Proposal for adding a search engine into the repository

by David Rozas (david.rozas@gmail.com)

Introduction

In order to continue with the implementation of the use cases related to the searching process, new functionalities need to be added to the RepositoryManager.
This document proposes the possibility of creating a search engine into the RepositoryManager, based on the use of the open source library Lucene[1].

Use Cases

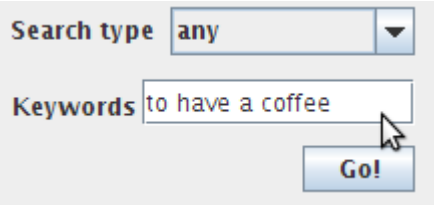
At least the following use cases (from the document "Design of the EUT: Application Manager") can be covered: D42_UC-REP04, D42_UC-REP05 and D42_UC-REP06. It would cover all the use-cases related to text-search, and it would be totally compatible with the searching functionalities based on ontologies (ex.: recommended applications).

Brief explanation

A brief explanation of the steps involved in the process is detailed next:

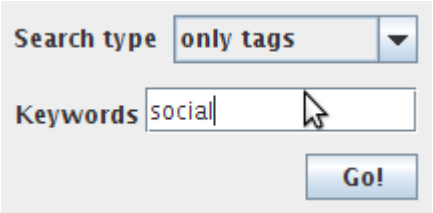
- ***Indexing***
  - An Index Search of all the meta-information related to the applications stored in the RepositoryManager is performed when this bundle is started
  - This Index is updated dynamically, by analyzing the meta-information of every applications which is shared (added) or removed from the Repository.
  - The information to analyze will be structured in different fields: description, associated tags, user, etc. In this way, we have a flexible way to perform queries based on different criteria. Depending on the kind of field, different operations can be performed to have an analyzed information of better quality (Ex.: remove articles and prepositions from the description).
- ***Querying***
  - The Repository will offer a transparent way to perform the queries. The services can be matched with the options selected by the user in the GUI (ApplicationManager GUI, PHP EUT, etc.) to perform the queries easily.

- Ex.
  - A sentence in natural language with filtering by no criteria:



  - would be transformed into HAVE or COFFEE in ANY

- Ex.:
  - A sentence in natural language with filtering by only tags



  - would be transformed into SOCIAL in Field=”associated_tags”

It has to be analyzed more carefully (and no screenshots can be shown yet), but a similar approach can be taken to perform a search by similarity:

- An user selects a local application which wants to be compared.
- Some local meta information of this application is sent to the repository (ex.: description), and some public information is recovered from the repository itself (ex.: public tags associated to that application).
- The Repository performs an analysis, compares with the applications and returns a list of the most similar ones.

Interface

A first approach to the new methods offered by IRepositoryManager is:

```
public String[] search(String query, String criteria);

public String[] searchBySimilarity(String appId, String appDescription);
```

Why using Lucene?

- It has a great performance.
- It is quite flexible and easy to extend if more functionalities are needed in the future.
- It is cross-platform.
- It is open-source.

References

- [1] http://lucene.apache.org/java/docs/

The last version of this document can be found at: http://basar.idi.ntnu.no/svn/astra/WP3/Y3/SOA.2.0/ApplicationManager/doc/search-engine-proposal/