Supplementary Specification for SST (Search String Translator)

Revised Inception Draft

September 29-30, 2004

Supplementary Functional Specifications

Environments

Every translation will take place in a search environment that includes a set of operations and features permitted in the source string and a set of supported search engines.

Environment developers can include instructors, course coordinators, and librarians.

Instructors and course coordinators should be able to create and modify a set of restricted instructional environments that will promote student learning on specific search-related topics.

Librarians should be able to create a default environment for library patrons tailored to the resources (like searchable subscription databases) that are available in the library and the needs of patrons. Librarians should also be able to create specialized environments for patrons with specialized needs.

The system should come packaged with two standard environments:

- The maximal environment includes all of the currently supported search engines and all of the currently supported operations and features.
- The minimal environment includes a set of popular search engines together with the operations and features supported by all of the included engines. (The point is to guarantee translation for (almost) all correctly formed source strings.)

Source String Format

A source string shall consist of a subject string separated followed by a set of zero or more restrictions. The available operations and restrictions are governed by the search environment.
A **subject string** is a series of **search terms** separated by **operations**. Operations include the Boolean operations AND, OR, and NOT (all capitals required), and the NEAR operation. The precedence is as follows: NOT has the highest precedence, then OR, then NEAR, and finally AND. The Boolean operation NOT may precede any search term. Unless the environment requires all operations to be entered explicitly, the default operation is AND, so that two juxtaposed search terms will be joined by AND.

A **search term** is a word (format to be specified later) or quoted phrase. A quoted phrase consists of one or more words separated by spaces enclosed in quotation marks.

**Restrictions** include the following:

- **Domain restrictions** are indicated by the keyword **domain**, followed by a colon, followed by a series of domains separated by Boolean operations. Some examples follow:
  - **domain**: .edu – restricts the domain to URLs ending in “.edu”.
  - **domain**: .edu OR .org – restricts the domain to URLs ending in “.edu” or “.org”.
  - **domain**: www.montclair.edu – restricts the domain to www.montclair.edu (i.e., only sites within this domain will be returned).
  - **domain**: NOT.com AND NOT .org – restricts the domain to sites whose domain does not end in “.com” or in “.org”.

- **Date restrictions** are indicated by the keyword **date**, followed by a colon, followed by a range of dates. Some examples follow:
  - **date**: after 4/1/2003
  - **date**: before 2/13/1999
  - **data**: from 3/1/03 to 6/1/04

- **Element restrictions** restrict the search to the specified part of a web page. They include, for example, **inURL**, **inTitle**, etc. (*List is incomplete.*) Each of these keywords is followed by a **subject string**.

**BNF Specification for source strings**

*This is a formal specification of the required form of source strings. It should be completed during the elaboration phase.*
Tutorials

When creating or modifying a tutorial, an Instructor should be able to specify different search environments for different sections of the tutorial. She should be able to create or modify environments as part of the process of constructing a tutorial.

She should be able to include the following elements in a tutorial:

- Explanatory test
- Screen shots from the Translate Search String use case
- Exercises for Students
- Responses to Students’ answers to exercises (based on common errors)

She should also be able to set up the tutorial to record student responses. This can be either in a summary form (without data on individual Student responses) for use in evaluating the effectiveness of the tutorial, or in a complete form (with individual response data) for use in evaluating each student.

Non-functional specifications

Usability: The intended users will have varying degrees of fluency in using information technology. A relatively naïve user should be able to use the basic functionality of the SST with minimal instruction. The SST should provide clear explanations when it is not possible to translate a source string for a particular target search engine. On-line documentation should be available and clear to beginners.

Accuracy: The SST should provide correct translations of the source string in 99% of instances in which translation is possible. It should correctly identify the sources of the problem for 99% of untranslatable source strings.

Response Time: The SST should return translations of 99% of translatable source strings in under one second.

Resource Usage: For translation, the translation process should run locally on the user’s computer after the search environment. It should be possible to configure the system to store environments on a server or locally. Storage of search environments should take up minimal space in the server.

Maintainability: Since search engine formats and features change frequently, it should be possible to adapt to most changes with no modifications to existing code. At worst, minimal modifications – mostly in the form of additional code – should be required. This issue will need to be revisited in the anticipated second version of the SST.
**Extensibility:** An anticipated future version of the SST will send the translated search strings to the target engines and display the results of the searches. The system should be designed to easily accommodate this extension.

**Security:** Only authorized users should be able to create, modify, or delete search environments.

**Implementation:** The SST will be open-source code written in the Java programming language.