

Translation Client for Idiom WorldServer

(SW project proposal)

Motivation

There are a lot of commercial and open-sourced tools used for translation / globalization of SW projects. Commercial tools used by large SW companies for professional translations contain rich, high-tech functionalities. Those tools are closed and expensive. In contrary, open-sourced solutions can be used mainly for translations of open-sourced projects by communities. Such tools are freeware, simple to use, open and developed and supported usually by communities.

Sun Microsystems Globalization Center is trying to connect its professional translation services based on commercial solution from Idiom WorldServer [1] with open source communities and open-tools used by them. A key problem is to find out the way how to connect external contributors to internal professional systems, how to leverage rich, non-opened services to external contributors.

Project Goals

The goal is to design and develop a simple, light-weight, multi-platform translation editor connecting the open-source community translators with Idiom WorldServer, Sun's internal globalization management system. Such a thin client allows the community to indirectly utilize the powerful features of WorldServer, which are not available in any free Computer Aided Translation (CAT) software, and at the same time it enables centralized configuration, administration and maintenance of all translation projects managed by WorldServer.

From the functional point of view, the application fulfills all requirements for online translation editor. The user interface must be intuitive and user friendly. The application will have clear and adequate user documentation. As a tool for localization, the tool itself must be internationalized, i.e. it must be possible to localize the interface to other language than English. It also has to support various scripts, right-to-left writing, etc. The new client may be an extended version of existing pilot project developed by Sun Microsystems [2].

The successful project will fulfill the following:

- requirements specified in [Technical Project Specification section](#);
- formal project requirements - [section Formal Project Specification](#);
- the project will be deployed and fully functional

Technical Project Specification

Main features of the translation client will be:

- ability to translate a segmented text
- access, search and edit in translation memory

- ability to import/export translation kits for the most popular translation editors (*)
- authentication and RBAC (Role Based Access Control)
- self registration
- support for translation glossary/terminology management
- ability to execute external tools (e.g. pdf files generation)
- fuzzy matches suggestions
- handling of concurrent edits

Gathering feedback and analysis of requirements is one part of this work. Therefore, the set of features is not fixed and may increased during development based on feedback from community.

The translation client will be web-based application. The back end server is WorldServer. All data presented and created in translation client will be stored in WorldServer. Communication protocol between translation client and WorldServer will be web service based. Description of web services - endpoints and schema - will be provided to translation client developers. Web services at WorldServer are developed by Sun Microsystems. We expect new web services will be added to WorldServer based on new requirements during translation client development.

We assume that translation client will be deployed at the same machine as WorldServer web services not to violate *Same Origin Policy* for AJAX/Javascript.

(*) A part of this task is to make a research, identify and analyze the most popular translation editors and the format they support. The outcome is that the developed translation editor can import/export translation kits that will be compliant with the most popular editors.

Formal Project Specification

The project will fulfill the following formal aspects:

- The project will be OpenSource project with own versioning system
- Public project web-sites will be created
- Bug-reporting system will be created and used
- Technical, Installation and User Documentation will be provided
- Project documentation (Project log) will be provided

Expected Work Progress

The following steps are expected:

1. Collection of the user requirements and prioritization
2. Identify what will be in the next release (Sun approval is required)
3. Development: Analysis, Implementation, Testing
4. Release deployment
5. Requirements revision and Project log update
6. Continue with step #2

An iterative development model is expected for this project. It means that at certain milestones (we expect monthly-based release cycle), the actual version of the client is taken, deployed and new cycle of development is started. It also helps development team to gather a feedback from real users and those comments/requests for enhancements can be implemented in further releases of the project. It is also expected that up-to-date user documentation is delivered with every release together with a

Project log update (the team will need to track project changes for each release).

The team

Expected number of students: 4-5

Release Date

The project should start on Jan 2009 and should finish within 9 months, i.e. 09/2009.

References

- [1] Idiom World Server: <http://www.idiominc.com/products>
 - [2] Sun Community Translation Interface: <http://cti.sunvirtuallab.com>
-