CommunityMashup - Basics

This page describes the basics of the CommunityMashup.

Structure

A CommunityMashup (instance) consists of

- a data set (DataSet)
- a set of sources for filling and extending the data set
- a set of interfaces for accessing the data set

Sources go through the following life cycle: fill (create InformationObjects), extend, update

When Items are created in the DataSet the CommunityMashups tries to merge them with other InformationObjects created earlier. For merging the mashup uses the isEqualItem() function defined for the Item class (and overloaded in classes derived from the Item class) - See package org.sociotech.communitymashup.data.impl for the implementations of the different item types. Checks currently implemented for determining if to merge:

- all Items: check if ident is equal or identifiers are equal,
- Person: checks if any pair of name and alternativeNames matches (ignoring case) - thereby name = firstname + " " + lastname (if defined)
- Organisation: checks if any pair of name and alternativeNames matches (ignoring case)

A special source is the CommunityMashupSource - which can be used to load data from one mashup instance to another. One use case for this is to have one server based mashup for loading and merging all the data - and one mashup instance in the client application (e.g. the CommunityMirror) that loads the data from the server and makes it available locally.

A CommunityMashupContainer can contain one or more CommunityMashup instances.

Data model

The data set can store any number of **Item** objects.

- **Item** (attributes: ident:String, uri:String, stringValue:String, xmlValue:String, created:Date, lastModified:Date)
  (additionally Items have a reference to the data set, to **Identifier** objects (key/value pairs), to meta tags (Strings)
  - ident is automatically set when an Item is added to the dataset
- Sub classes (of Item):
  - **InformationObject**
  - **Identifier** (attributes: key:String, value:String)
  - **Extension**
  - **Classification**
  - **MetaTag** (attributes: name:String, references to metaTagged:Item*)

The core object class for storing information is **InformationObject**:

- **InformationObject** (additional attributes: name:String, alternativeNames:String*, verifiedName:Boolean)
  (additionally InformationObjects have a reference to categories, tags, images, binaries, metaInformations, starRankings, thumbRankings, viewRankings)
- Sub-classes (of InformationObject):
  - **Person** (additional attributes: title:String, lastname:String, firstname:String, dateOfBirth:Date)
    (references to leaderOf:Organisation*, authored:Content*, contributed:Content*, persons:Person*, ranked:Ranking*)
  - **Organisation** (additional attributes: NONE)
    (references to leader:Person, parentOrganisation:Organisation, participants:Person*, organizations:Organisation*)
  - **Content** (attributes: locale:String)
    (references to contents:Content*, contributors:Person*, author:Person, documents:Document*, parentContent:Content, videos:Video*)

InformationObjects (persons, organisation, content) can be further extended by **Extension** objects:

- **Extension** (no additional attributes)
  (additionally Extensions have a reference to tags:Tag*)
- Sub classes (of Extension)

- **MetaInformation** (references to: informationObjects:InformationObject*)
  - **Event** (additional attributes: date:Date)
  - **Location** (additional attributes: street:String, houseNumber:String, zipCode:String, country:String, longitude:String, latitude:String, city:String, state:String)
    References to: indoorLocations:IndoorLocation*
  - **IndoorLocation** (additional attributes: name:String)
    References to: location:Location, parentIndoorLocation:IndoorLocation , indoorLocations:IndoorLocation*
  - **Email** (additional attributes: dress:String)
  - **Phone** (additional attributes: areaCode:String, countryCode:String, number:String)
  - **WebSite** (additional attributes: adress:String, title:String, shortenedUrl:String)
  - **WebAccount** (additional attributes: username:String, service:String)
  - **InstantMessenger** (additional attributes: username:String)
- **Ranking** (additional attributes: date:Date, reference to ranker:Person)
  - **StarRanking** (additional attributes: normalizedValue:Double, reference to rankedInformationObject:InformationObject)
  - **ViewRanking** (reference to rankedInformationObject:InformationObject)
  - **ThumbRanking** (reference to rankedInformationObject:InformationObject)
- **Attachment** (additional attributes: fileUrl:String, cachedFileUrl:String, cachedOnly:Boolean, fileExtension:String, fileIdentifier: String, cachedFileName:String, noCache:Boolean)
  - **Transformation** (reference to transformed:Content*)
  - **Document** ()
  - **Video** ()
  - **Image** (additional attributes: width:Integer, height:Integer)
  - **Binary** (additional attributes: bytes:ByteArray)

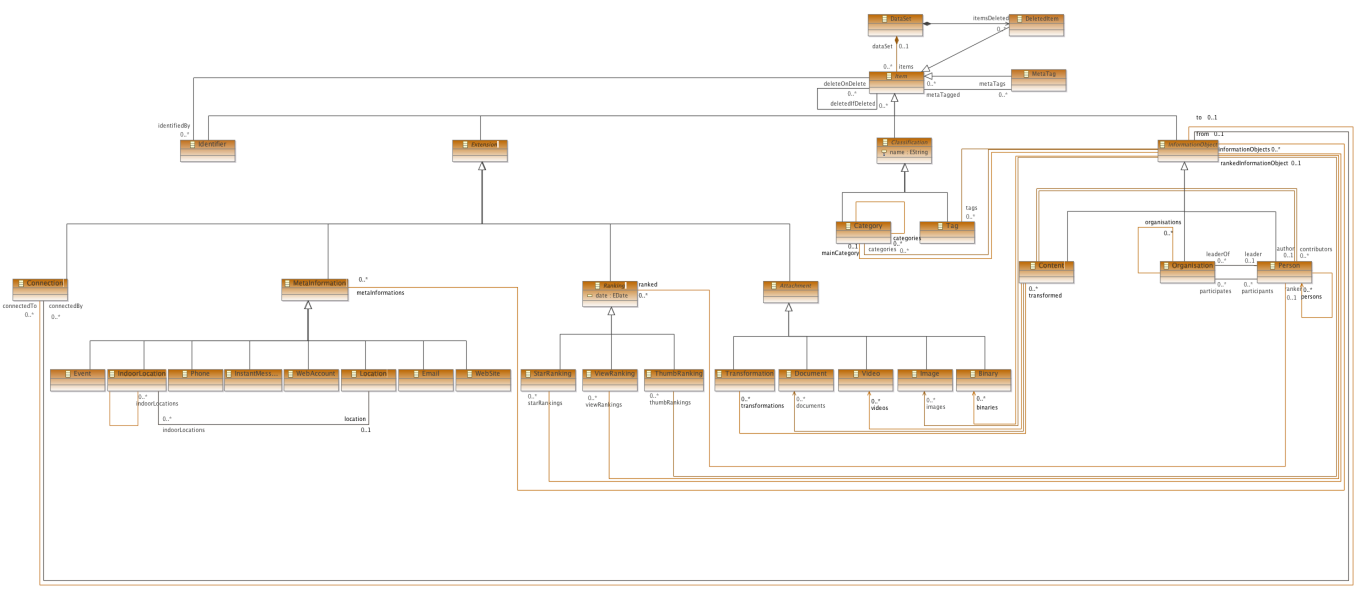
InformationObjects can be linked to each other using **Connection** objects (a sub-class of Extension):

- **Connection** (Item attributes: ident:String, uri:String, stringValue:String, created:Date, lastModified:Date)
  (additionally Connections have a reference to an InformationObject:from, and to an InformationObject:to)

InformationObjects can be classified using **Classification** objects:

- **Classification** (additional attributes: name:String)
- Sub classes of Classification
  - **Category** (additional references to categorized:InformationObject*, parentCategory:Category, categories:Category*, mainCategorized:InformationObject*)
  - **Tag** (additional reference to tagged:InformationObject*)

The complete data model currently (because it can be extended - e.g. by new Extension sub classes) looks like:



Publications

The basic concepts of the CommunityMashup have been documented in this publication: Lachenmaier, P., & Ott, F. (2011). Building a Person-Centric Mashup System - CommunityMashup: A Service Oriented Approach. In D. Eichhorn, A. Koschmider, & H. Zhang (Eds.), Proceedings of the 3rd Central-European Workshop on Services and their Composition (ZEUS'2011) (pp. 122–129). Karlsruhe, Germany: CEUR-WS.org. Retrieved from http://ceur-ws.org/Vol-705/paper17.pdf.

Additional publication on the CommunityMashup:

- Lachenmaier, P. (2016). Personenzentrierte Integration sozialer Daten – Ein modellgetriebener Entwicklungsansatz. Dissertation. Fakultät für Informatik, Bundeswehr University Munich.
- Lachenmaier, P., Ott, F., & Koch, M. (2013). Model-driven development of a person-centric mashup for social software. Social Network Analysis and Mining, 3(2), 193–207. doi:10.1007/s13278-012-0064-x
- Lachenmaier, P., Ott, F., Immerz, A., & Richter, A. (2011). CommunityMashup - A Flexible Social Mashup Based on a Model-Driven-Approach. In R. Alhajj, J. Joshi, & M.-L. Shyu (Eds.), Proceedings of the 12th IEEE International Conference on Information Reuse and Integration (IRI'2011) (pp. 48–51). Las Vegas, NV: IEEE. doi:10.1109/IRI.2011.6009519