
Your password that you received from Happenr in order to query this webservice.

eventid
The id of the event.

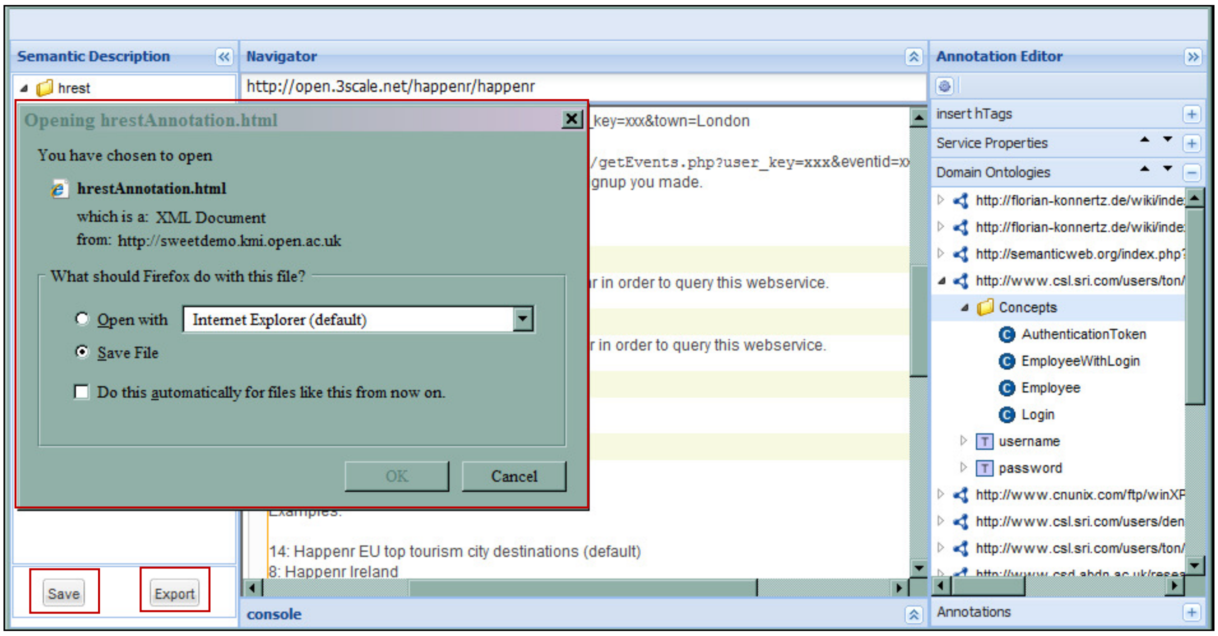
channelid
Optional parameter.
Defines the database that will be queried.
Examples:

14: Happenr EU top tourism city destinations (default)
8: Happenr Ireland

Figure 5: Deleting Semantic Annotations

Saving or exporting the annotated RESTful service

Finally, when the user is finished with annotating the HTML RESTful service description with hRESTS and MicroWSMO tags, the resulting annotated service can be saved or exported by clicking on the "Save" or "Export" buttons. Figure 6 shows that when the user clicks on "Save", a pop-up window is opened, which allows him/her to choose a destination for saving the annotated HTML. The export provides a RDF transformation of the annotated HTML.



Opening hrestAnnotation.html

You have chosen to open

hrestAnnotation.html

which is a: XML Document
from: http://sweetdemo.kmi.open.ac.uk

What should Firefox do with this file?

Open with Internet Explorer (default)

[✓ Save File]

Do this automatically for files like this from now on.

http://open.3scale.net/happenr/happenr

key=xxx&town=London

/getEvents.php?user_key=xxx&eventid=x
gnup you made.

r in order to query this webservice.

r in order to query this webservice.

14: Happenr EU top tourism city destinations (default)
8: Happenr Ireland

http://florian-konnertz.de/wiki/inde
http://florian-konnertz.de/wiki/inde
http://semanticweb.org/index.php?
http://www.csl.sri.com/users/ton/
http://www.cnunix.com/ftp/winXP
http://www.csl.sri.com/users/den
http://www.csl.sri.com/users/ton/

Figure 6: Saving Annotated HTML

The result is the MicroWSMO description of the semantically described RESTful service

Further functionalities of the extended version of SWEET

It is important to point out that SWEET provides options for customizing the way service descriptions are viewed. First, if the *Navigator* panel displays services, which already contain MicroWSMO elements, these elements will be recognized and automatically highlighted so that the user can manipulate them and integrate them in his/her own annotation of the service. Second, the way the service properties and semantic information is highlighted can modified very easily by simply substituting the current CSS file with a new one, which uses different text font and colours.