User Modeling and Personalization 4: User Modeling Frameworks and Interoperability

Eelco Herder

L3S Research Center / Leibniz University of Hanover
Hannover, Germany

25 April 2016

Outline I

Outline II

User Modeling Systems

Generic User Modeling Systems

A user modeling system is defined as a 'generic user modeling system' (GUMS) if it is independent from the architecture and from the user model of a specific user-adaptive application.

Generic user modeling systems serve as a separate user modeling component in an application system at runtime.

They may be part of a local area network or a wide area network and serve more than one application instance at a time.

Kobsa, A. (2001). Generic user modeling systems. User modeling and user-adapted interaction, 11(1-2), 49-63.

Advantages of GUMS

- All information about the user is maintained in a repository with clearly defined points of access
- User information is at the disposal of more than one application at a time
- User information acquired by one application can be employed by other applications, and vice versa
- Information about users is stored in a non-redundant manner

(continued on next slide)

- It is easier to maintain consistency and coherence of information gathered by different applications
- Information about user groups (e.g. stereotypes) can be maintained with low redundancy
- Methods and tools for system security, identification, authentication, access control and encryption can be applied and maintained
- Privacy, transparency

Amazon: The Need to Scale (1/2)

Size

A large retailer might have tens of millions of customers and millions of distinct catalog items.

Speed

The results need to be returned in less than half a second, while still producing high-quality recommendations.

From: Greg Linden, Brent Smith, and Jeremy York. Amazon.com Recommendations - Item-to-Item Collaborative Filtering. Industry Report 76 Jan-Feb 2003. IEEE Computer Society.

Amazon: The Need to Scale (2/2)

New Users

New customers typically have extremely limited information, based on only a few purchases or product ratings.

Changing Interests

Customer data is volatile: Each interaction provides valuable customer data, and the algorithm must respond immediately to new information.

From: Greg Linden, Brent Smith, and Jeremy York. Amazon.com Recommendations - Item-to-Item Collaborative Filtering. Industry Report 76 Jan-Feb 2003. IEEE Computer Society.

Requirements/Characteristics of GUMS

Several characteristics of generic user modeling systems have been regarded as very important over the years:

- **Generality:** the system should be usable in as many domains as possible
- **Expressiveness:** different kinds of user characteristics should be stored in various suitable formats
- **Inferential capabilities:** GUMS are expected to support complex assumptions and complex reasoning, to be useful in a wide range of domains
- **Quick adaptation:** Provide reasonable (or default) assumptions even if there is only a limited amount of user data available

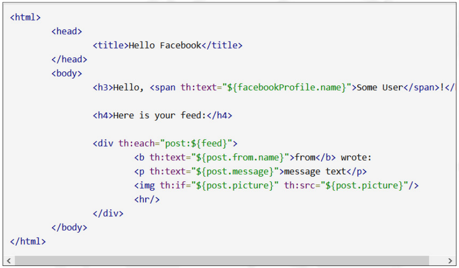
- **Extensibility:** Interfaces that allow for the (possibly bi-directional) exchange of user information
- **Import functionality:** Existing data (for example LDAP) should be accessible to the user modeling server
- **Management of distributed information:** Integration of several sources of user information, such as user profiles, purchase records, . . .
- **Support for open standards:** such as LDAP, FOAF, OpenID

- **Load balancing:** response delays or denials of requests should only occur in emergency situations
- **Failover strategies:** fallback mechanisms in case of a breakdown
- **Transactional consistency:** avoid inconsistencies due to parallel read/write on the user model
- **Privacy support:** adhere to privacy policies, provide privacy-enhancing services

Facebook as a GUMS

The infrastructure must support more than 1 million web sites and 550,000 applications using the Facebook Connect platform.

http://www.datacenterknowledge.com/the-facebook-data-center-faq/



```
<html>
        <head>
                <title>Hello Facebook</title>
        </head>
        <body>
                <h3>Hello, <span th:text="${facebookProfile.name}">Some User</span>!</

                <h4>Here is your feed:</h4>

                <div th:each="post:${feed}">
                        <b th:text="${post.from.name}">from</b> wrote:
                        <p th:text="${post.message}">message text</p>
                        <img th:if="${post.picture}" th:src="${post.picture}"/>
                        <hr/>
                </div>
        </body>
</html>
```

Figure: Hello World for using Facebook user data.

https://spring.io/guides/gs/accessing-facebook/

Services of GUMS

The following services are frequently found in GUMS:

User Data Acquisition

- The recording of users' behavior, particularly interaction with the system

Knowledge Inference

- The formation of assumptions based on the interaction history
- The generalization of the interaction histories of many users into stereotypes
- The drawing of additional assumptions about the current user based on initial ones

Kobsa, A. (2001). Generic user modeling systems. User modeling and user-adapted interaction, 11(1-2), 49-63.

User Model Representation

- The representation of assumptions about one or more types of user characteristics
- The representation of relevant common characteristics in specific user subgroups (i.e. stereotypes)
- The classification of users to a subgroup or stereotype

User Model Management

- Consistency maintenance in the user model
- The provision of current assumptions about the user, as well as justifications for these assumptions
- The evaluation of the entries in the current user model and the comparison with given standards

Amazon: How To Scale User Modeling and Recommendation

Item-to-Item Recommendation

Because existing recommendation algorithms cannot scale to Amazon.coms tens of millions of customers and products, we developed our own.
Our algorithm, item-to-item collaborative filtering, scales to massive data sets and produces high-quality recommendations in real time.

Offline Computation

For very large data sets, a scalable recommendation algorithm must perform the most expensive calculations offline.

Decentralized Approaches

Centralized GUMS have several disadvantages:

- GUMS may impose a restrictive centralized model, or insufficient interfaces for the exchange of user data
- Reliability and load balancing may be improved by introducing mirrors, but this leads to problems with synchronization and coordination
- Who is responsible for the protection of the user data (the client adaptive systems or the GUMS administrators)?

Decentralized User Modeling Systems

In decentralized user modeling approaches, each adaptive system maintains its own user model, as needed for its own purposes. The systems communicate directly in a peer-to-peer manner to exchange user profiles and/or user data.

Decentralized user modeling solutions typically offer

- functionality for the mapping and integration of different types of models
- solutions for the discovery of communication partners and services

Mixed Approaches

Mixed centralized and decentralized approaches exist as well:

- user models are stored locally (decentral), but adhering to a common format
- user models are stored in a decentralized way, but mediation is done through a centralized point of access

Centralized and decentralized approaches to GUMS

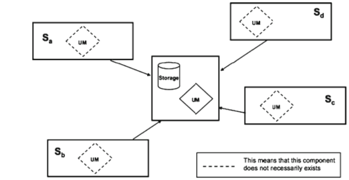


Fig. 1 Centralized approach to UM interoperability

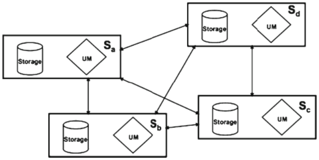


Fig. 2 Decentralized approach to UM interoperability

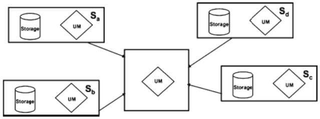


Fig. 3 Mixed approach to UM interoperability

Figure: Centralized, decentralized and mixed UM approaches

Facebook: Where is User Data Stored

From Small to Large

With more than 900 million active users, Facebook is the busiest site on the Internet. The social networking site was launched in February 2004, initially out of Mark Zuckerbergs dorm room at Harvard University and using a single server.

Decentralization

Loading a users home page typically requires accessing hundreds of servers, processing tens of thousands of individual pieces of data, and delivering the information selected in less than one second,

http://www.datacenterknowledge.com/the-facebook-data-center-faq/

Facebook: Need for Dedicated Software?

Open Source

The site is written primarily in the PHP programming language and uses a MySQL database infrastructure. To accelerate the site, the Facebook Engineering team developed a program called HipHop to transform PHP source code into C++ and gain performance benefits.

MySQL

Facebook has one of the largest MySQL database clusters anywhere, and is the worlds largest users of memcached, an open source caching system.

http://www.datacenterknowledge.com/the-facebook-data-center-faq/

Enriching user profiles with the Semantic Web

Tim Berners-Lee

The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.

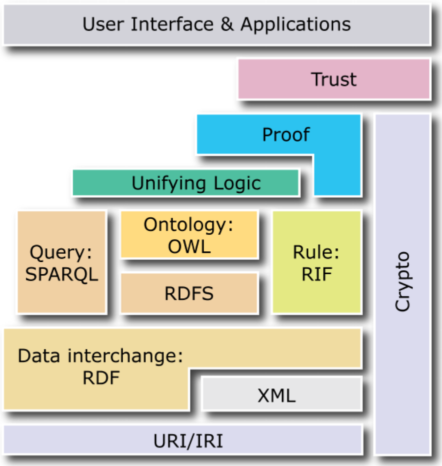
The Semantic Web

The Semantic Web provides a common framework that allows data to be shared and reused accross application, enerprise and community boundaries.

It is a collaborative effort led by W3C with participation from a large number of researchers and industrial partners.

It is based on the Resource Description Framework (RDF), which integrates a variety of applications using XML for syntax and URIs for naming.

Semantic Web Layers



RDF

RDF is a model for representing information about resources in the WWW.
In particular: metadata about Web resources (e.g.,title, author, version, publication date, ...)

RDF can also be used as a model for representing information about things that can be identified in the WWW.
In particular: metadata about items (e.g., prices, specifications, availability information, etc.)

In RDF, things are identified using Web Identifiers (Uniform Resource Identifiers, URIs)

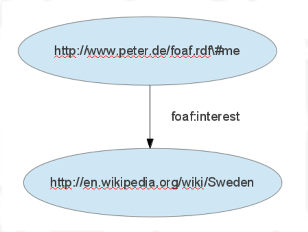
Things are described using properties and property values

The core building blocks of RDF are *triples*: subject - predicate - object.

‘Peter is interested in Sweden’

subject: http://www.peter.de/foaf.rdf#me;
predicate: foaf:interest;
object: http://en.wikipedia.org/wiki/Sweden;

This statement can be represented as a graph.


http://www.peter.de/foaf.rdf\#me
foaf:interest
http://en.wikipedia.org/wiki/Sweden

RDF Schema and ontologies

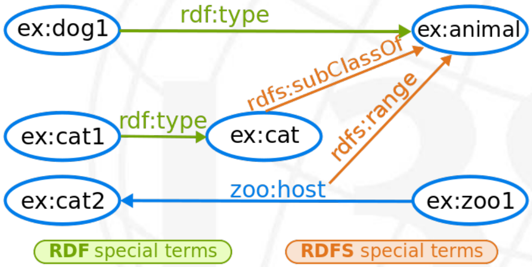
RDF Schema (RDFS) is a set of classes with certain properties using the RDF extensible knowledge representation language, providing basic elements for the description of ontologies.

An **ontology** formally represents knowledge as a set of concepts within a domain, and the relationships between pairs of concepts. It can be used to model a domain and support reasoning about entities.

For example, if we want to represent the following statements:

- Dog1 is an animal
- Cat1 is a cat
- Cats are animals
- Zoos host animals
- Zoo1 hosts the Cat2

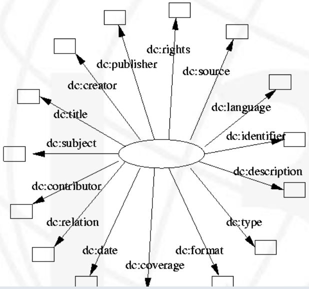
This would result in the following RDF graph:



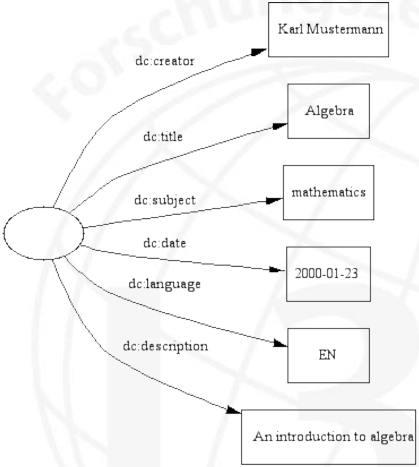
Dublin Core

The Dublin Core metadata terms are a set of vocabulary terms which can be used to describe resources for the purposes of discovery.

The terms can be used to describe a full range of web resources (video, images, web pages, etc.), physical resources such as books and objects like artworks.



An example Dublin Core description of a book

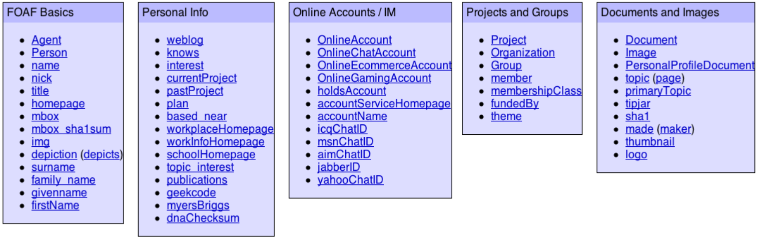


The same description in XML

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns\#"
            xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description>
    <dc:creator>Karl Mustermann</dc:creator>
    <dc:title>Algebra</dc:title>
    <dc:subject>mathematics</dc:subject>
    <dc:date>2000-01-23</dc:date>
    <dc:language>EN</dc:language>
    <dc:description>An introduction to algebra
  </dc:description>
  </rdf:Description>
</rdf:RDF>
```

Friend Of A Friend (FOAF)

FOAF is a language to describe people and resources in the Web.



FOAF Basics

- Agent
- Person
- name
- nick
- title
- homepage
- mbox
- mbox_sha1sum
- img
- depiction (depicts)
- surname
- family_name
- givenname
- firstName

Personal Info

- weblog
- knows
- interest
- currentProject
- pastProject
- plan
- based_near
- workplaceHomepage
- workInfoHomepage
- schoolHomepage
- topic_interest
- publications
- geekcode
- myersBriggs
- dnaChecksum

Online Accounts / IM

- OnlineAccount
- OnlineChatAccount
- OnlineEcommerceAccount
- OnlineGamingAccount
- holdsAccount
- accountServiceHomepage
- accountName
- icqChatID
- msnChatID
- aimChatID
- jabberID
- yahooChatID

Projects and Groups

- Project
- Organization
- Group
- member
- membershipClass
- fundedBy
- theme

Documents and Images

- Document
- Image
- PersonalProfileDocument
- topic (page)
- primaryTopic
- tipjar
- sha1
- made (maker)
- thumbnail
- logo

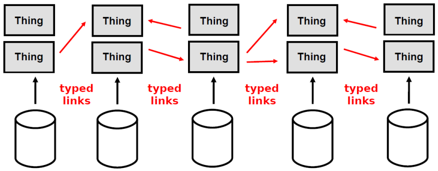
Linked data

The Web enables us to link related documents. Similarly it enables us to link related data.

Linked Data refers to a set of best practices for publishing and connecting structured data on the Web.

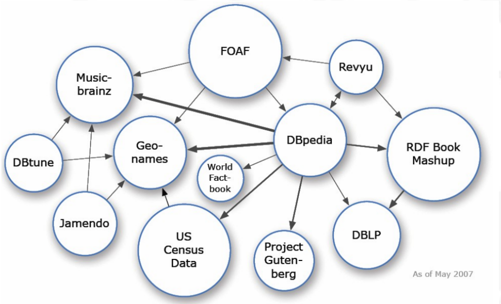
Key technologies that support Linked Data are

- **URIs** - a generic means to identify entities or concepts in the world
- **HTTP** - a simple yet universal mechanism for retrieving resources, or descriptions of resources
- **RDF** - a generic graph-based data model with which to structure and link data that describes things in the world



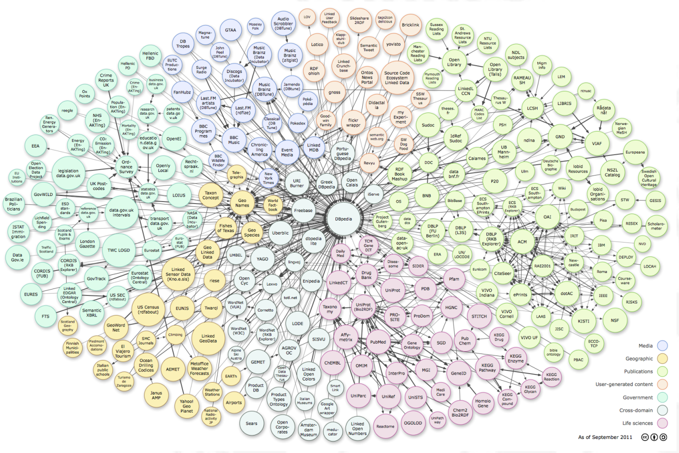
The LOD cloud

The Linking Open Data (LOD) project takes existing open data sets, makes them available on the Web in RDF and interlink them with other data sets.



2007

The LOD cloud expands quickly

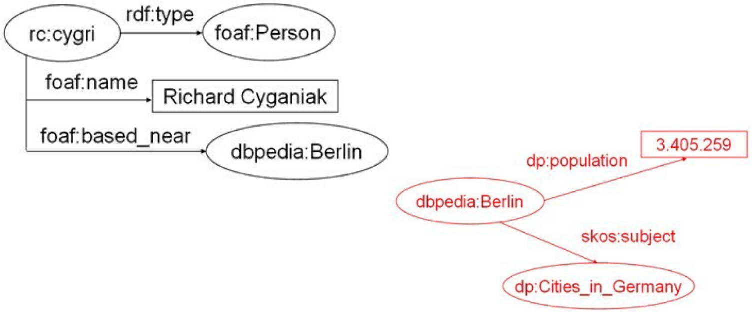


2011

How to make use of Linked Data

Making use of linked data principles and the “LOD Cloud”, several pieces of knowledge can be connected.

For example, making use of DBPedia, we can infer that Richard Cyganiak lives in a city with a population of 3.405.259.



User Model Interoperability

Interoperability

The ability to cooperate and exchange data despite differences in languages, interface and execution platform

The main advantages of interoperable user models are:

- acquiring more data and more accurate data about users
- acquiring functionalities (both user model and user modeling functionalities) that systems do not themselves implement

Interoperability for better user models

Cross-application user model interoperability is a strategy to cope with the *cold start problem*, which refers to the difficulty for applications to provide suitable adaptations for new users.

Interoperability may also relieve users from the pain of training new systems or wasting time filling in their profile for every new application.

In general, it is believed that interoperable user models lead to more user data and more accurate models.

Dimensions of interoperability

In order to successfully exchange and (re-)use user profile data, requirements on several levels need to be met:

- *Structural level*: Protocols for communication (i.e. interfaces)
- *Syntactic level*: Languages for the exchange of user model data
- *Semantic level*: Agreement on definitions what certain concepts (such as knowledge), terms and expressions, exactly mean

Interoperability on a structural level

Protocols for communication are abundant:

- Remote calls (remote procedure calls, Java remote service invocation, other client-server approaches)
- Web-based protocols (HTTP POST/GET, WSDL/SOAP, XML, JSON, OAI, RESTFUL interfaces)

Syntactic and semantic interoperability is much harder, as we will see in the next part of this lecture.

Two basic approaches

In essence, there are two ways to ensure interoperability between two (or more) adaptive systems and their user models:

- **Lingua franca:** an agreement between all parties on a common representation and semantics
  This is the philosophy underlying the generic (centralized) user model server approach
- **Conversion** of user model information between the different systems.

Lingua Franca Approaches

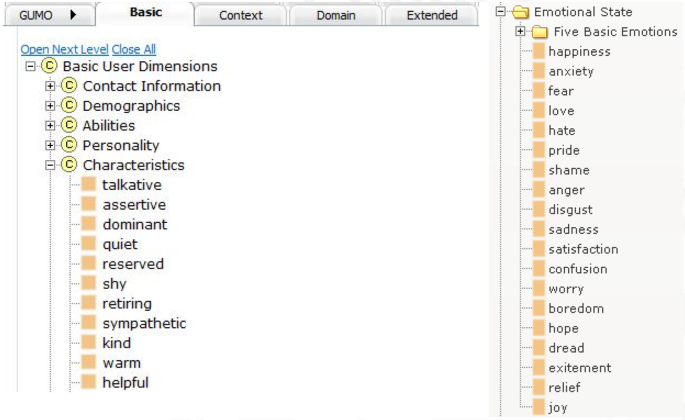
Generic User Modeling Ontology

GUMO is developed at the DFKI German Institute for Artificial Intelligence and is divided in four parts:

- *GUMO-Basic*: models user dimensions like personality traits and user characteristics
- *GUMO-Context*: models the world directly around the user, including the events and the environmental context
- *GUMO-Domain*: includes a general interest (domain overlay) model
- *GUMO-Extended*: ranges, rating dimensions and predicates that address other attributes

By adding new parts to GUMO, other ontologies (for example elearning-related) can be integrated.

GUMO



GUMO ▸ | **Basic** | Context | Domain | Extended

Open Next Level Close All

- Basic User Dimensions
  - Contact Information
  - Demographics
  - Abilities
  - Personality
  - Characteristics
    - talkative
    - assertive
    - dominant
    - quiet
    - reserved
    - shy
    - retiring
    - sympathetic
    - kind
    - warm
    - helpful

- Emotional State
  - Five Basic Emotions
  - happiness
  - anxiety
  - fear
  - love
  - hate
  - pride
  - shame
  - anger
  - disgust
  - sadness
  - satisfaction
  - confusion
  - worry
  - boredom
  - hope
  - dread
  - exitement
  - relief
  - joy

Figure: Screenshot of GUMOs user interface

Conversion Approaches

Conversion allows for flexible and extensible user models, at the price of possible loss of information in the conversion process:

- models may be simply incompatible (there is no suitable mapping)
- mappings may be incomplete (information required in one model is not available in the other)
- the observations in the different systems may lead to contradictions

Conversion: A rule-based example (1/2)

We illustrate conversion approaches by means of a Grapple Derivation Rules that convert data from one format into another.

This simple rule maps test scores of the elearning system CLIX to a more generic external knowledge format by simply multiplying the score by 10:

Conversion: A rule-based example (2/2)

```
<gdr:rule
  xmlns:gdr="http://www.grapple-project.org/grapple-derivation-rule/"
  xmlns:gc="http://www.grapple-project.org/grapple-core/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns\#"

  id="1"
  name="Quiz Level to External Knowledge (CLIX)"
  description="Maps the quiz/test result of a specific CLIX score to external knowledge"
  creator="http://pcwin530.win.tue.nl:8080/grapple-umf/client/1">
   <gdr:premise dataspace="1">
      <gc:subject>?user</gc:subject>
      <gc:predicate rdf:resource="http://www.grapple-project.org/ims-lip/completedTest"/>
      <gc:object>solar-system-quiz-id</gc:object>
      <gc:level>?level</gc:level>
   </gdr:premise>
   <gdr:consequent dataspace="1">
      <gc:subject>?user</gc:subject>
      <gc:predicate rdf:resource="http://gale.tue.nl/predicate/knowledge"/>
      <gc:object>gale://gale.tue.nl/cam/DavidTestCAM/SolarSystem</gc:object>
      <gc:level>op:multiply(?level,10)</gc:level>
   </gdr:consequent>

</gdr:rule>
```

Aggregation of User Models

Users leave different types of profile traces on the Social Web.

People fill out their profile attributes, such as name, affiliations, etcetera.

Social tagging systems capture tagging activities of the users to create *tag-based profiles*.

Form-based profiles

The form-based profile of a user u is a set of attribute-value pairs.

$$\textit{UM(u)} = \{\{a, v\} | a \in A_{UM} \textit{ and } v \textit{ is in the range of a}\}$$

A_{UM} defines the vocabulary of attributes that can be applied to describe characteristics of the user u. The value v associated with an attribute a must be in the range of a.

Traditional attributes might be name or email address:

- UM(u_1) = {(name, 'Bob'), (email, `bob@mail.com`)}

Tag-based profiles

The *tag-based profile* of a user u is a set of weighted tags where the weight of a tag t is computed by a certain strategy w with respect to the given user u.

$$P(u) = \{\{t, w(u,t)\} | t \in T_{source}, u \in U\}$$

$w(u,t)$ is the weight that is associated with tag t for a given user u. T_{source} is the source set of tags from which tags are extracted for the tag-based profile $P(u)$.

For example, P(u_1) $=$ {(research, 0.65), (semantic web, 0.2), (jazz, 0.15)}

Aggregation of form-based profiles

As they serve different purposes, all Social Web Platforms collect their own sets of user data.

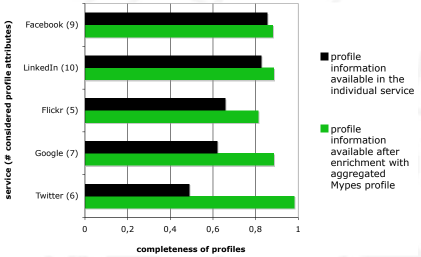
form-based profile attributes	Face-book	LinkedIn	Twitter	Blog-Spot	Flickr	Delicious	Stumble Upon	Last.fm	Google
nickname	x	x	x	x	x	x	x	x	x
first name	x	x							
last name	x	x							
full name	x	x	x		x				x
profile photo	x		x		x				x
about		x							x
email (hash)	x				x				
homepage	x	x	x						x
blog/feed			x	x	x	x	x	x	
location		x	x		x				x
locale	x								
interests		x							
education		x							
affiliations	x	x							
industry		x							

Fabian Abel, Eelco Herder, Geert-Jan Houben, Nicola Henze, Daniel Krause. Cross-system User Modeling and Personalization on the Social Web. UMUAI (2-3), 2013, pp 169-209

Completeness of user profiles

Users often do not fill out their profiles completely. For example, Twitter only asks 6 attributes, but these profiles are only completed up to 49%.

Aggregating data from different sources leads to more complete user profiles.



Fabian Abel, Eelco Herder, Geert-Jan Houben, Nicola Henze, Daniel Krause. Cross-system User Modeling and Personalization on the Social Web. UMUAI (2-3), 2013, pp 169-209

Aggregation of tag-based profiles

How useful is a tag-based profile? Intuitively, if the profile contains just one tag, this profile is not very useful.

The profile becomes more varied if other tags are included - in particular if there is not one tag that appear 8 out of 10 times in the profile.

One way to measure the usefulness of a tag-based profile is the *entropy* of the collection.

Entropy

Entropy is a measure of disorder, or more precisely unpredictability.

For example, a series of coin tosses with a fair coin has maximum entropy, since there is no way to predict what will come next.

A string of coin tosses with a two-headed coin has zero entropy, since the coin will always come up heads.

Most collections of data in the real world lie somewhere in between.

Calculating Entropy

The entropy of a tag-based profile T, which contains of a set of tags t, is computed as follows:

$$entropy(T) = \sum_{t \in T} p(t) \cdot \textit{self-information}(t) \tag{1}$$

In Equation 1, $p(t)$ denotes the probability that the tag t was used by the corresponding user. Self-information is the logarithm of $p(t)$ multiplied by -1:

$$\textit{self-information}(t) = -log_2(p(t)) \tag{2}$$

Using base 2 for the computation of the logarithm allows for measuring self-information as well as entropy in bits.

User data used for tag-based entropy

For modeling the probability $p(t)$ that a tag t appears in a given user profile, we apply the individual usage frequencies of the tags: for a specific user u the usage frequency of tag t is the fraction of u's tag assignments where u referred to t.

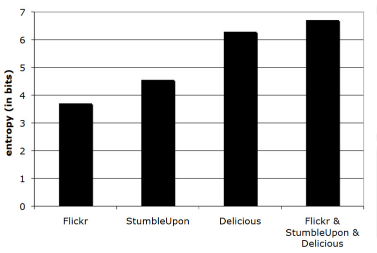


Figure: Tag entropy in Delicious, StumbleUpon and Flickr. In the photo sharing service Flickr, entropy is lowest.

Overlap between user profiles

Aggregation of tag-based profiles is meant to reveal more information about the users than the profiles available in some specific service.

If you combine two more or less identical profiles, the added value is rather low. If you combine two completely different profiles (e.g. your favorite music with your research interests), the entropy will increase, but the aggregated profile is probably less useful.

Therefore, there needs to be at least *some* overlap between the profiles that you combine.

Calculating overlap between profiles

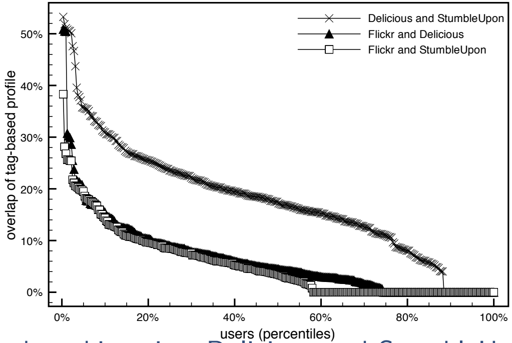
For each user u and each pair of services A and B, we compute the overlap as specified in Definition 3.

$$overlap(u_A, u_B) = \frac{1}{2} \cdot \left(\frac{|T_{u,A} \cap T_{u,B}|}{|T_{u,A}|} + \frac{|T_{u,A} \cap T_{u,B}|}{|T_{u,B}|}\right) \qquad (3)$$

$T_{u,A}$ and $T_{u,B}$ denote the set of (distinct!) tags that occur in the tag-based profile of user u in service A and B respectively. Hence, $|T_{u,A} \cap T_{u,B}|$ is the number of (distinct) tags that occur in both profiles, u_A and u_B.

Calculating overlap in this way compensates for differences in profile size of services A and B.

Example profile overlaps

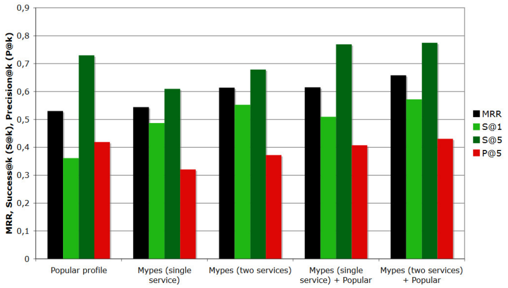


The social bookmarking sites Delicious and StumbleUpon have the biggest overlap, but still rather small (for more than 50% of the users less than 20%). The photo sharing site Flickr is different from the other two.

Benefits of aggregated profiles for tag recommendation

In an experiment, we showed that using the user's own tags from other services performs significantly better than using the most popular tags of the target system.

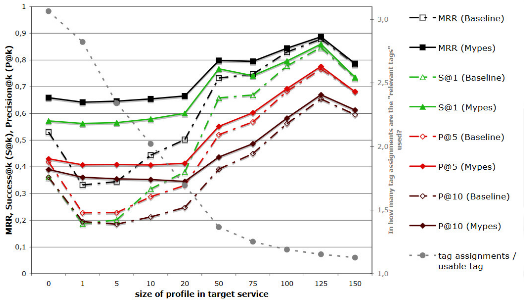
A mixture of the user's own tags and the most popular tags performs even better.



Limitations of aggregated profiles

Using the user's own tags from other systems is particularly useful for solving the *cold start* problem.

When the target profile size exceeds 100 tags, the performance differences are no longer significant.


MRR, Success@k (S@k), Precision@k (P@k)
In how many tag assignments are the "relevant tags" used?
size of profile in target service
MRR (Baseline)
MRR (Mypes)
S@1 (Baseline)
S@1 (Mypes)
P@5 (Baseline)
P@5 (Mypes)
P@10 (Baseline)
P@10 (Mypes)
tag assignments / usable tag