



Users documentation

Poznań, Poland, 2006-08-08

Authors:
Maciej Gawinecki
Paweł Kaczmarek

Project site:
<http://sourceforge.net/projects/e-travel>

Contact:
maciej.gawinecki@ibspan.waw.pl

Agent-based Travel Support System
Copyright (C) 2006 Maciej Gawinecki & Pawel Kaczmarek

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Setting up the system

The system was tested on a single machine with MS Windows XP and therefore it provides only batch files for running the system. Users of Linux and Unix systems are encouraged to prepare own shell scripts, similar to those in *bin/* directory.

Preparing environment

Runtime platform

Runtime environment:	JRE 1.5 http://java.sun.com
Agent platform:	JADE 3.4 http://jade.tilab.com
Semantic framework:	JENA 2.4 http://jena.sourceforge.net
Access to OWL via JavaBeans	Jastor 1.0.3 http://jastor.sourceforge.net
RDF/XML to HTML (WML) Transformer:	Raccoon server (Rx4RDF 0.6.0) http://www.liminalzone.org/Raccoon
	Prerequisite: <ul style="list-style-type: none">• Python 2.3, http://www.python.org/• 4Suite XML library 1.0b3, http://4suite.org/• RDFLib, http://www.rdfliib.net/
Relational Database for Persistent Model Storage:	PostgreSQL 8.0 http://www.postgresql.org/
JDBC Connector:	JDBC 3 for Version 8.0 http://jdbc.postgresql.org/

Configuring access to database

1. Create a database account, which will be used for the system to authorize access to the database.
2. Create database (e.g. named *tss*), where the system will store persistent data models. Account created in step 1 must be granted all privileges to this database.
3. Configure access to the database in file: *etc/db-conn.properties*.
4. Configure path with JDBC driver in *bin/setenv.bat* file.

Configuring Raccoon server

1. Install Raccoon together with prerequisites.
2. Configure Python root path in *bin/setenv.bat* file.
3. Fix `$PYTHONDIR/Lib/site-packages/rx/RxPathDom.py` file by putting the patch somewhere about line 1838, by changing the following instructions:

```
def invokeRxSLT(RDFPath, stylesheetPath):
```

```
#_4suiteModel, db = RxPath.deserializeRDF( RDFPath )
#model = RxPath.FtModel(_4suiteModel)

uri = RxPath.Uri.IndexPathToUri(modelPath)
model = RxPath.MemModel(RxPath.parseRDFFromURI(uri))
rxPathDom = RxPath.createDOM(model)
stylesheetContents = file(stylesheetPath).read()
return RxPath.applyXslt(rxPathDom, stylesheetContents)
```

into these (note the line with the patch comment):

```
def invokeRxSLT(RDFPath, stylesheetPath):
    #_4suiteModel, db = RxPath.deserializeRDF( RDFPath )
    #model = RxPath.FtModel(_4suiteModel)

    modelPath = RDFPath # PATCH

    uri = RxPath.Uri.IndexPathToUri(modelPath)
    model = RxPath.MemModel(RxPath.parseRDFFromURI(uri))
    rxPathDom = RxPath.createDOM(model)
    stylesheetContents = file(stylesheetPath).read()
    return RxPath.applyXslt(rxPathDom, stylesheetContents)
```

Configuring CLASSPATH

1. Adjust paths to root directory of Jena, JADE and Jastor directories in *bin/setenv.bat* file.

Launching the system

1. From *bin/* directory call *run-tss.bat* script.

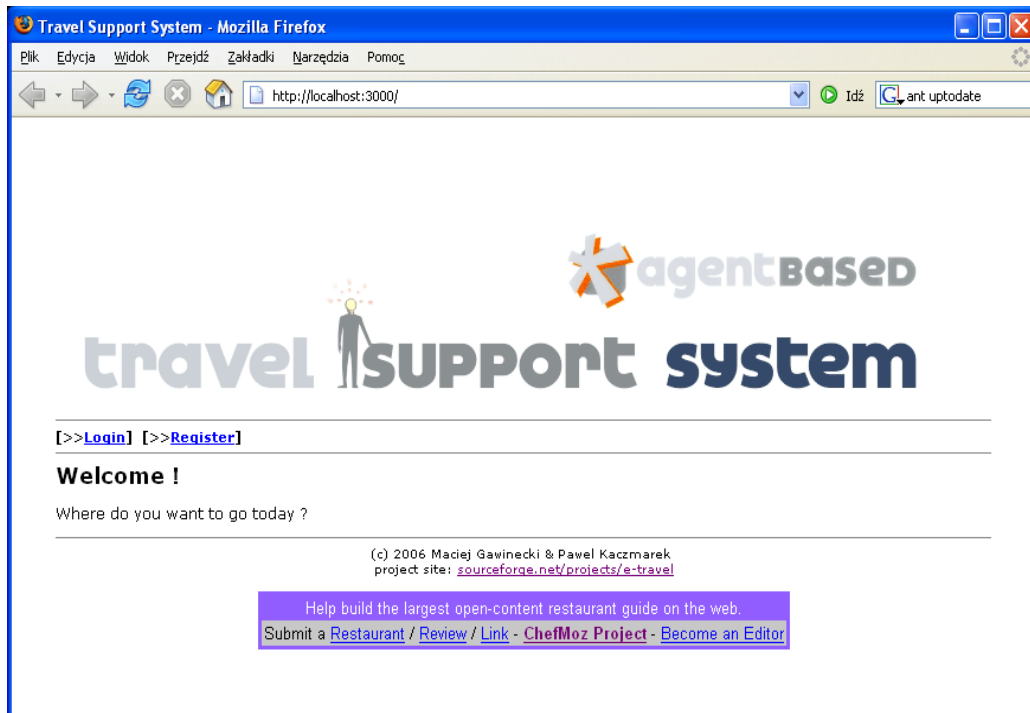
```
tss> cd bin
tss\bin> run-tss
```

Two console windows should be opened: one with Raccoon server and the one system starting. Please be patient, since for the first time systems must load all necessary data into database. Especially loading database with restaurants is time consuming (about 3-4 hours).

When the systems finished initializing without any problem, you will be informed by following information on the system console:

```
2006-08-27 06:25:21 ibspan.tss.agents.pra.http.MainServer <init>
INFO: HTTP Server for html-media listening on port 3000 ...
2006-08-27 06:25:21 ibspan.tss.agents.pra.http.MainServer <init>
INFO: HTTP Server for wml-media listening on port 3001 ...
```

2. When the systemsFrom *bin/* directory launch *browse-tss.url* file, which will open start page of the system in your favourite web browser:



3. Surf on!

Acknowledgements

The authors would like to thank you the following person for their contribution Wiktor Moderau (for dig-art), Mateusz Dominiak (for support for individuals in Jastor), Adam Souzis and Michał Olczak (for support in configuration of Raccoon in Python), Mateusz Kruszyk (for stereotypes design), Michał Szymczak (for design of restaurant ontology and referenced domains), Minor Gordon (for vital design ideas), dr Maria Ganzha for mathematical considerations and diagrams validation and many thans to Marcin Paprzycki for his enforcing power to do it. Also many thanks for people from very active jena-dev mailgroup and also for authors of Raccoon, JADE and Jastor for their support.