NoTube
Networks and Ontologies for the Transformation and Unification of Broadcasting and the Internet

FP7 – 231761

D3.2 User Modelling Service, v.1

Coordinator: Michele Minno
With contributions from: Libby Miller, Balthasar Schopman
Quality Assessor: Lora Aroyo
Quality Controller: Michele Minno

<table>
<thead>
<tr>
<th>Document Identifier:</th>
<th>NoTube/2009/D3.2/Vx.x</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Deliverable:</td>
<td>NoTube EU-IST-2009-231761</td>
</tr>
<tr>
<td>Version:</td>
<td>version 1.0</td>
</tr>
<tr>
<td>Date:</td>
<td>December 10, 2010</td>
</tr>
<tr>
<td>State:</td>
<td>final</td>
</tr>
<tr>
<td>Distribution:</td>
<td>public</td>
</tr>
</tbody>
</table>
EXECUTIVE SUMMARY

This document is the first version of D3.2 - User Modelling Service.
DOCUMENT INFORMATION

<table>
<thead>
<tr>
<th>IST Project Number</th>
<th>Acronym</th>
<th>NoTube</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP7 – 231761</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Full Title</th>
<th>Project URL</th>
<th>Document URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Networks and Ontologies for the Transformation and Unification of Broadcasting and the Internet</td>
<td><a href="http://www.notube.tv/">http://www.notube.tv/</a></td>
<td>EU Project Officer Leonhard Maqua</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Deliverable Number</th>
<th>Deliverable Title</th>
<th>Work Package Number</th>
<th>Work Package Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
<td>User Modelling Service v1</td>
<td>3</td>
<td>User and Context model Specification</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date of Delivery</th>
<th>Contractual</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M23</td>
<td>31-12-2010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Status</th>
<th>Nature</th>
<th>Dissemination Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>version 1.0</td>
<td>prototype ☑ report ☐ dissemination ☐</td>
<td>public ☑ consortium ☐</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Authors</th>
<th>(Partner)</th>
<th>Resp. Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michele Minno (Pronetics), Libby Miller (BBC), Balthasar Schopman (VU)</td>
<td>Michele Minno (Pronetics)</td>
<td>E-mail <a href="mailto:michele.minno@pronetics.it">michele.minno@pronetics.it</a></td>
</tr>
<tr>
<td>Partner Pronetics</td>
<td>Phone +xx (xxx) xxx-zzzz</td>
<td></td>
</tr>
</tbody>
</table>

Abstract (for dissemination)

This document describes the status of work of user profiling and recommendation services within the NoTube architecture.

Keywords

Semantic Web, profiler, recommendations, TV, activities, SKOS, Object Property

Version Log

<table>
<thead>
<tr>
<th>Issue Date</th>
<th>Rev No.</th>
<th>Author</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>December 10, 2010</td>
<td>1</td>
<td>Michele Minno</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Important information for Quality Assessors/Quality Controllers

(Please delete in the EC-ready version)

- Send a PDF version AND a Latex/MS-Word of the deliverable to the Quality Assessor/Quality Controller to ensure that no links are broken, pictures are correctly converted, etc.,
- ALL pictures in the deliverable must be B&W readable,
- After addressing the Quality Assessor’s comments, report back to him/her using the review form.
## PROJECT CONSORTIUM INFORMATION

<table>
<thead>
<tr>
<th>Participant’s name</th>
<th>Partner</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vrije Universiteit Amsterdam</td>
<td></td>
<td>Griss Schreiber&lt;br&gt;Phone: +31 20 598 7739/7718&lt;br&gt;Email: <a href="mailto:schreiber@cs.vu.nl">schreiber@cs.vu.nl</a></td>
</tr>
<tr>
<td>British Broadcasting Corporation</td>
<td></td>
<td>Libby Miller&lt;br&gt;Phone: +44 787 65 65 561&lt;br&gt;Email: <a href="mailto:Libby.Miller@bbc.co.uk">Libby.Miller@bbc.co.uk</a></td>
</tr>
<tr>
<td>Asemantics SRL Uninominale</td>
<td></td>
<td>Alberto Reggiori&lt;br&gt;Phone: +39 0639 7510 78&lt;br&gt;Email: <a href="mailto:alberto@asemantics.com">alberto@asemantics.com</a></td>
</tr>
<tr>
<td>Engin Medya Hizmetleri A.S.</td>
<td></td>
<td>Ron van der Heiden&lt;br&gt;Phone: +31 6 2003 2006&lt;br&gt;Email: <a href="mailto:ron@engin.tv">ron@engin.tv</a></td>
</tr>
<tr>
<td>Institut fuer Rundfunktechnik GmbH</td>
<td></td>
<td>Christoph Dosch&lt;br&gt;Phone: +49 89 32399 349&lt;br&gt;Email: <a href="mailto:dosch@irt.de">dosch@irt.de</a></td>
</tr>
<tr>
<td>Ontotext AD</td>
<td></td>
<td>Atanas Kiryakov&lt;br&gt;Phone: +35 928 091 565&lt;br&gt;Email: <a href="mailto:naso@sirmag.bg">naso@sirmag.bg</a></td>
</tr>
<tr>
<td>Open University</td>
<td></td>
<td>John Domingue&lt;br&gt;Phone: +44 1908 655 014&lt;br&gt;Email: <a href="mailto:j.b.domingue@openg.ac.uk">j.b.domingue@openg.ac.uk</a></td>
</tr>
<tr>
<td>RAI Radiotelevisione Italiana SPA</td>
<td></td>
<td>Alberto Morello&lt;br&gt;Phone: +39 011 810 31 07&lt;br&gt;Email: <a href="mailto:a.morello@rai.it">a.morello@rai.it</a></td>
</tr>
<tr>
<td>Semantic Technology Institute</td>
<td></td>
<td>Lyndon Nixon&lt;br&gt;Phone: +43 1 23 64 002&lt;br&gt;Email: <a href="mailto:lyndon.nixon@sti2.org">lyndon.nixon@sti2.org</a></td>
</tr>
<tr>
<td>Stoneroos B.V.</td>
<td></td>
<td>Annelies Kaptein&lt;br&gt;Phone: +31 35 628 47 22&lt;br&gt;Email: annelies.kaptein@stoneroos</td>
</tr>
<tr>
<td>Thomson Grass Valley France SA</td>
<td></td>
<td>Raoul Monnier&lt;br&gt;Phone: +33 2 99 27 30 57&lt;br&gt;Email: <a href="mailto:raoul.monnier@thomson.net">raoul.monnier@thomson.net</a></td>
</tr>
<tr>
<td>TXT Polymedia SPA</td>
<td></td>
<td>Sergio Gusmeroli&lt;br&gt;Phone: +39 02 2577 1310&lt;br&gt;Email: <a href="mailto:sergio.gusmeroli@txtgroup.com">sergio.gusmeroli@txtgroup.com</a></td>
</tr>
<tr>
<td>KT Corporation</td>
<td></td>
<td>Myoung-Wan Koo&lt;br&gt;Phone: +82 2 526 6347&lt;br&gt;Email: <a href="mailto:mskim@kt.co.kr">mskim@kt.co.kr</a></td>
</tr>
</tbody>
</table>
# Table of Contents

## List of Figures

- 6

## List of Tables

- 7

## 1 Introduction

- 8

## 2 User Profiling - Update

- 9
  2.1 Overview ................................................. 9
  2.2 Resource classification .................................. 9
    - 2.2.1 Popularity ........................................... 10
  2.3 Choosing the set of resources for user profiling ................. 11
    - 2.3.1 Limitations of DBpedia Categories ..................... 11
    - 2.3.2 Alternatives: SKOS hierarchies ....................... 12
    - 2.3.3 Alternatives: non-SKOS graphs ....................... 12
  2.4 Extending the profile ..................................... 12

## 3 Recommending

- 13
  3.1 Overview ............................................... 13
  3.2 Recommendation space .................................... 13
  3.3 User Profile to item ...................................... 13
    - 3.3.1 SKOS ................................................ 14
    - 3.3.2 Property ............................................ 15
    - 3.3.3 Pattern ............................................. 17
  3.4 Item to item ............................................. 17
    - 3.4.1 Simple linking ...................................... 19
    - 3.4.2 Pairwise connectedness .............................. 19
    - 3.4.3 Fanning ............................................ 20

## 4 Beancounter New Features

- 22
  4.1 New features ............................................ 22
    - 4.1.1 Activity/Interest Ingest Service ...................... 22
    - 4.1.2 UI-oriented user services ......................... 22

## 5 Evaluation

- 23
  5.1 Evaluation dimensions .................................... 23
  5.2 User activities evaluation ................................ 23
    - 5.2.1 Technical evaluation ................................ 24
    - 5.2.2 End user evaluation ................................ 24
  5.3 User profile evaluation ................................... 24
    - 5.3.1 Technical evaluation ................................ 24
    - 5.3.2 End user evaluation ................................ 25
  5.4 Recommender(s) evaluation .................................. 25
    - 5.4.1 Technical evaluation ................................ 25
    - 5.4.2 End user ............................................ 25

## References

- 25
LIST OF FIGURES

3.1 Recommendation space ........................................... 14
3.2 SKOS recommender .................................................. 15
3.3 Property recommender ................................................ 15
3.4 Linked Data Cloud .................................................... 16
3.5 Patterns ................................................................. 18
3.6 Simple graph ........................................................... 19
3.7 Flattened graph ........................................................ 19
3.8 Pairwise connected .................................................... 20
3.9 Fanning ................................................................. 20
5.1 Evaluation space ....................................................... 23
LIST OF TABLES
1. Introduction

The aim of these pages is to briefly describe the status of work of Work Package 3 in NoTube at month 23, so at nearly two thirds of the project time period. Two broad work areas can be defined:

- User Profiling
- Recommending

Then the evolving of the implementation and some evaluation criteria are discussed.
2. User profiling - update

2.1 Overview

As described in the previous deliverable (D3.1 - 2th version), the user profile is built from activities the user performed among different social networks over the web. These atomic activities (such as: "User x liked item y", "User x listened to song z") are collected and semantically lifted. This means that heterogeneous activities performed on different social networks are aggregated according to an unique activity model. Then these activities are structured as instances of the concept 'Activity' using our activities ontology (http://xmlns.notu.be/air/). Therefore a list of RDF triples corresponding the activities are produced, linked to potential concepts/resources that may concern them and stored as RDF triples in the semantic triple store of the Bean-counter. Thus the resulting user graph will be a flat list of user activities, with activity objects (the item y, the song z) enriched with additional RDF triples linking these objects to equivalent (at least claimed as equivalent) resources described in some ontology of the Linked Data Cloud (http://linkeddata.org), mainly taken from DBpedia (http://dbpedia.org). Each user has one user graph, containing his/her own activities. Then another big graph is built, called ”user-profiles”, containing the user profiles of all users. These profiles are computed starting from the activity graphs, specifically from activity objects. As we have just seen, activity objects come with DBpedia resources, attached during the enrichment phase of the activity gathering. These resources are classified in DBpedia according to a commonly used ontology: SKOS. SKOS (http://www.w3.org/2004/02/skos/) stands for Simple Knowledge Organization System. It is a light ontology used to classify concepts by means of concept hierarchies, and to point the resources belonging to a certain concept. Each DBpedia resource has as subject some SKOS concept, linked through the 'skos:subject' object property. Thus each activity object will have one or more SKOS concepts coupled with it, reached browsing those properties. According to the occurrence frequency of a concept among the user activity objects, a weight of the concept is computed. Concepts and associated weights make the profile of the user, modeled using our weighted interests ontology (http://xmlns.notu.be/wi/). The resulting user profiles are potential inputs of one of the devised recommenders. I.e. they are currently used by the news recommender to find possible interest points in the news item y for the user x.

2.2 Resource classification

Some relevant work has been done to devise algorithms that from the list of all SKOS subjects belonging to a resource extract the sublist of relevant subjects or even just one subject that would describe that resource in the most specific and characteristic way. The main algorithm that have been created foresees an intermediate step to work. The idea is to find some special resources, one per SKOS concept, which will be considered as the most representative resources of the SKOS concepts. I.e.: Bill Gates could be considered as the most representative resource for SKOS concept: 'Business people in software'. Once found these resources, current common resources are compared with them. In particular, a resource r with ten SKOS subjects s1, .. , s10 will be compared with the ten special resources which are the most representative resources of SKOS
concepts respectively s1, ..., s10. The most delicate step is the computation of the most representative resources, as can be easily seen from the sketched description above. Following paragraphs show two adopted complementary approaches and a third one where the two approaches are simultaneously used in a mixed fashion.

2.2.1 Popularity

The simplest way to find the most representative resource is to find the most famous one. Among all resources classified with a certain SKOS concept the most popular among all the resource set is taken. To measure popularity we use the number of inbound semantic arcs. A resource with a high number of inbound semantic arcs is a resource that is object of many properties starting from any other resource. This is certainly a hint of the fact that this resource is well represented and 'well known' among all other resources in the dataset.

Cooccurrence

The second way rests upon a more complex approach, based on cooccurrences. Cooccurrence between two events is the probability that both events will happen together. In our case we compute cooccurrences between SKOS concepts. If two concepts happen to be together in a lot of resource classifications, perhaps these two concepts are semantically 'near'. Looking at all SKOS concepts of all resources, we can compute clusters of concepts. Each cluster represents a set of concepts often found together among resources categories. I.e.:

- American_comedians
- American_film_actors
- American_television_actors
- Best_Drama_ACTor_Golden_Globe_(film)_winners

A cluster like this one exists if a lot of resources have been categorized using these concepts together. Thus each resource will have a set of clusters of concepts (or parts of those clusters), all together covering all the SKOS concepts which describe it. A resource that owns the totality of a cluster and nothing more (apart from very generic spotted clusters about human beings, places, and so on) is a good candidate according to this second approach for being the most representative one for all SKOS concepts belonging to that cluster. It can be noticed that in this case a very representative resource of a cluster C (and thus of all concepts belonging to C) is a resource that is classified only with concepts in C, so it is for sure not the most famous one, but maybe the most specific one.

Mixed mode

A combined solution has been successfully adopted, in which both approaches are exploited. Two weights are used to set the specific relevance of popularity (L) and specificity by cooccurrence values (C) of each resource, so that the final value against which resources are measured for aspiring at the most representative is:
\[ a \times L + b \times C \]

where \( a \) and \( b \) are rational numbers among 0 and 1

In this way a good mix of popularity and specificity is what is required for the most representative resource.

A test set has been run to obtain the best \( a \) and \( b \) values. A set 100 iterations have been run, each with different values according to the following law:

\[ a = x, \quad b = 1 - x \quad (0 \leq x \leq 100) \]

Results showed that popularity is much more important than specificity by cluster cooccurrence, even if it alone doesn't assure good classifications.

2.3 Choosing the set of resources for user profiling

The choice of resources which have to describe and represent the user profile is a very crucial one. From that choice it depends all the good or bad behaviour of the recommender (at least of recommenders which leverage user profiles in order to work). As we have seen, our choice fell on SKOS concepts used in DBpedia to classify resources. In the following paragraphs some considerations about this choice and other possible alternatives are drawn.

2.3.1 Limitations of DBpedia Categories

Concepts in DBpedia which are the target of skos:subject arcs can easily act as user profile interests. This has been our choice in the project until now. It is a very straightforward solution, since in DBpedia, which is the main ontology we are inspecting in order to find semantic connections between things, skos:subject property is broadly used; almost every resource in DBpedia has got at least one SKOS concept used to describe it. Nevertheless this solution hides some relevant drawbacks. First of all, it requires to trust a classification, that, how ever clever it could be, is just a possible classification, and therefore a possible bottleneck for errors and biases. Secondly, and related to this, it doesn’t leverage connections between resources, but only connections between a resource and a concept. Thus the power of the semantic graph remains not fully exploited. As third drawback, we have to remark the fact that these SKOS concepts are sometimes too compact. Labelling one concept as 'Cancer_death_in_1987', for instance, we are putting into the label too much meaning and complexity: it is in any case a string, with no semantic structure, and therefore all this complexity lies there as an untangled bundle. This is a direct consequence from the provenance of the DBpedia Categories, ie that it is directly derived from Wikipedia Categories. From the perspective of the Recommender System, the SKOS hierarchy of the DBpedia Categories are a disadvantage. The semantic distance between concepts that are connected via the skos:broader relation are highly variable and unpredictable. Furthermore, the DBpedia Categories that are often used for classifying TV programs are very generic, eg dbpedia:Living_People. This is the outcome from qualitative analysis of a functioning recommender system.
2.3.2 Alternatives: SKOS hierarchies

For the exposed drawbacks, alternative or complementary ways of representing user profile have been investigated. Together with skos:subject relations, another aspect of SKOS Ontology that can be leveraged is the set of hierarchies among concepts made of skos:broader and skos:broaderOf chains. These chains link together concepts which can be considered as semantically included one into the other. I.e.: Woman (skos:broader) Person (skos:broader) Living Being. Starting from user profile concepts and browsing these chains, more relevant concepts can be further added to the profile. Thinking at the third drawback of the original SKOS solution, these additional concepts can provide the profile with some concepts which explode the complexity of the starting concept, i.e.:

- Cancer death in 1987
- → Cancer death
- → death

Collecting the latter two concepts together with the original one, a more generic placement of it can be achieved, useful to set its ‘semantic boundaries’.

2.3.3 Alternatives: non-SKOS graphs

Leaving SKOS aside, other ontologies can be leveraged to build a user profile. For example, the full DBpedia namespace can be used to identify user interests. This increases the expressiveness of the User Profile substantially. Among Linked Data collected ontologies other useful properties are present, like rdf:type or the Freebase ontology type. Unfortunately they suffer from the same drawbacks found for the SKOS ontology choice.

2.4 Extending the profile

Some data is missing in the model of the profile. The most important one is the context of the interest. According to the wi ontology, any interest comes with some information about the context of the interest itself, specified in term of time, place, device or evidence. The only part that has been implemented is the evidence till now. It gathers the reasons why that interest is present among user profile interests. Time, place and device are missing, whereas they are of great importance for any recommender.
3. Recommending

3.1 Overview

Recommendation concerns the core part of the research works. There are different strategies which have been devised. The main common point among all of them is the idea that both media items to be recommended and user interests are universally and unambiguously identified across the web with URIs and interlinked via other identified resources defined somewhere else. In this sense the different algorithms described in the following sections aim all at leveraging some knowledge already out there on the web, that takes its strength from the established connections among resources belonging to different datasets and ontologies. Apart from this common point, each strategy differs from the other ones in terms of what requires as input and which ‘semantic arcs’ chooses to follow.

3.2 Recommendation space

We can identify a recommendation space, made of the two dimensions just mentioned:

- input
- graph or single arcs to browse

Two are the possible inputs to the recommender:

- the user profile: a set of couples (interest, weight) or simply a set of user activities shaped as (verb, object, context)
- a media item: the recommender passes from a given media item to another one, considered as similar for some reason.

Regarding the possible graphs to browse, three different strategies have been investigated:

- skos:subject arc browsing
- any property arc browsing
- specific sequence of property (pattern) browsing

3.3 User Profile to item

This class of recommenders provide the user with personalized recommendations, computed leveraging the profile of the user, made of weighted interests, or simply of his/her activities.
3.3.1 SKOS

What we leveraged in this context for the SKOS recommender is at first blush just the skos:subject arc that links a resource to the concept it belongs. Thus starting from both user activities and media item provided by the content owners, the recommender looks for skos:subject arcs, contained mainly in the DBpedia triple set. Actually, on the user side, the recommender has available already the SKOS concepts themselves, because user profile is a list of SKOS concepts computed following the skos:subject arcs coming from the activity object resources. Then, once this two lists of concepts are there, the recommender simply tries to find concepts in common among the two lists. The number of common concepts relating to the same media item affects the weight of the recommendation of that specific media item for that specific user.

Implementation

Two versions of this recommender have been implemented:

- News recommender: takes as input a news item descriptor and a user profile and gives as output a value among 0 and 1 representing the satisfaction index of the specific news item for the specific user. The provided news descriptor contains some URIs of DBpedia resources being involved in the news item.
- Datawarehouse recommender: takes as input a list of parameter (about EPG start date, EPG end date, channel, ..) and a user profile and gives back a list of recommended programmes in the requested temporal range and provided by the specified channel. Internally it leverages a Notube Datawarehouse service providing EPG information.
3.3.2 Property

There is a second recommender that is in a development phase. It embraces a more agnostic approach, since it doesn’t have a specific graph type to follow, as it happens with the skos:subject arcs of the previous recommender class. It simply tries to follow every inbound and outbound arc starting from each user activity object and from each media item to be recommended. In slightly more technical terms, the recommender browse all the Object Properties that start from both sides. The aim of the recommender is to find one or more paths from an activity object to a media item or viceversa. A lot of paths means a lot of semantic connections: the reached media item will be recommended to the user because it has to do with some user activity objects in many ways. Therefore this recommender doesn’t need any user profile in order to work, but just user activities.

A refinement process is obviously needed, in order to prevent the recommender from following useless or trivial property graphs, such as owl:Thing or dbpedia-owl:person,
stating that something is, respectively, a thing or a person. The strength of this recommender resides in the potentially infinite new ways in which user activity objects and media item resources are linked by semantic properties, connections that can be leveraged without knowing exactly what they represent or stand for. A further refinement is to narrow the browsing to specific datasets in the Linked Data Cloud. In the following picture the possible chosen datasets are depicted with a red circle around.

![Linked Data Cloud Diagram](image)

Figure 3.4: Linked Data Cloud

Each dataset has a specific reason for being chosen and a specific interesting domain for resources involved in both user activities and recommended media items. In particular:

- **DBpedia** ([http://dbpedia.org](http://dbpedia.org)): as we have seen, it is the main generic domain ontology. It is the semantic transposition of Wikipedia ([http://www.wikipedia.org](http://www.wikipedia.org)) and thus it is automatically generated from users contribution.

- **LinkedMDB** ([http://data.linkedmdb.org](http://data.linkedmdb.org)): it is the semantic trasposition of IMDB ([http://www.imdb.com/](http://www.imdb.com/)), the most famous movie repository. It can be queried to find properties starting from user objects and concerning watch activities or also generic ‘likes’ activities about persons, places, .., which can be discovered as actors, directors, locations, and so on.

- **MusicBrainz** ([http://musicbrainz.org](http://musicbrainz.org)): dataset about music bands and artists. Queriable to find user properties starting concerning listen activities.

- **Freebase** ([http://www.freebase.com](http://www.freebase.com)): it is a generic domain ontology. The main difference from DBpedia lies in the fact that Freebase has a proprietary dataset, with very structured user contributions ([http://wiki.freebase.com/wiki/Creating_a_new_topic](http://wiki.freebase.com/wiki/Creating_a_new_topic)).
• **BBC Ontology** ([http://www.bbc.co.uk/ontologies/programmes](http://www.bbc.co.uk/ontologies/programmes)): it is an ontology about BBC programmes, series, brands, and so on, developed and maintained by BBC itself. It is useful to describe and enrich programmes coming from BBC channels.

• **Geonames** ([http://www.geonames.org](http://www.geonames.org)): ontology about places. Leveraged for user activities involving places and, on the media item side, location of programmes, movies, living places of actors.

### 3.3.3 Pattern

Pairing together somehow the idea behind the SKOS recommender and that one behind the Property recommender, the Pattern recommender aims at the same specificity of the SKOS recommender of picking up just single kinds of semantic arcs and at the idea of the Property recommender of building property paths among resources. This recommender looks for patterns: sequences of properties with a semantic well-defined meaning. I.e. one of these patterns could be in very abstract terms:

\[
\text{person} \ (\text{director}) \rightarrow \text{movie} \ (\text{located}) \rightarrow \text{place}
\]

The Pattern recommender strategy is currently being developed. With the idea that certain patterns formalize interesting knowledge while others state the obvious, the Pattern recommender aims to find interesting links between user interests and audiovisual items. See the figure below for a simple pattern above the striped horizontal line and an instance of that pattern below the striped horizontal line. Patterns involving people and organizations are often considered interesting. In the example of the said figure, ‘Facebook’ could be the interest of our user, and the unknown CEO of Facebook ‘Chris Hughes’ is in a talk show on the BBC. This shows that our recommender will be able to generate ‘long tail’ recommendations, i.e. recommend items based on facts and not on popularity.

### 3.4 Item to item

This class of recommenders gives as output the same recommendations for any user. Actually an item2item recommender does its best to find similar items of an item given as input. Similar here can be on the basis of different aspects of the item:

- genre
- actor, director
- place
- date
- ...

Any user profile is required by this kind of recommenders. It’s therefore the simplest and maximally privacy preserving option, and often the only option for services that do not require a user login.

Two very useful scenarios would be:
improve non-personalised recommendations

show an improvement to recommendations for the user when logged in

From the idea of programme-to-programme (non-personalised) recommendations, it has been examined how to make interesting recommendations to users. Using a very specific dataset in first experiments, and now beginning to see how to generalise this further: in particular for any graph where the nodes are items to be recommended the aim is to be able to give some measure of the interestingness of the recommendations, and specifically, whether the interestingness rises or falls with addition of more data, such as preferences or activity streams.

Consider a simple graph of programmes linked by various properties, such as category, contributor, theme, location:

Since user’s interests or activities are unknown, this graph is the only information available to go on. It can be simplified by turning the non-programme nodes into properties:
This resulting graph can be leveraged at least in the following three ways.

### 3.4.1 Simple linking

Picking one or more particular links in the graph from one programme to another (e.g. 'contributor'), recommendations can be done of programmes that share the same value for those links as the starting programme. A common version of this is just to pick 'genre', which normally gives rather uninteresting recommendations because there will only be a short list of genres to choose from.

### 3.4.2 Pairwise connectedness

There can be programmes that are highly related to other programmes, by having multiple relationships between them (for example the same director, same location, same genre). This could be a bit more interesting in that the relationship between the two programmes will be more specific, and since the user is presumably interested enough in the starting programme to have arrived at it, they will also be interested in recommendations that are close matches. However, some of those related properties will be irrelevant to any particular user, so using this technique implies the risk of excluding options that may be of interest.
3.4.3 Fanning

Programmes that are related to other programmes by many different properties can also be recommended. This is a bit subtle, but the idea is that this 'fanning out' gives a diversity of suggestions based on a single programme.

Our hypothesis that these last two characteristics of a graph of this kind are indicative of interestingness to the user: that the interest is higher when more than one category links two programmes, or when a programme is linked to others by more than one category. These two features can be termed as respectively 'pairwise connectedness' and 'fanning'. A possible suggestion is that browsing interest is higher if pairwise connectedness and fanning are both high. When they are both equal to 1, this comes down to simple linking.
Since these are simple graphs, an investigation into ‘interestingness’ might be extended into RDF graphs in general, and in particular, once having a measure of it, it could be useful to tell if the addition of another graph, suitably flattened - for example a set of preferences or an activity stream - is likely to increase interest or not. Graph theory has to be further investigated in order to find any other features of graphs that can help characterise their interestingness, and to test evaluate our hypothesis about pairwise connectedness and fanning.
4. Beancounter new features

With the following brief chapter we intend to describe some new features which have been added to the Beancounter.

4.1 New features

External services are now available that read from/write in the Beancounter triple storage.

4.1.1 Activity/Interest Ingest Service

An ingestor has been implemented for user activities and user interests. Instead of passing through the usual flow:

- user performs an activity on a social network
- the activity is gathered and semantically lifted by the Beancounter proper Tubelet
- a proper Beancounter Reasonlet computes and stores an interest out of the activity

an activity or an interest are sent directly to the Beancounter triple store, explicitly expressed according to agreed formats and structures.

4.1.2 UI-oriented user services

Some REST services have been developed where user data stored into the RDF graphs are read and aggregated according to different dimensions, such as:

- date of activity
- source (social network)

Moreover, recommendation reasons have been added, in order to give the justification for any recommended media items the user gets. The reason plays an important role in the chance the recommendation has to be followed.
5. Evaluation

Both user profiling and recommenders need a strong evaluation process. For this workpackage, evaluation is a matter of great delicacy, because there are no objective criteria to assess the good quality of a profiler or a recommender. It is only a very subjective and relative assessment that can be done. Still, some discrimination about kinds of evaluation can be successfully drawn.

5.1 Evaluation dimensions

One first big separation line can be drawn between technical evaluation of profiling and recommending services and end user perceived satisfaction. Therefore this is the first dimension of our evaluation space. The second one relates to what we want to evaluate. Three items can be identified:

- user activities
- user interests
- media recommendations

So eventually this will be our evaluation space:

![Evaluation space](image)

Figure 5.1: Evaluation space

5.2 User activities evaluation

User activities are the means Beancounter to compute the user profile. So the choice of important activities among the whole set of things a user can do throughout the web is relevant. Social networks are growing more and more numerous and different.
The subset of social networks the Beancounter calls in order to get user data has to cover all relevant aspects of a user’s social life. Moreover, even if the services subset is well chosen, the gathered activities must be correctly transposed into the internal model and rendered in RDF. Evaluation about user activities gathering concerns these two points, more something about user privacy preservation.

5.2.1 Technical evaluation

From a technical perspective, an evaluation step comprises trying to answer to the following question:

- How much do the gathered activities shaped according to our ontology (http://xmlns.notu.be/aair) match the original activities performed on the different social networks?

5.2.2 End user evaluation

From an end user perspective, an evaluation process can be set up, focussed on the following points:

- How much does the user agree on how activities in the beanounter match her/his activities on social networks?

- How much does the user agree on which service is called for each social network in order to get useful information about user activities on that specific social network? (i.e.: call Lastfm recent listened tracks service is better than call Lastfm user playlists?)

- Does the user find anything against the privacy policies she/he had specified about activities?

5.3 User profile evaluation

Given a good representation of activities, the next evaluation step involves the generation of the user profile itself. At this step, a technical evaluation is very difficult, because there’s no right profile or interest representation. The only way would be to compare our profiler with other profilers, having given them the same input.

5.3.1 Technical evaluation

Similarly to what we said for user activities evaluation, we ask ourselves the following question:

- How much do the computed interests in the user profile successfully cover the activity objects contained in the user activity graph?
5.3.2 End user evaluation

The main core of user profiling evaluation involves the end users, the only ones who can really judge on profile they have been given:

- How much does a user recognize her/himself in the profile they’ve been given?

- How much does a user agree with the weights of the interests in the profile, having given different weights to the sources she/he had granted permission for. I.e.: LastFM is a good source for my profile, but Facebook is not.

- How much does the profile follow feedbacks from the user: - implicit: based on what the user watches - explicit: based on what the user does directly on the profile (i.e. deleting interests)

- Does the user find anything against the privacy policies she/he had specified about interests?

5.4 Recommender(s) evaluation

The same consideration of user profiling can be applied here.

5.4.1 Technical evaluation

- How much does notube recommender achieve in comparison with traditional recommendation algorithms (i.e.: [http://blog.smellthedata.com/2009/06/netflix-prize-tribute-recommendation.html a netflix one])?

5.4.2 End user

- How much does the user enjoy notube recommendations?

- How much does notube recommender follow user view activities in general?

- How much does notube recommender follow user view/ignore activities on something being recommended?

- Does the user find anything against the privacy policies she/he had specified about recommendations?