Brigitte Endres-Niggemeyer *Editor*

# Semantic Mashups

## Intelligent Reuse of Web Resources

Springer

Semantic Mashups

Brigitte Endres-Niggemeyer
Editor

# Semantic Mashups

## Intelligent Reuse of Web Resources

Springer

*Editor*
Brigitte Endres-Niggemeyer
Hanover, Germany

ISBN 978-3-642-36402-0 ISBN 978-3-642-36403-7 (eBook)
DOI 10.1007/978-3-642-36403-7
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013936081

ACM Computing Classification (1998): H.3.5, I.2, H.5, D.2



Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

# Preface

At first sight, mashups are easily defined. They integrate existing web resources in order to produce helpful new resources or services. Users are in focus—they are thought to easily create mashups for a current use situation.

Mainly because of their simplicity, their reuse of existing resources, and their user orientation, mashups are so widely distributed on the web.

Mashups are semantic mashups—to different degrees. Some understanding of the incoming resources is a precondition for combining them reasonably. Selective perception is common, but evidence for non-semantic mashups is still missing. Until further notice we assume that the semantic features of mashups are ubiquitous. They are prime mashup properties, convincing by their practical value and more.

Mashups are wide-spread and their proper ordering is notoriously difficult. Koschmider et al.[1] promise to elucidate the mashup hype (sic) distinguishing mashups depending on

- what mashups display:
  - dimension 1: presentation mashups, data mashups, functionality mashups
  - dimension 2: mapping mashups, photo/video mashups, search/shopping mashups, news mashups
- where mashups are put together: server-side mashups, client-side mashups
- how mashups get input: extraction mashups, flow mashups
- mashup users: consumer mashups, business or enterprise mashups

Every real existing mashup is entitled to participate in a choice of these categories, and to add some others that Koschmider et al. do not mention. Mashups mixing client-side and server-side activity are as normal as mashups obtaining content by information extraction from text and picking up video clips. More examples are easy to imagine but not needed.

The chapters of this book render a part of real-life mashup diversity. Their basic organization is simple.

[1] http://mashup.pubs.dbs.uni-leipzig.de/files/paper14%5B1%5D.pdf.

- The two overview chapters of the beginning take readers around in the mashup environment:
  - First mashups are followed through their manifold habitats/ecosystems.
  - The second chapter concentrates on the regulations (standards, guidelines, APIs) that mashups must rely on for integrating web resources of independent producers.
- In the next sequence mashups are traced in their web home stations. The semantic mashups must be explained in more detail, whereas web environments like DBpedia or search engines are familiar to almost all readers. In contrast the Web of Things (WoT) is new and of big impact. Mashups in web contexts are considered in chapters on:
  - DBpedia mashups
  - mashups for web search engines
  - mashups for sensors and the web of things

  Chapters of this group may be particularly attractive for a developer audience.
- With the following properly application-oriented mashups, the lovers of multicolored and specialized mashup domains will get their money's worth. The authors explain mashups on:
  - mathematical knowledge
  - speech
  - emergency crisis management
  - similarity usage
  - traveling
  - in-town surroundings

Especially here, readers may be taken to fields where they run out of prior knowledge. To ease their life, the book ends with a substantial glossary and subject index.

A book covering a wide range of mashups must assemble a group of authors contributing chapters on their own research fields. The authors of this book met during the AI Mashup Challenge. It ran four times, first at the 2009 German AI Conference in Paderborn and the last three times during the Extended Semantic Web Conferences (ESWC) 2010–2012.

Fortunately the AI Mashup Challenge was well supported. Elsevier, Linguatec, O'Reilly, and Addison-Wesley sustainedly sponsored its runs. We gratefully acknowledge their contribution.

For their review of book chapters we thank Felix Burkhardt, Rui Cai, Emanuele Della Valle, Michael Hausenblas, Pascal Hitzler, Krzysztof Janowicz, Paul Librecht, Horacio Saggion, Jevon Wright, and Mao Ye. Special thanks go to Christoph Lange for his assistance in editor tasks.

Lucca, Italy Brigitte Endres-Niggemeyer

# Contents

# Chapter 1
# The Mashup Ecosystem

**Brigitte Endres-Niggemeyer**

**Abstract** The web is growing quickly, substructures are coming up: a {social, semantic, etc.} web, or the {business, services, etc.} ecosystem which includes all resources of a specific web habitat. In the mashup ecosystem, developers are in intense scientific activity, what is easily measured by the number of their recent papers. Since mashups inherit an opportunistic (participatory) attitude, a main point of research is enabling users to create situation-specific mashups with little effort. After an overview, the chapter highlights areas of intensive discussion one by one: mashup description and modeling, semantic mashups, media mashups, ubiquitous mashups and end-user related development. Information is organized in two levels: right under the headings, a block of topic-related references may pop up. It is addressed to readers with deeper interest. After that, the text for everybody explains and illustrates innovative approaches. The chapter ends with an almost fail-safe outlook: given the growth of the web, the ecosystem of mashups will keep branching out. Core mashup features such as reuse of resources, user orientation, and versatile coordination (loose coupling) of components will propagate.

## 1.1 The Mashup Ecosystem

**On the Ecosystem:** [10, 14, 18, 28, 71, 109, 126, 152, 155, 171, 187, 196, 208, 209, 224, 229, 230, 239, 240, 242]

Mashups are advancing on the internet, the web, and the semantic web. They have no problems to adapt to the cultures in the web [10], performing on the semantic web as on the internet or web in general. Their count is going up. They expand their services into new areas. They take root. Their simple principle of building upon work of others is gaining acceptance. As far as one can see mashups will remain on the move. In [208] Spivack illustrates how he anticipates the web and the semantic web will go on developing (see Fig. 1.1). Corresponding to the fast expansion of the web, people tend to define substructures: a social web, a web of services, a semantic

B. Endres-Niggemeyer (✉)
Heidegrün 36, 30179 Hanover, Germany
e-mail: brigitteen@googlemail.com

B. Endres-Niggemeyer (ed.), *Semantic Mashups*,
DOI 10.1007/978-3-642-36403-7_1, 

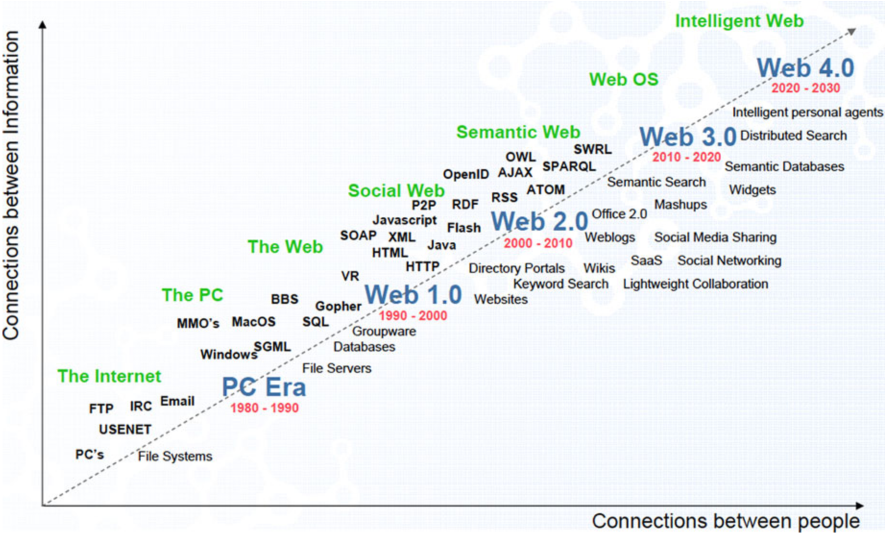



**Fig. 1.1** Timeline of internet and web development (from [208]). Notice mashups coming up towards 2010

web, a mobile web, a web of things, and so on. The subwebs overlap as shown in Fig. 1.2. Like smaller geographical or organizational units, let us say the regions of a country, these subwebs partition the web universe, so that local communities can concentrate on the concerns of their own subunit.

**Fig. 1.2** Subwebs of the web (from [90])

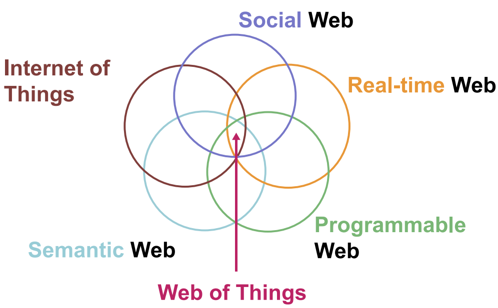
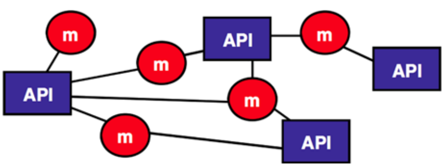



**Fig. 1.3** The mashup ecosystem linking mashups and APIs. Mashups in *red circles*, APIs in *blue squares* (from [240])



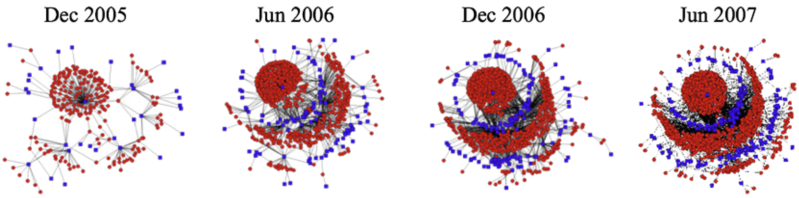



**Fig. 1.4** The mashup ecosystem evolution. Mashups in *red*, APIs in *blue* (source: [240]) (Color figure online)

Mashups are a relatively new web concept. Their history begins with DJ mashups of songs and with web 1.0 portals. The oldest mashup on Programmable Web[1] was added in 2005. The mashup ecosystem [240] may be seen as linking mashups and web APIs (see Fig. 1.3). [196] conceives it as configuration of service providers, mashup authors, and users without any central authority. The mashup ecosystem also appears as a specific software ecosystem (details in [28]). Thus the mashup ecosystem integrates mashups and their cohabitants wherever they may be spread on the web. Like a biological ecosystem, it interconnects all species that are needed for its functioning, such as users, tools or script languages. The affinity of mashups to composite web services [71, 171] is evident, so that methods from both sides cross the border without trouble.

The mashup ecosystem is growing quickly. Some evidence available for instant inspection is shown in Fig. 1.4. [229–231] describe a growth model in detail. Success factors for mashups are the activation of end-users as creators/designers, the attractivity of the most popular APIs (all readers will guess right: Google Maps, Twitter, YouTube and so on—more on the ProgrammableWeb hit list[2]), and the simple technique of copying—the reuse of existing resources. The mashup ecosystem shares the innovation rate of the web and its service ecosystem (also called internet/web of services—more detailed description in [14, 187]). [126, 155] explain the computational marketplace ecosystem. It serves mashups, too—why should mashups pick up their APIs anywhere on the web instead of going straight to the service market for shopping?

As the whole service ecosystem, the mashup ecosystem is assumed to follow a pattern of open innovation (Fig. 1.5), branching out, advancing into new domains, reaching more developers and users, and so on.

In the following we inspect the mashup ecosystem where the discussion is particularly active and innovative: mashup description and modeling, semantic mashups, media mashups, ubiquitous mashups, and end-user related development.

[1] http://www.programmableweb.com.

[2] http://www.programmableweb.com/apis/directory/1?sort=mashups.

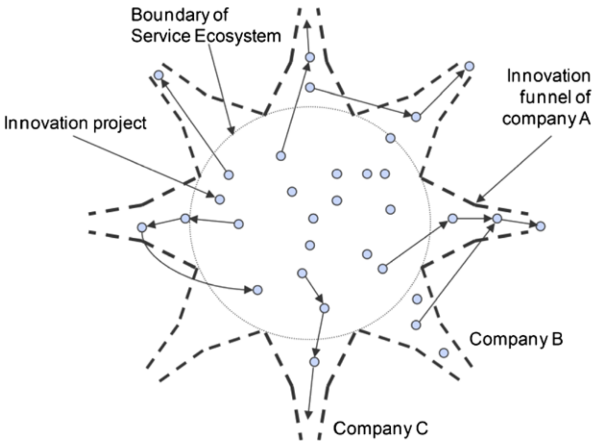



**Fig. 1.5** Innovation expanding from a service ecosystem (source: [187])

## 1.2 Mashup Description and Modeling

**On Model Descriptions:** [8, 47, 59, 61, 62, 66, 69, 78, 80, 96, 101, 105, 112–114, 117, 151, 164, 165, 177, 179, 190, 191, 221]

Mashups came later and as lightweight web applications into an environment where enterprise WSDL/SOAP web services with their more elaborated scheme were already established. In particular for enterprise mashups in intranets, the standards of earlier web services were and are kept up, while consumer mashups are being watched less for WSDL/SOAP compliance. Possibly mashups may, however, relax the climate for enterprise services. [79] states that

> enterprise mashups must realize the benefits already touted by end-user mashups.

This would summon earlier monoliths to adapt to the more flexible and abstract mashup concept.

Much effort is observed in modeling and description of mashups. Many mashup developers pursue the functional standards of the SOA-based web services habitat. In parallel, enterprise services and mashups begin to exploit web features such as semantic annotation, so that both parties are sharing more common ground.

A choice from the competing modeling and description activities on the market:

- Web Mashup Scripting Language (WMSL—[190])
- Enterprise Mashup Markup Language (EMML)[3] of the Open Mashup Alliance (OMA)
- Mashup Component Description Language (MCDL—[78])
- Universal model of components and composition [62]
- Universal model based on MetaObject Facility (MOF)[4] standards [180]

[3] http://www.openmashup.org/omadocs/v1.0/index.html.

[4] http://www.omg.org/mof/.

- UML2 model for a set of integrated mashups [80]
- ResEval Mash [113, 114] with a domain-specific description language (DSL)

The first and the last approach are chosen for closer inspection:

- The WMSL AM-AO use case because of its OWL ontology alignment of web services
- ResEval because of its two-level model with an abstract and a domain-related layer and the requirement-driven interface

From the mashup quality models [185], PEUDOM [39] is selected for a more detailed description.

Context awareness and personalization are main modeling issues as well. As they mostly happen in a ubiquitous environment, they will be dealt with there.

### *1.2.1 AM-AO: Web Mashup Scripting with OWL Ontology Use*

Imagine that AM (Air Mobility) and AO (Air Operations) cooperate. The AM system is responsible for missions like mid-air refueling and the movement of vehicles while the AO system is primarily concerned with offensive and defensive missions [192]. Each party has an ontology of its own [78, 190, 191].

A Web Mashup Scripting Language (WMSL) script permits end-users to combine AM and AO services. WMSL uses both its own script language and standard

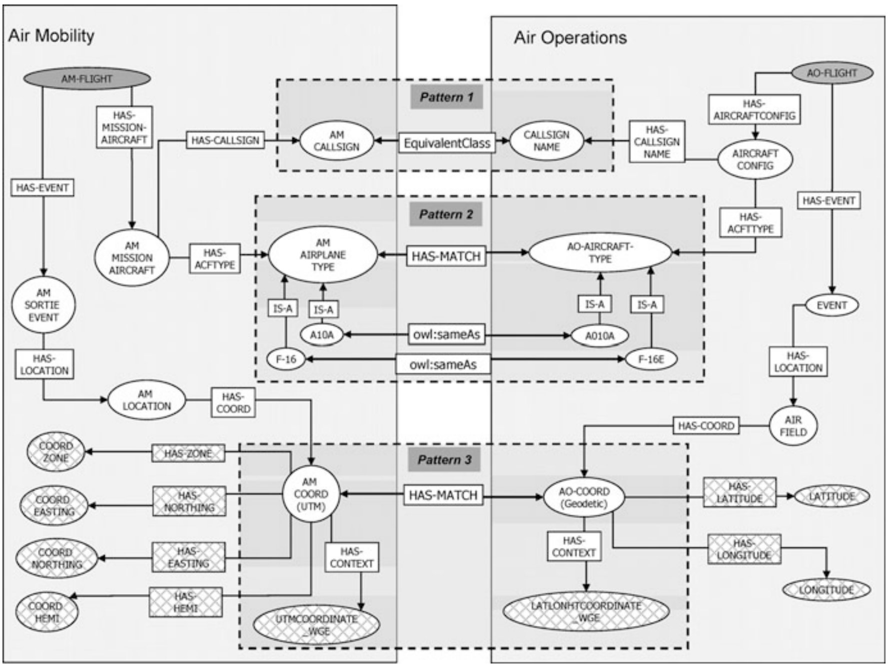



**Fig. 1.6** Alignment of diverging OWL ontologies of Air Mobility and Air Operations (source: [192])

HTML commands/tags. The scripts deal with input of resources (WSDL files, schemas, ontologies, and WSML scripts), with the alignment of concepts, and with workflow.

WMSL embeds mapping relations in HTML. Look at the encoding for a concept alignment (compare pattern 1 in Fig. 1.6) in the AM and AO ontology:

```
<dl class = "owl-equivalentClass">
<dt> <a href= "http://mitre.org/owl/1.1/AM#CallSign">
AM#CallSign</a><dt>
<dt> <a href= "http://mitre.org/owl/1.1/AO#CallSignName">
AO#CallSignName</a><dt>
<dl>
```

In Fig. 1.6 the OWL ontologies of Air Mobility (AM) and Air Operations (AO) are reconciled by three mediating patterns. Pattern 1 uses the simple equivalence of two concepts with different names whereas the match in pattern 2 depends on the 'owl:sameAs' identity of subconcepts on both sides.

## *1.2.2 Domain-Specific Description and Modeling: ResEval Mash*

While most mashup tools are domain-independent, the ResEval Mash [113, 114] is dedicated to a specific task with an own body of knowledge: research evaluation. The authors combine a generic mashup meta-model with a domain-specific description language (DSL) as a sublanguage. The DSL specifies a class of mashups, in the present case for research evaluation, using terms specified in cooperation with domain experts. The more abstract generic mashup meta-model is addressed by IT developers, e.g. for entering new components, whereas a graphical user interface with a visual DSL (Fig. 1.7) helps domain experts to set up their mashups for concrete tasks.

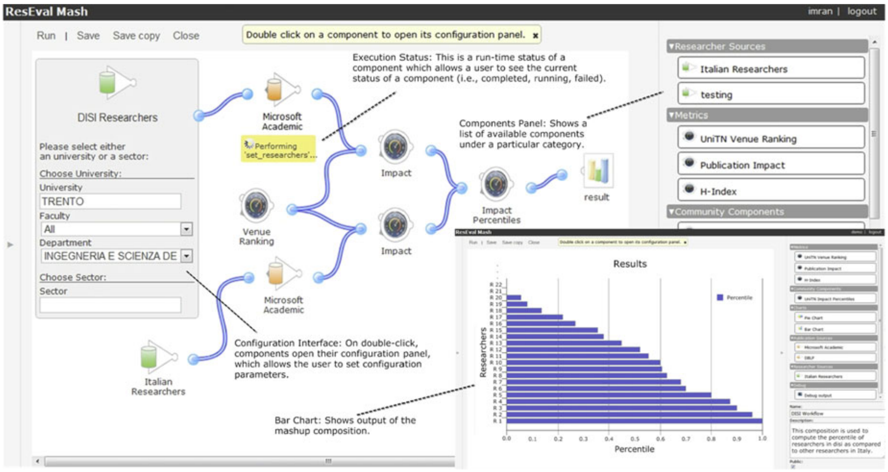



Fig. 1.7 ResEval Mash: The user interface (from [113])

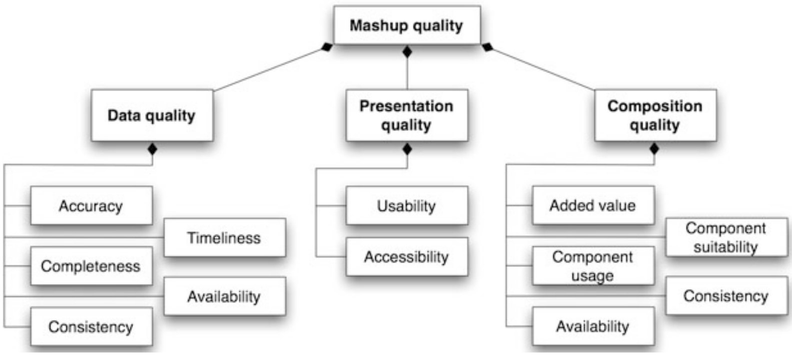



**Fig. 1.8** Mashup quality model (from [38])

```
<component id="googlemaps" name="GoogleMaps"
        description="Geolocalization service.">

        <operation name="centerMap" description="."
                similarity-meaning="CenterMap"
                similarity-verb="center" similarity-object="map">

                <param name="latitude" description="Latitude"
                        direction="input" type="xsd:decimal"
                        similarity-meaning="Geography.owl#Latitude"/>
                <param name="longitude" description="Longitude"
                        direction="input" type="xsd:decimal"
                        similarity-meaning="Geography.owl#Longitude"/>
        </operation>
        ....
        <event name="positionOnClick" description="..."
                similarity-meaning="ProvidePosition"
                similarity-verb="provide" similarity-object="position">

                <param name="latitude" description="Latitude"
                        direction="input" type="xsd:decimal"
                        similarity-meaning="Geography.owl#Latitude"/>
                <param name="longitude" description="Longitude"
                        direction="input" type="xsd:decimal"
                        similarity-meaning="Geography.owl#Longitude"/>
        </event>
        ....
```

```
<qualityAttributes>
        <reputation>1</reputation>
        <languages>
                <language>javascript</language>
        </languages>
        <dataFormats>
                <dataFormat>JSON</dataFormat>
                <dataFormat>XML</dataFormat>
                <dataFormat>VML</dataFormat>
                <dataFormat>KML</dataFormat>
        </dataFormats>
        <security>no authentication</security>
        <timeliness>0.9</timeliness>
        <accuracy>0.9</accuracy>
        <completeness>0.8</completeness>
        <availability>1</availability>
        ...
</qualityAttributes>
```

**basic quality features**

**Fig. 1.9** Google Maps description: Code sample for event-based handling on the *left* and quality attributes on the *right* (source: [38])

## *1.2.3 Mashup Quality—The PEUDOM Mashup Tool*

**On Quality:** [2, 12, 23, 36–38, 40, 45, 176, 185, 247, 249, 255]

Mashup content largely decides on mashup quality, so that external resources have a big impact on it. The rest of a mashup's quality results from good component integration and a well-designed visualization interface. The quality assessment of a mashup as a whole is puzzled together from the quality scores of its parts, so that it is complex enough for an explicit quality description or model.

The mashup quality model [38] displayed in Fig. 1.8 organizes its features in three dimensions: data of the components, the presentation on the user interface, and

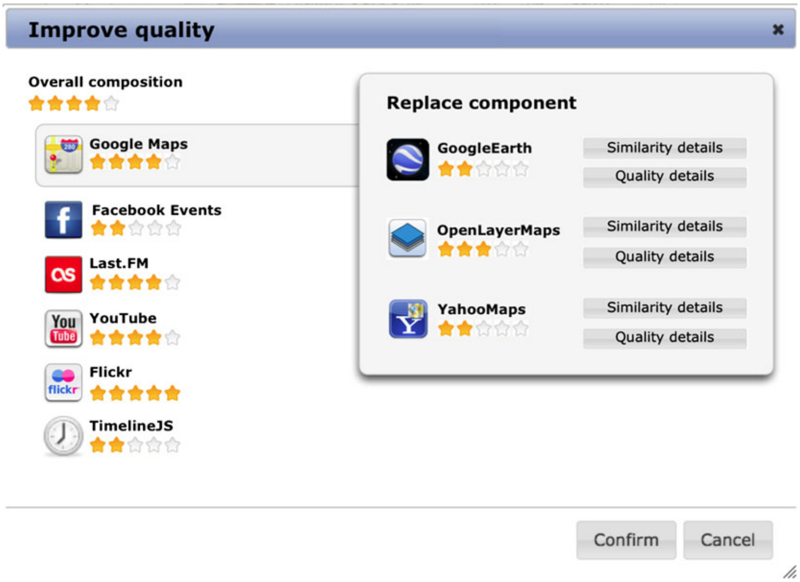



**Fig. 1.10** Google Maps with alternatives on the PEUDOM user interface, the alternatives being ranked by quality (source: [38])

the composition quality. Though quality criteria are context-dependent, the quality features for incoming API data are widely shared. This is because accuracy, timeliness, completeness, availability, and consistency are crucial for all mashups that reuse web services. In case of input problems, like missing data delivery from a chosen API, the mashup has to react, e.g. by switching to a substitute resource.

For quality assessment, component descriptions must be instantly available from a repertory [38, 185]. Figure 1.9 displays a sample component description in XML format for Google Maps, the most popular API of the web. It reappears on the user interface of the PEUDOM mashup tool (Fig. 1.10) as the first option with a set of possible replacements. The alternative map services are ranked according to their quality features (cf. left column and content of the green box in Fig. 1.9). The PEUDOM ranking mechanism combines several probabilistic technologies.

## 1.3 Semantic Mashups

**On Semantics:** [4, 9, 19, 20, 24, 25, 46, 48–50, 74, 78, 87, 88, 115, 116, 119, 131, 134, 137, 138, 143–145, 148–150, 157, 159, 160, 166, 172–174, 190, 193, 198, 200, 201, 222, 232]

Semantic mashups are at home in the semantic web, although they may also reach outside resources. As regular inhabitants they share the common semantic features of the semantic-web ecosystem. If one conceives the semantic web as being characterized by semantic annotations (markup or metadata) coded in RDF or OWL, the background of basic semantic mashups is all set.

By using semantic annotations, neutral mashups permute into semantic mashups. Where metadata, e.g. from an ontology, state which items web services offer (see Fig. 1.6 above), semantic mashups indeed improve the possibilities to choose and match the right input items [157, 198]. Semantic mashups with this profile convince many netizens. W3C provides SAWSDL[5] for semantic annotation of WSDL components via ontology referencing.

The restricted view on mashups explained a moment ago is having its defenders (e.g., [150]). A mashup can be semantic to different degrees. Lightweight semantics is well known in the semantic web.

The further reaching claim about the semantic web is that it achieves a deeper understanding of meaning than other ecosystems would enable. To speak the truth, some penetration into the meaning of content occurs almost everywhere in the web, albeit it may be very limited. Thus the semantic web only emphasizes a feature that was and is widely distributed, only we see that in the semantic web, meaning and semantics score much higher.

Now the illustration and application to semantic mashups:

- Meaning is handled almost in all web applications, but to different degrees. All mashups that deal with symbolic data, from interpreted fact databases to virtual reality, are assumed to be semantic unless they prove the contrary. Developers who feel to have a semantic-free mashup are invited to present it. Until this happens, one can put the non-semantics issue aside.
- Treating meaning is by no means restricted to markup, annotation, and metadata. Take information extraction as an example. It may use metadata, but just as well syntactic or semantic templates. Or look at mashups interpreting data via semantic rules or probabilistic methods (inspect the Black Swan below). Semantic mashups are semantic because they apply semantic methods—all available ones. Semantic mashups can contribute much more semantics than an alignment of data sources via metadata.

[119] presents an example. The authors innovate the classical RDF triple-store-based bookshop scenario (see below) with more internal intelligence. Mashup knowledge is stored in an ontology, Pellet[6] is used for reasoning.

WSDL/SOAP-oriented (semantic) web services are said to have not been as popular as expected. Probably mashups in their ecosystem do not fare better. [172] state the point and anticipate a new wave of services: linked services, mostly coded in RDF. As you may think, the more recent RDF data are still less established in the mashup ecosystem. By today (02-08-2012) only 65 APIs on ProgrammableWeb are of RDF format, the oldest ones from 2006. All DBpedia mashups fit on one page.

[5] http://www.w3.org/TR/sawsdl/.

[6] http://clarkparsia.com/pellet/.

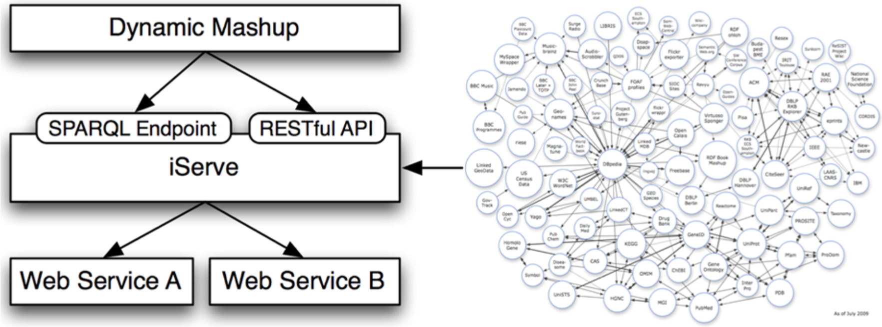



**Fig. 1.11** The dynamic mashup using iServe support (source: [143])

Work in the linked data mashups ecosystem is illustrated by:

- a dynamic mashups technology with iServe repository [143]
- the classic RDF book mashup [24]
- Semantic Web Pipes (SWP—[137, 138])
- the Black Swan mashup[7] for interpretation of rare events [145]
- the FlyBase Insitus mashup [159] for the Drosophila genome

## *1.3.1 Dynamic Mashups with iServe Support*

While traditional mashups tend to be static, dynamic mashups supported by a linked data server (called iServe [143, 174]) can select suitable resources during runtime. The mashups use linked data, but also other REST-based web services. iServe disposes of a repertory filled with annotated services. When a mashup requires a service, iServe looks it up in its directory, and in case of problems replaces it with a better equivalent. Thus the mashup delegates resource invocation and gains flexibility (see Fig. 1.11).

## *1.3.2 RDF Book Mashup*

In their classical RDF Book Mashup, the authors of [24] demonstrate how a mashup works in an RDF environment. Books and authors have URIs with an RDF description. A SPARQL query engine handles the search inside the RDF triple pool. The RDF descriptions contain outbound web links, in the present case to Amazon and Google APIs (see Fig. 1.12). They fill the local query result with additional

[7] http://blackswanevents.org/.

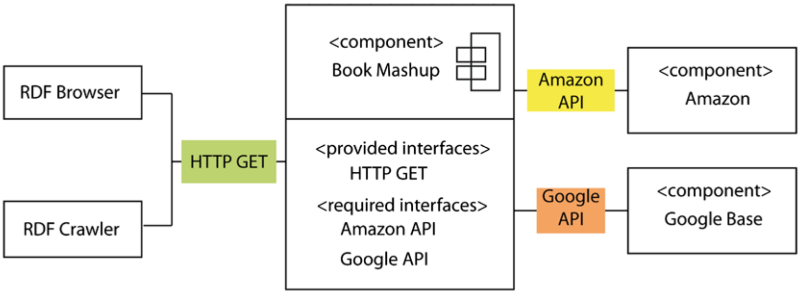



**Fig. 1.12** RDF book mashup structure, remake (source: [24])

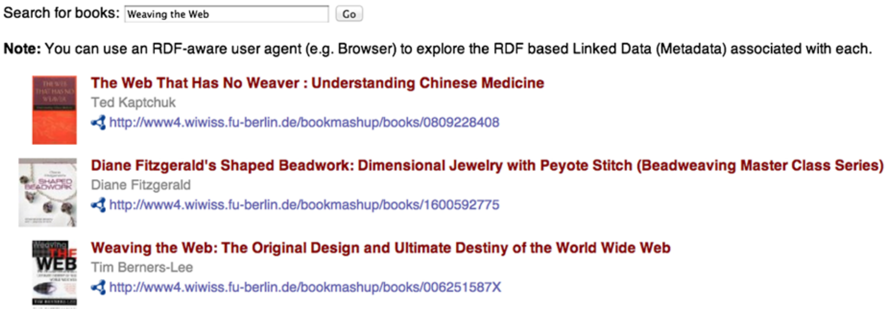



**Fig. 1.13** RDF book search results

data. A short PHP script manages the HTTP communication. An output snippet (Fig. 1.13) shows data imported from Amazon.

### *1.3.3 Semantic Web Pipes (SWP)*

Yahoo Pipes[8] are the most popular tool for end-user mashup development. With aggregated pipes (processes accepting inputs and delivering an output) users define their mashups on a graphical interface. [137, 138, 160] reconstruct the pipes approach for a semantic-web environment. The editor/graphical user interface is maintained in the well-known pipes style, but the operators change. Consider the fetches in Fig. 1.14: instead of the Yahoo-own 'Fetch CSV' or 'Fetch Feed', Semantic Web Pipes offers 'RDF Fetch', 'HTML Fetch', 'HTTP GET', 'Sparql Result Fetch', and so on. RDF, XML, Microformats, JSON, and binary streams are accepted. Pipes can be entered into other pipes. As with Yahoo Pipes, users can store and publish their pipes.

[8] http://pipes.yahoo.com/pipes/.

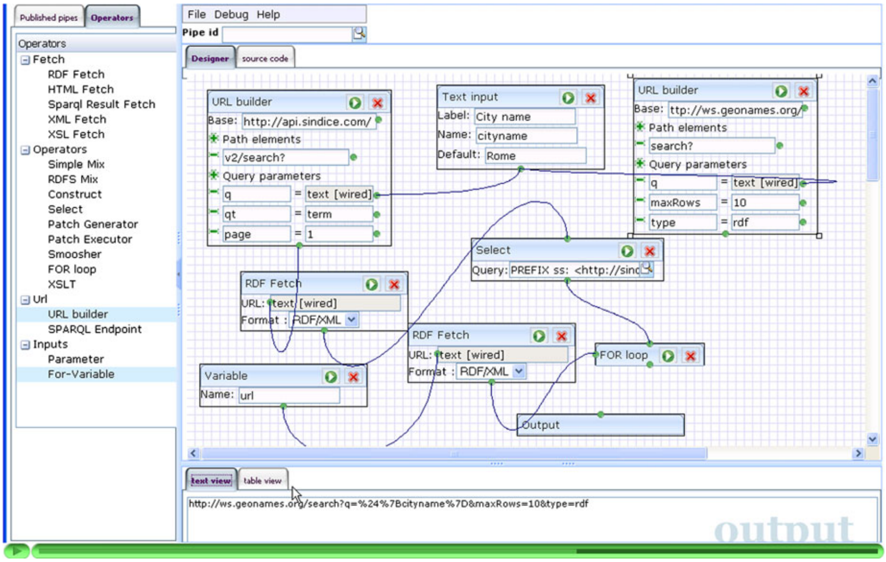


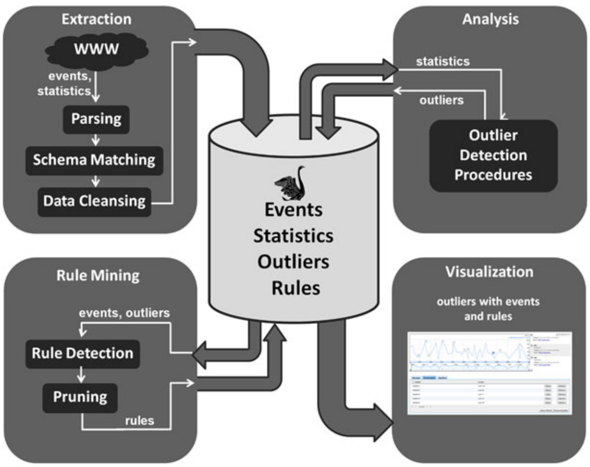

**Fig. 1.14** SWS mashup development (from http://pipes.deri.org/cityfacts.html)

## *1.3.4 Black Swan—Discovering Events That Matter*

Black swan events are as rare as black swans, but they exist.

The Black Swan mashup [145] enables users to explore timelines of statistical data. The timelines are decorated with factual event knowledge that may help to interpret the data. One may for instance ask whether a war outbreak influences the income per capita. Black Swan uses event data from DBpedia, Freebase, NOAA, Correlates of War, EM-DAT and BBC Timeline. Locations are imported

**Fig. 1.15** Black Swan. The architecture overview (from http://blackswanevents.org/?page_id=179)



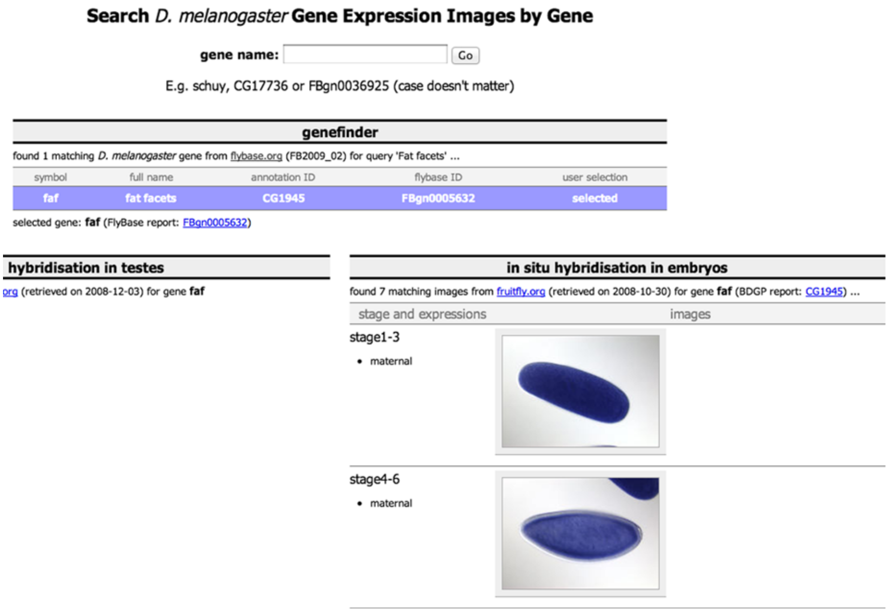



**Fig. 1.16** Insitus mashup from FlyBase—abridged, many images cut off

from GeoNames, statistics from GapMinder.[9] Black Swan exemplifies an intelligent semantic mashup because it heavily reworks resources before visualizing them for a user. Methods include information extraction, geo-localizing of data, rule mining, and several regression techniques (cf. Fig. 1.15).

### *1.3.5 FlyBase Insitus Mashup*

The FlyBase[10] contains genetic, genomic and functional data of the fruit fly (Drosophila Melanogaster). Drosophila Melanogaster is common, has a very fast generation alternation and a well-explored genome. The FlyBase collects the genome data from the research literature and other external sources/databases.

Insitus[11] [159] is a mashup integrating data from the Berkeley Drosophila Genome Project (BDGP) and from the Drosophila Testis Gene Expression Database (Fly-TED). It references the FlyBase for disambiguation of gene names. As shown for the fat facets gene in Fig. 1.16, it depicts expressions of genes in different states in embryos and testes. The implementation uses AJAX and SPARQL end-points.

[9] http://www.gapminder.org/.

[10] http://flybase.org/.

[11] http://openflydata.org/search/insitus.

## 1.4 Media Mashups

No mashup type outperforms maps/geodata mashups in popularity. Countless mashups visualize Google maps and locate some data on it. Together with GPS and location-based services (LBS) they help to find points of interest (POIs), for instance a restaurant nearby. Google, Yahoo, MapQuest and some others provide their maps for worldwide use. No wonder that in this book, three chapters are using geographical maps. Interested readers switch to Travel Mashups, Urban Mashups and/or Mashups for Emergency Management.

Compared to the popularity of map mashups, other species of the media mashup ecosphere remind one of the above-cited black swans: mashups integrating speech, video or augmented/virtual reality (AR/VR) are uncommon, but they exist.

The following presentation concentrates on the less known:

- Given the presence of map mashups in three book chapters, only crowd-sourced collaborative map mashups are discussed in this section.
- Speech mashups deserve to be mentioned. As a whole chapter in this book treats them, they are only briefly presented.
- An augmented painting and a weather webcam with integrated data from web services exemplify AR mashups. Virtual reality mashups come from Second Life.
- The Virtual Director and a mock-up emotional video mashup illustrate video mashups.

### *1.4.1 Map Mashups*

**On Maps:** [16, 17, 29, 33, 85, 97, 110, 132, 134, 140, 141, 182, 218]

More often than not, map mashups appear as the prime mashup type [33]. In common practice, users or developers apply a geo-map API and locate some of their own data on it. The general state of map development is reported in [182]. [134] explains how to enrich geoinformatics systems with semantics.

Inside GISs (Geographic Information Systems) map mashups incorporate neogeography[12] tendencies aimed at placing cartography into the reach of nonprofessional users and developers. Crowd-sourced map mashups are a fact: the community creates free editable maps in a wiki style [17, 140]. A notable example is OpenStreetMap.[13] Google Map Maker[14] follows the approach on the commercial side.

[12] http://en.wikipedia.org/wiki/Neogeography.

[13] http://wiki.openstreetmap.org/wiki/Main_Page.

[14] http://www.google.com/mapmaker.

In MapTube[15] of [17] a Java program called GMapCreator accepts an ESRI[16] (Environmental Systems Research Institute) shape file from Google maps. It generates an overlay of user-own data and puts this layer onto the Google map. GMapCreator displays the result as a Google Map in a web page format.

3D maps are available from Google Earth, Bing Maps, and others. In Google Maps, the 3D view is an integrated function. User additions are possible.

### *1.4.2 Speech Mashups*

**On Speech:** [68]

Speech mashups use web voice services. Multilingual voice-as-a-service (VaaS) options are available worldwide for all networked devices. Most popular are smartphone and tablet applications. Speech mashups are a core tool for assisting disabled web users.[17]

Web voice services perform speech recognition and text-to-speech processing on the provider's web server. A web client uses the services, so that it listens and talks without storing the (huge) databases that encode the voices. All voices for all languages in the provider's repertory are available.

The speech mashups chapter of the book explains the details as seen in the pioneering AT&T environment. If you want, have a look at a speech mashup iPad app[18] of the author. It supports writers during document revision with an Acapela[19] voice service.

### *1.4.3 Augmented Reality*

**On Augmented Reality/Virtual Reality:** [17, 72, 73, 82, 84, 133, 237, 238]

In augmented reality (AR) the image before the eyes of a user is enhanced with a computer-generated virtual part. A good example is disentangling the messed cables in a machine. It helps when an AR image presents the user with the target bundling of the cables and the step-by-step procedure of how to reach it.

#### 1.4.3.1 CAMAR—An Augmented Real-World Object

Consider an AR mashup that adds virtual information to a real-world painting in the user's environment. [238] calls this combination of a real object and related virtual

[15] http://www.maptube.org/.

[16] http://www.esri.com.

[17] http://www.research.att.com/projects/AssistiveTechnology/.

[18] https://sites.google.com/site/nospeech3/.

[19] http://www.acapela-vaas.com.

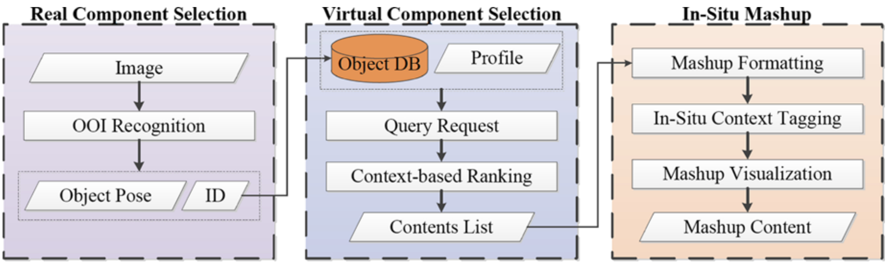



**Fig. 1.17** Scheme of mashup with real and virtual component (from [238])

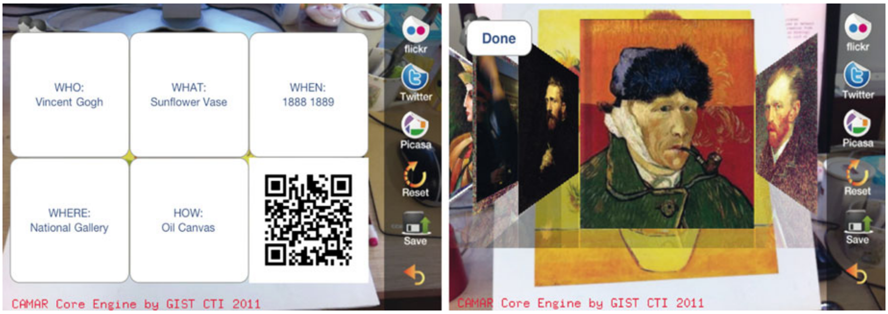



**Fig. 1.18** Bar code identification of the object of interest and augmented reality output (from [238])

content in-situ mashup:

> In-situ AR mashup is seamlessly combining additional contextual information to a real-world object to enrich content in one or more senses, where mashup process and its outcome are enhanced with context awareness and visualized with augmented reality for intuitive UI/UX.[20]

In the architecture sketch in Fig. 1.17 the mashup receives a bar code from the user's smartphone camera, identifies the current object of interest (OOI) in the real world (e.g. the painting) and superimposes annotations from a database, so that the mashup components come in part from the real world, in part from an object database. All information is visualized (see Fig. 1.18).

#### 1.4.3.2 Augmented Reality Weather Cam

[82] augments the image of a weather webcam on a building of the University of Münster. As Fig. 1.19 shows, the authors stack data layers from web services onto the basic layer of the image. For the bottom layer image, the camera is turned into

[20] UI: user interface, UX: user experience, more in [102].

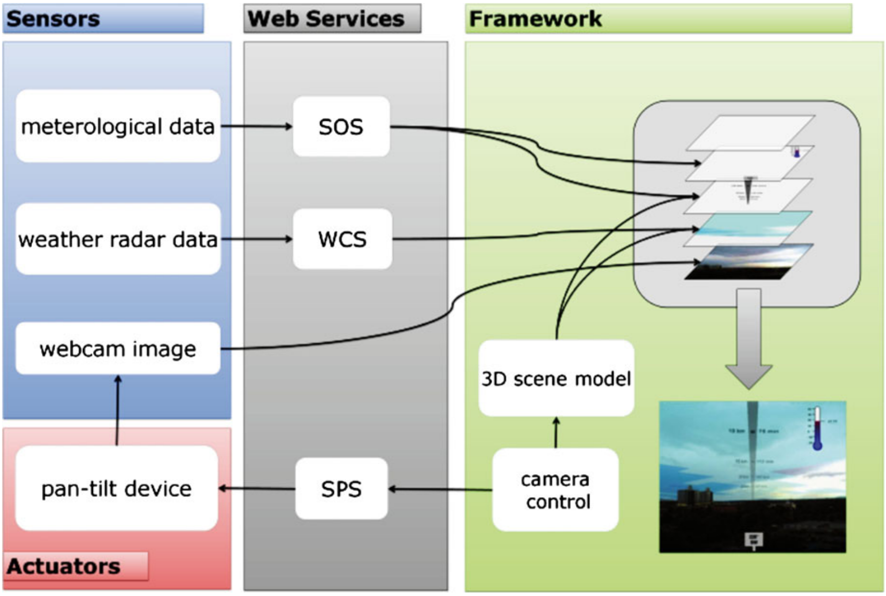



**Fig. 1.19** Weather webcam: Standardized sensor data added layer-wise onto the webcam image (from [82])

the wind direction with data from the Sensor Observation Service (SOS) and Sensor Planning Service (SPS). On the second layer the weather radar data are presented with the help of the Web Coverage Service (WCS). The third layer shows a combined scale with physical and temporal distance for weather/rain clouds to come in. On layer four, current temperature and wind speed are displayed. Textual data are added on layer five.

### 1.4.3.3 Virtual Reality: Mashups in Second Life

In virtual reality (VR) the whole interaction takes place in a computer-generated virtual world. [73] discusses mashups and their semantics in the virtual reality of Second Life.[21] Who wants to develop a Second Life mashup is invited to inspect the API.[22] ProgrammableWeb lists a few Second Life mashups.[23] [17] illustrates their maps in Second Life. Other mashups enter specific APIs like delicious or Flickr into Second Life. The Planespotting[24] mashup of Google Earth and Second Life tracks

[21] http://secondlife.com/.

[22] http://wiki.secondlife.com/wiki/Linden_Lab_Official:Map_API.

[23] http://www.programmableweb.com/api/secondlife/mashups.

[24] http://nwn.blogs.com/nwn/2007/09/planespotting-g.html.

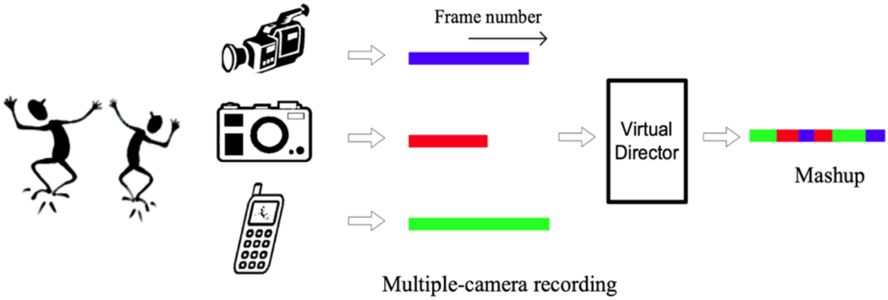



**Fig. 1.20** The Virtual Director integrates multiple camera recordings into a video mashup (from [203])

planes at LAX (Los Angeles) airport. An up-to-date mashup makes the iKnow app[25] available inside Second Life, so that speakers of Japanese can improve their English vocabulary in virtual reality.

### *1.4.4 Video Mashups*

**On Video:** [5, 34, 42, 162, 202, 203, 249]

A video mashup is the result of combining multiple audiovisual sources. It is a product with its own identity, so that its meaning/semantics can widely deviate from the content of the source videos.

#### 1.4.4.1 Virtual Director: Mashup of Multiple Event Recordings

[203] integrates multi-cam recordings of events into a video mashup. Figure 1.20 illustrates the situation. Input movies may come from all sorts of devices and resources (camera, webcam, web resource, smartphone, etc.) with different technologies, so that they differ in angle, duration, frame rate, part of the event covered, sound quality, image quality (e.g. shaking, dizziness, lighting, etc.). A mashup choosing the best frames of all movies reaches a better quality and avoids the boring effect of one persistent camera and view angle. With a suitable tool, an end-user can generate such a video mashup. A professional creator can do the same with a higher artistic endeavor. The resulting mashup is a single video stream as known from the early music and video mashups. The Virtual Director follows user requirements collected in an explorative study with 18 video camera users. The users asked for: synchronization, image quality, diversity, tuning to user preferences, suitable point cuts, semantics, suitable segment duration, and completeness.

[25] http://iknow.jp/.

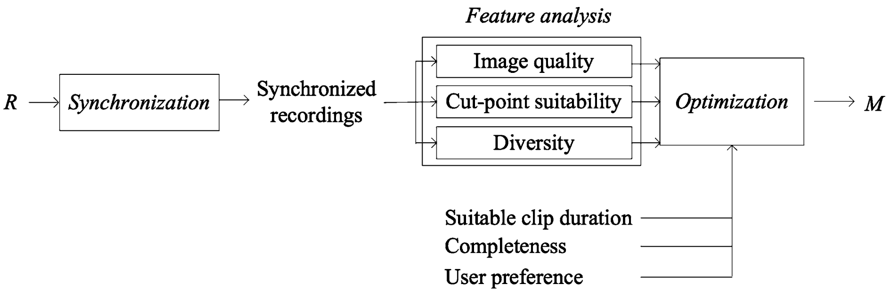


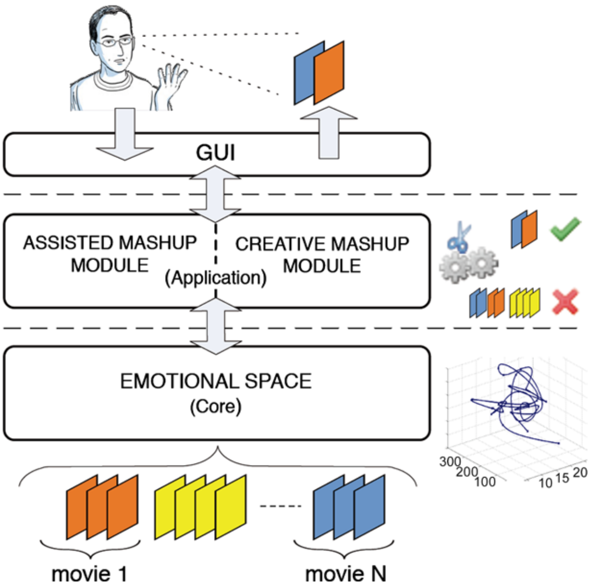

**Fig. 1.21** Virtual Director overview of the mashup process. $R$: recording, $M$: mashup (from [203])

The overall mashup quality is represented by an objective function that combines the requirements. The segments for a mashup are selected so that the function is maximized. Figure 1.21 shows how the system proceeds.

#### 1.4.4.2 Emotional Video Mashup

In a movie directors interleave the emotional stimulation(s) for the moviegoers. An emotional video mashup [34] (see Fig. 1.22) follows this practice. The authors build their system around an emotional space with the dichotomies warm/cold, dynamic/slow and energetic/minimal on its axes. The emotional development during the movie trajectory is recorded. Mashups are expected to keep the mood while being composed from different sources. Users composing their movie can choose between an assisted mode and a creative mode of mashup building.

**Fig. 1.22** Emotional video mashup creation (from [34])



When a user proposes to mix a movie in, the best fit shots are calculated by Euclidean distance in the emotional space. A ranked list of emotionally good shots is offered to the user.

## 1.5 Ubiquitous Mashups

The ubiquitous web covers the whole web and reaches out into the physical world. It can be seen as a synthesis of the web and ubiquitous (pervasive, ambient intelligence) computing. Ubiquitous mashups may reside on all sorts of networked units—for instance sensors, mobiles, navigation systems, intelligent appliances. The Manhattan Story Mashup of [217] made pervasive devices from phones to large public displays support a big public cooperative story telling event.

Ubiquitous applications may dip into very different contexts. A simple example is an app being projected on a phone, a tablet or a huge TV screen. Or change the user: the mashup happens to be managed in Russian, Italian or Japanese instead of English. Or tune the depth of context penetration: a sensor of a mashup may sit on a lamp post, but think of a heart rhythm monitor under the skin of a patient. Context awareness is a first-order issue with the multitude of situations where ubiquitous applications/mashups may need to accommodate and to perform.

After dealing with context awareness, the report focuses on mobile mashups in different surroundings, sensor mashups, and (embedded) physical mashups.

### *1.5.1 Context Awareness*

**On Context Awareness:** [21, 31, 43, 44, 60, 70, 77, 106, 123, 124, 128, 144, 178, 180, 181, 188, 194, 214, 215, 217, 220, 225, 237, 253, 254]

In principle, a context-aware system executes its main job while respecting some context features. Figure 1.23 displays this basic view. There the system adapts automatically, for instance by raising the voice in noisy surroundings. Alternatively it might ask the user to adapt the sound level. The adaptation to the user and the use situation/task is often called personalization.

For a mashup, context-driven accommodation may begin during service selection and content composition. Two examples:

- If the web communication is weak, a service from nearby is a better choice than an overseas mirror.
- For presentation on a smartphone, mashup content must be restricted to a minimum, while on a tablet, one can afford to spread more information.

[181] lists more reasons for the context-aware mashup modification. [180] proposes a context-aware mashup model including an event-based submodel for adaptivity.

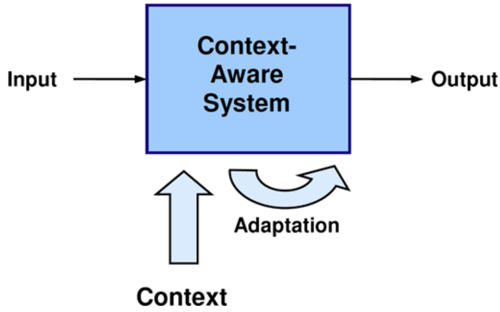



**Fig. 1.23** Context-aware system with active adaptation (source: [254])

## 1.5.2 *Mobile Mashups*

**On Mobile Mashups:** [7, 15, 21, 26, 30, 31, 41, 44, 55, 56, 68, 77, 81, 106, 112, 118, 120, 128, 144, 161, 175, 183, 195, 204, 213, 223, 224, 226, 236, 248, 249]

Mobile mashups are at home on mobile devices such as smartphones or tablet computers.

As they are made for changing their location all the time, smart devices need ubiquitous computational logistics, e.g. a wireless internet connection. Smartphones feature sensors. The most popular ones pick up GPS location data. There are a camera and a microphone so that the device perceives local context data. Monitoring the surroundings is possible and useful, so that context-aware, adaptive, and personalized applications prosper. A smart mobile web device has to adapt to the user and to the local context, thus improving the user's grasp on the surroundings.

The cast for the most instructive mobile mashups yielded four players with different context-awareness behavior:

- TELAR introducing POIs (Points of Interest) of [30]
- Cooperating mobile mashups [183]
- Personal health mashups [21, 213] digging deep into the everyday life of their users
- Telco mashups integrating telephone and web services

### 1.5.2.1 TELAR: Mobile Mashup with Context Awareness

The TELAR mobile mashup platform [30] is designed for a Nokia N810 Internet Tablet. It is implemented as a client–server system. End-users can configure mashups running on the mobile browser. Web services and context information of local sensors are integrated, so that the mashup adapts to the current location of the phone (see Fig. 1.24). The Google Map is centered to this position via GPS data. Three web services for Points of Interest (POIs) are mapped in: Fon, Panoramio, and Wikipedia. Local sensor data are handed over in a DCCI specification.[26]

---

[26] http://telardcci.garage.maemo.org.

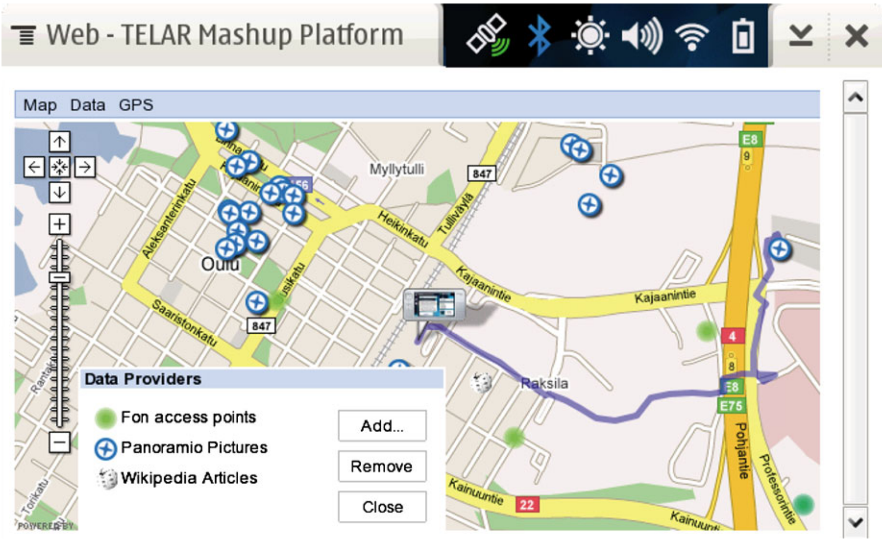



Fig. 1.24 Mobile TELAR mashup sample (source: [30])

### 1.5.2.2 Cooperating Mobile Mashups

Imagine a group of cooperating mashups: one host and two guests. The host is on an Android phone, the guests/clients are on iPhones. The group (proposed by [183]) compares prices while their users are shopping in a department store. On behalf of their users the guest mashups scan bar codes of interesting goods. The host/server collects them and asks the Google Search API for Shopping[27] for reference prices. It communicates the results to all cooperating mashups via normal SMS or email channels.

Custom agents implemented on the guests and a custom communication center on the host execute the communication processes. Figure 1.25 shows the user-side configuration. Users/mashup composers specify the intended mashup configuration in an XML-based description language called C-MAIDL (Cooperation—Mobile Application Interface Description Language). A mashup generation engine realized in Java reads the description and produces the cooperating mashup applications. Figure 1.26 explains the generation process.

### 1.5.2.3 Personal Health Mashup

Change of scene—please consider the scenario for the personal health mashup with computerized and connected consumer appliances. Among other things, there are

[27] https://developers.google.com/shopping-search/?hl=de.

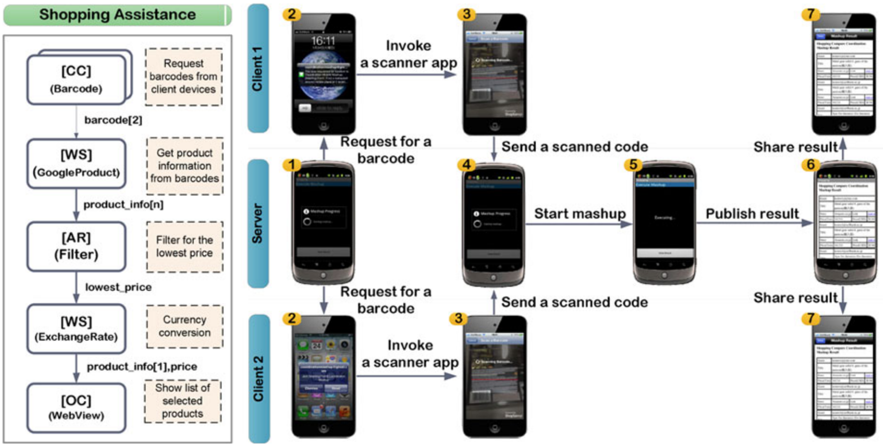



**Fig. 1.25** Shopping assistance by cooperating mashups (source: [183])


**Fig. 1.26** Mashup generation process for cooperating mashups (source: [183])

bathroom scales[28] that record and upload their users' weight. WiFi activity trackers[29] count the steps and the stairs taken on a day, the hours of sleep and so on. From the smartphone calendar the user's appointment load can be retraced. The users may provide some other data, for instance about their meals. Public sources can add context information, for instance about the weather or about traffic jams. All user-related data can be interpreted statistically with public statistics data as background.

The mashup can present results on common health factors depending on the own behavior so that users understand relationships that they normally do not know. The risks of fast food might be spelled out. Some practical advice can be given, such as: "More steps are better for your health!". Monitoring and recommendations can improve individual well-being, because individuals know more about their own behavior and possibly improve it.

[28] http://www.withings.com.

[29] http://www.fitbit.com/product/features.

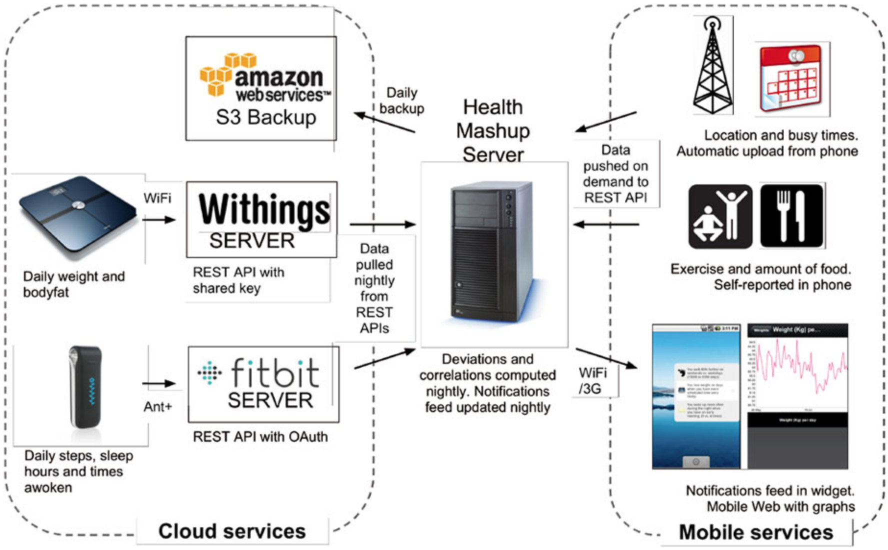



**Fig. 1.27** The Personal Health Mashup server (from [213])

This sort of health monitoring is offered by the personal health mashup [21, 213]. It has been developed and tried out with a small group of users from Stockholm and Chicago.

The mashup is server-based. Figure 1.27 shows how webdata communication works. The Withings scale and the Fitbit trackers upload the measured sensor values; the mashup obtains them from the public APIs of the respective companies, so that the following values are automatically available: weight (Withings scale), body fat (Withings scale), step count (Fitbit), hours slept (Fitbit), times awoken (Fitbit), location (phone), hours busy (in calendar). As said above, users may log details on their food and exercise.

The main contribution of the mashup server in the clouds is calculation: aggregating individual health parameters with a statistical correlation/digression evaluation. The reference data seeds were drawn from the test group in a first assessment phase. Statistical calculation runs by night on the mashup server. RSS feeds distribute the results to the users.

Users have two health mashup widgets on their phone or desktop. An output widget presents mashup statements on the user's health data (see Fig. 1.28, left image). When clicked, it opens a screen with more detail, e.g. value timelines (right image of Fig. 1.28). The second widget (not shown) accepts user-written food and exercise descriptions.

Besides the technical results the user test revealed some social behavior influencing the application of the mashup. For instance, sensor data (e.g. of the scale) were missing when users felt bad that day. It took some time till the users gained more


**Fig. 1.28** The output widget with weight timeline extension (from [213])

insight into the conditions of their personal well-being and reacted to the data. 8/10 lost weight during the test.

#### 1.5.2.4 Telco Mashups

Telco ("Telecom") applications and mashups combine services from telephony with web services [81]. Telco mashups are mashups as seen from the viewpoint of telecommunication companies:

> We define a telco mashup as a Web mashup that, in addition to optional data, application logic, and UIs, also integrates telco services or device APIs to support communication and collaboration among multiple users or provide them with individual telco features (such as an advanced GPS navigation mashup).

The main technical point is to integrate pre-existing (heterogeneous) communication networks such as the Public Switched Telephone Network (PSTN) or the Universal Mobile Telecommunications System (UMTS) via network gateways of telco operators. Figure 1.29 illustrates the web communication: in telco mashups, all internet access passes the gateways of telecom operators. [56] focuses on the telco mashup project OMELETTE for end-users.

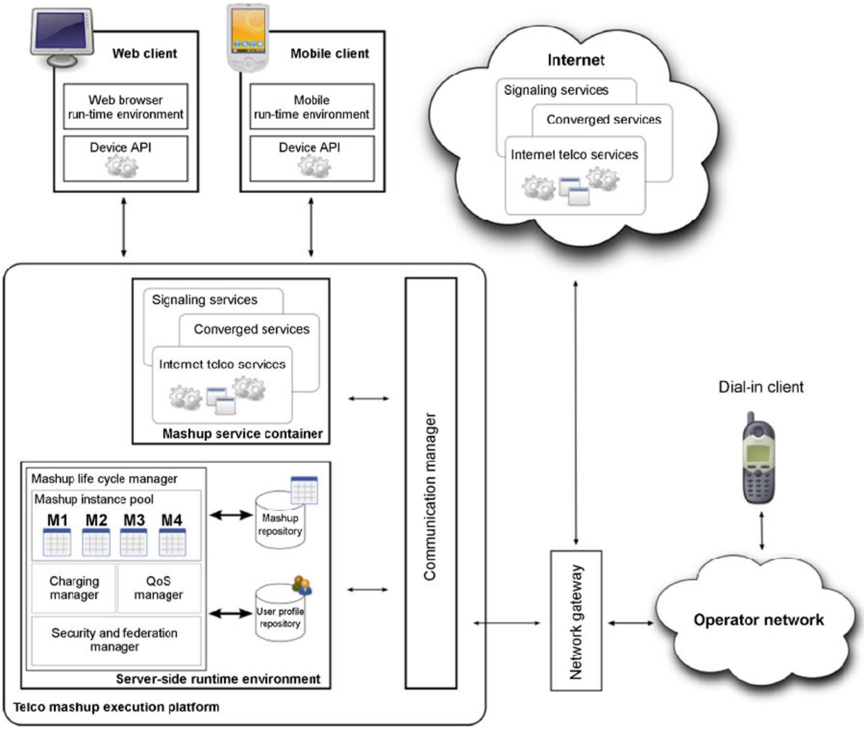



Fig. 1.29 Working environment of a telco mashup (from [55])

## 1.5.3 Sensor Mashups

**On Sensors:** [13, 27, 29, 57, 67, 83, 87, 88, 106, 125, 130, 135, 136, 138, 139, 156, 163, 167, 197, 199]

We are already talking about sensor mashups in the web: The mobile health mashup described above uses sensor data of the bathroom scale. They are uploaded from the scale (the sensor) to the scale producer's (Withings) web server and reach the mashup from the server. Instead of a weight measured in kg, an application might also receive a temperature measured in degrees Fahrenheit by a sensor in Key West FL and transmitted by some web service. Now that this basic principle of the sensor mashup ecosystem has been recalled, readers are invited to inspect two specific inhabitants:

- the SensorMasher [135] working in the Linked Data Cloud [136, 138]
- the web-of-things kit WoTKit [27] serving sensor data to portals and pipe-style mashups

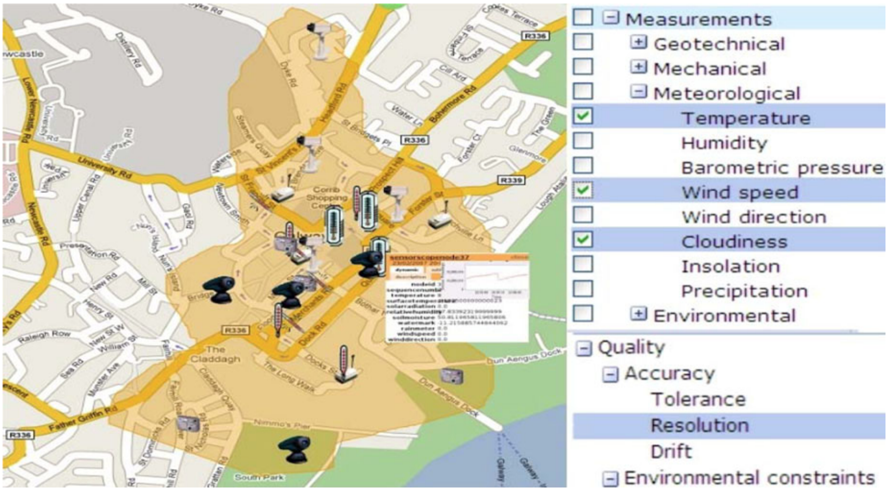



**Fig. 1.30** Faceted selection of sensor feed on the SensorMasher user interface (from [135])

### 1.5.3.1 The SensorMasher

The SensorMasher[30] [135, 136, 138] is a platform for setting up sensor mashups from linked data resources. It loads raw sensor feeds as scheduled by user-own annotations. The sensor feeds can be used for setting up mashups, and the mashups with their linked open data resources can be published as linked open data for later reuse.

As sensors and their data are web resources identified by URIs, they can be linked into an RDF graph. For the SensorMasher the graph is virtual (with distributed components). As required by the user's annotations, ontology concepts from the SensorMasher ontologies are activated together with their stream data structured in RDF triple form. Data can come from anywhere in the linked data cloud. Sensor data from several sources can be integrated into a new data source.

For the SensorMasher interface, users can select suitable feeds and the values they want to obtain (see Fig. 1.30). [136] explains how the user query is reworked and executed.

### 1.5.3.2 WoTKit—A Lightweight Toolkit for the Web of Things

WoTKit[31] [27] is an easy-going tool that enables end-users to adopt working with sensors in the web of things (WoT). By designing a sequence of small WoT appli-

[30] http://sensormasher.deri.org/.

[31] http://demo.sensetecnic.com/SenseTecnic/login.jsp, documentation on http://www.sensetecnic.com/mediawiki/index.php?title=Documentation:User_Documentation.

cations the authors framed out which requirements their WoT toolkit should fulfill. They list:

- Simple integration between a variety of things, both physical and virtual, and the toolkit
- Easy to use visualizations of data from a thing, and user interface to control things remotely, using the web
- An easy to use information processing capability for simple data processing and alert generation
- The capability for users to share their integrated things and other toolkit components with others
- The ability to scale up simple prototypes to more advanced applications by providing a comprehensive and easy to use API

WoTKit aggregates sensor data, visualizes them and combines them into mashups. On the GUI users can select public sensors (see Fig. 1.31); they can also register own ones. Sensors are annotated with some basic properties. Their value curve can be visualized. The processing is defined with (Java) pipes that appear on the interface in the proved and tested Yahoo Pipes style. Users can write Python scripts in order to develop custom solutions.

The architecture overview (Fig. 1.32) shows that WoTKit uses RESTful APIs for web communication. Sensors, actuators and web services are attached with gateways. There is no mention of semantics in the paper of [27], so that their WoTKit contrasts with the explicitly semantic SensorMasher above.

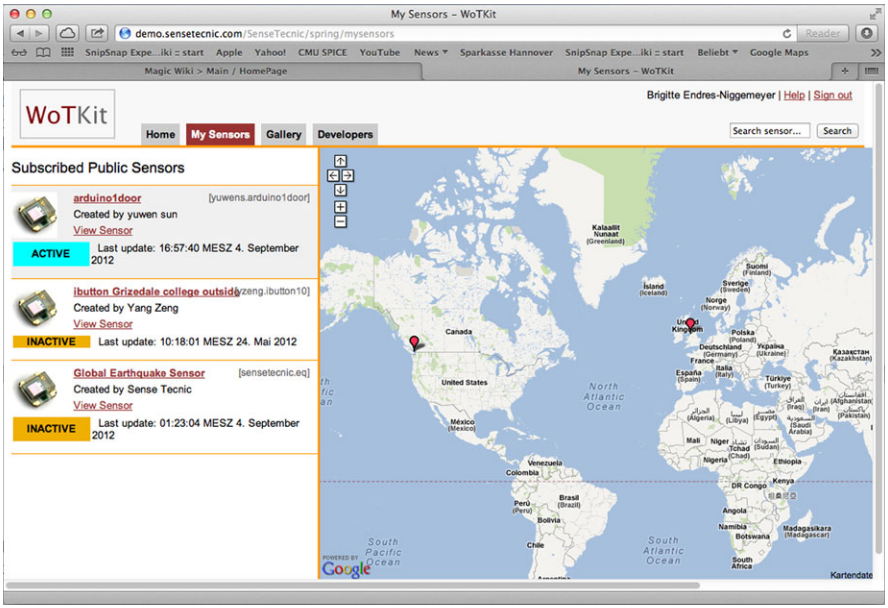



Fig. 1.31 Subscribed sensors of the author after a few minutes of testing

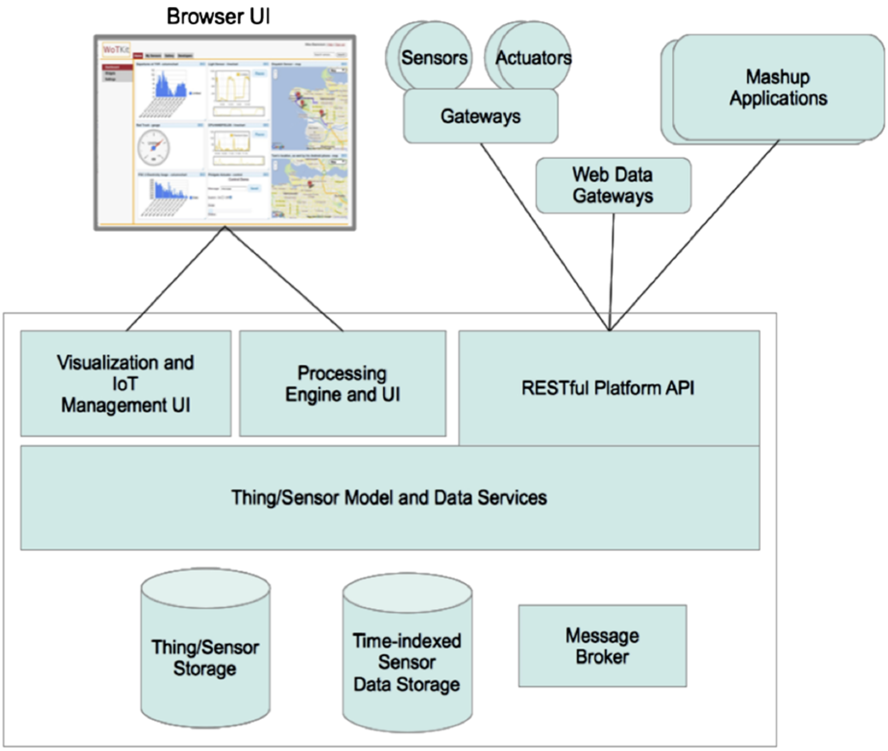



**Fig. 1.32** WoTKit architecture (from [27])

## *1.5.4 Physical Mashups*

**On Physical Mashups:** [90–95, 121, 122, 127, 129, 153, 154, 211]

Smart things are sensor and actuator networks, embedded devices, electronic appliances and digitally enhanced everyday objects. They have their place in the physical world, but they can be connected to the web with common techniques (JavaScript, AJAX, HTML, REST) and some additional methods e.g. for RFID processing [92, 93]. As soon as they have their URIs/presence on the web, they can be browsed, searched, steered and changed there. Figure 1.33 illustrates a web-enabled washing machine.

"Physical" mashups can refer to their URIs and to other web resources. Figure 1.34 shows two ways to link smart objects to the web. They enter directly via IPv6 lowpan[32] (IPv6 over Low power Wireless Personal Area Network) if they have an own web server. If this is not the case, a smart gateway mediates between the possibly proprietary communication of the objects and the web technologies. Smart

[32] http://de.wikipedia.org/wiki/6LoWPAN.

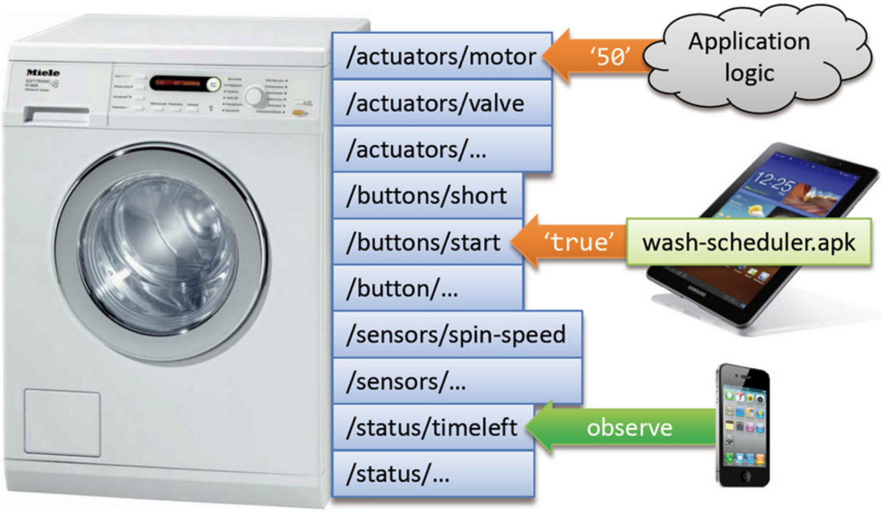



**Fig. 1.33** IP-enabled washing machine (from [129])

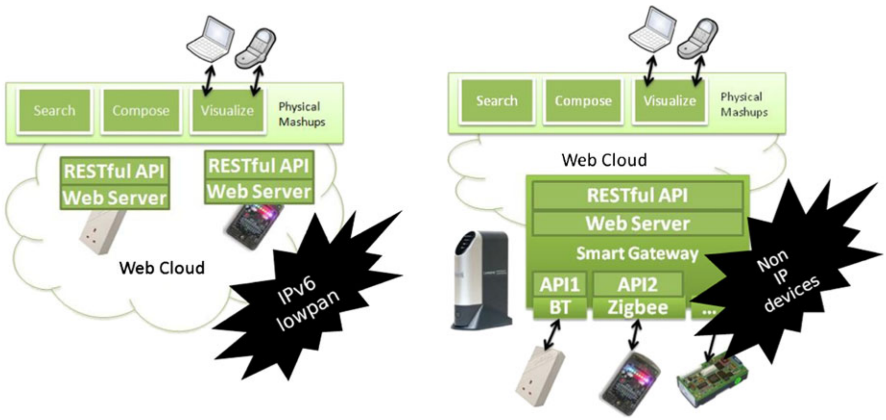



**Fig. 1.34** Smart things linking to the web—directly (*left side*) or with the help of a smart gateway (*right side*) (source: [90])

homes are a good example for web-controlled environments. They may have a web-based control of the main appliances so that they can be controlled from outside. A physical mashup for at home is likely to be an app that links to the devices of your household, using IPv6 or a gateway—why not as shown in Fig. 1.35?

Mashups are free in their choice of resources. They provide a new service from whatever is useful, so that web services/APIs of all sorts are welcome:

- Mashups accept output of SPARQL endpoints.
- They integrate user interfaces as a service.
- They accept sensor data from smart things.

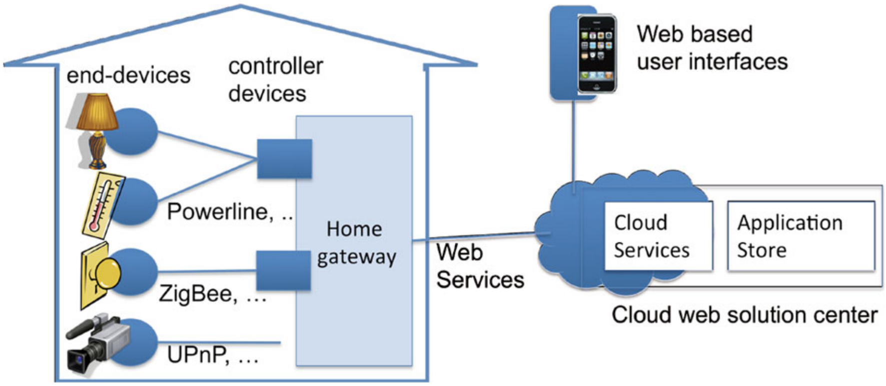



**Fig. 1.35** High-level architecture for the smart home. Appliance control from a smartphone app (source: [127])


**Fig. 1.36** Social mashup on Facebook announcing high temperature event at home (source: [121])

After this warming-up the scene is prepared for the Facebook-based Sociale Homer [121] (see Fig. 1.36). Homer expresses a smart home as a social experience for all household members. Besides overviews it also reports on preset events. In the example in Fig. 1.37, the temperature at home has exceeded 30 degrees Celsius. Possibly some actor should react, e.g. close the sunblinds or activate the air conditioning.

## 1.6 End-User Mashup Creation

ProgrammableWeb[33] advises users who consider developing an own mashup to weigh their coding skills. They also refer them to two exemplary interface-based

[33] http://www.programmableweb.com/howto.

mashup tools, but with emphasizing their limits. Quite to the contrary, the introduction video[34] of the classical Yahoo Pipes promises that an own pipe is an effort of some minutes. Anyhow the list of available pipes is long, so that many users managed to build pipes the Yahoo way. Certainly tech-savvy prosumers are better off.

End-user development (EUD) for mashups is one of the background trends for better user integration. [32] pass an EUD definition due to Lieberman:

> a set of methods, techniques and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify, or extend a software artifact.

Seen from the EUD perspective a mashup appears as an

> end user driven recombination of web-based data and functionality [86].

Mashup tools pursuing EUD practices should support active users in their development or adaptation work, thus recapitulating convictions inherited earlier from hacking, mashing, gluing [98, 100].

The following sequence of user-related approaches presents:

- Empirical user requirement studies
- Recommendation systems for helping users in their service selection and combination, featuring Baya [54] as highlighted realization
- Approaches for disburdening users from programming tasks by interface support via spreadsheets, programming by example or demonstration
- Widgets on the user interface, illustrated by WIPLE for widget-based personal learning [206, 207]

### *1.6.1 User Studies*

**On End-User Development:** [3, 22, 28, 32, 35, 39, 51, 54, 58, 63–65, 75, 86, 98, 100, 107, 108, 111, 158, 170, 186, 189, 206, 207, 212, 219, 227, 236, 241, 243–246, 251, 253]

It is good to keep the divergent views on end-user mashup reality in mind. Many mashup tools for end-users have been proposed, but success rates differ. [227] distinguishes four types of mashup interfaces:

- based on a flow-chart model (Yahoo Pipes since 2007)
- using a spreadsheet approach (the authors' Mashroom from 2009)
- following a tree pattern (Intel Mashmaker [75]—gone in 2008)
- integrated in a browser (Mozilla Ubiquity—finished in 2009[35])

[34] http://www.youtube.com/watch?v=J3tS_DkmbVA.

[35] https://wiki.mozilla.org/Labs/Ubiquity.

Yahoo Pipes is alive, Mashroom is recent, Mashmaker and Ubiquity have already been left behind. Besides registering the four types one observes a moderate survival rate of mashup tools. It seems that, although users are main players in mashups, a matter-of-fact mashup development considering user needs is still out. One may ask why. Listen to [244]:

> However, much of the recent work in this area focuses on the building new tools, but not realizing the needs of the end-user. Rather than concentrate on the tools and technologies surround the web, we should seek better ways of engaging the user. (sic)

This view is confirmed by [51]. Even in the most successful Yahoo Pipes the authors find constructs that exceed non-programmers' understanding.

User studies are a valid scientific reaction to this state of affairs. [22, 253] investigate the acceptance of consumer mashups. They apply the Unified Theory of Use and Acceptance of Technology (UTAUT) model and discover several incentives for users setting up a mashup. In their review the expected performance of the mashup is most motivating, in particular if it helps in organizing everyday duties. [64] reports on a requirements elicitation study with accountants. [212] advocates requirements engineering in mashup tools development, with some short hints to empirical requirements elicitation.

## *1.6.2 Recommendation Systems*

**On Recommendation Systems:** [1, 48, 51–54, 70, 74, 89, 118, 146, 147, 168, 169, 175, 176, 184, 250, 252]

Recommender systems are intended to support users setting up their mashups. Help may be restricted to advice, e.g. by listing good components, but it may also take over more of the mashup composition. There are plenty of approaches, for mashup composition among many others [1, 89, 250]. [168, 169] propose recommendations in a linked data environment. [146, 147] leverage social networks. [175] presents a mobile mashup with a recommender fed from semantic annotations.

### 1.6.2.1 Recommending Community Composition Knowledge: Baya

Baya [52, 54] offers to its users preset patterns harvested from earlier users, so that composition knowledge is propagated. It is a Firefox extension for Yahoo pipes.[36] On its screen (see Fig. 1.37) it offers mashup components left of the canvas in the middle. On the recommendation panel on the right users can select patterns and other predefined constructs. They come from a persistent pattern knowledge base on the Baya server that is offered as a service. The Baya recommendation

[36] Demo and explanation at http://www.youtube.com/watch?v=RNRAsX1CXtE.

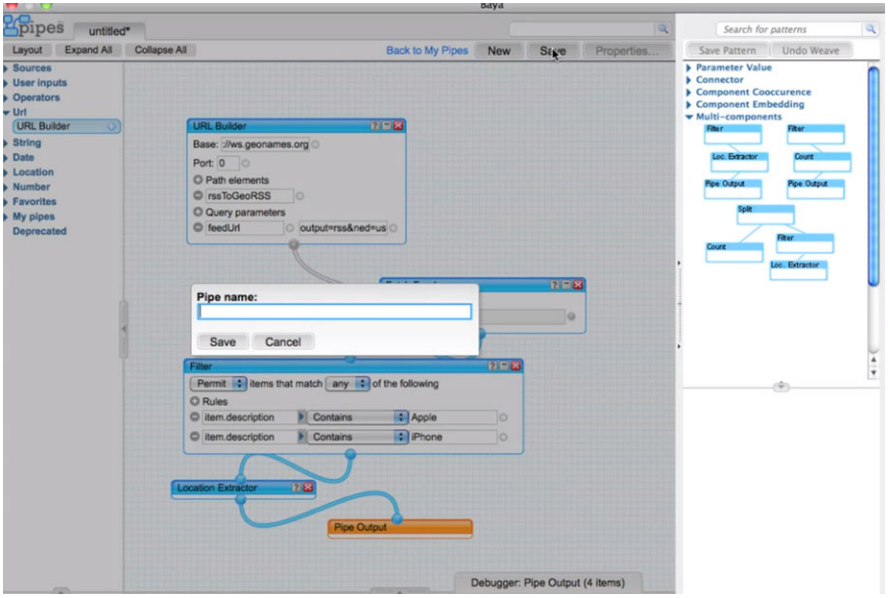



**Fig. 1.37** Baya assisted mashup creation. Components on the *left* of the canvas, recommended patterns on the *right*

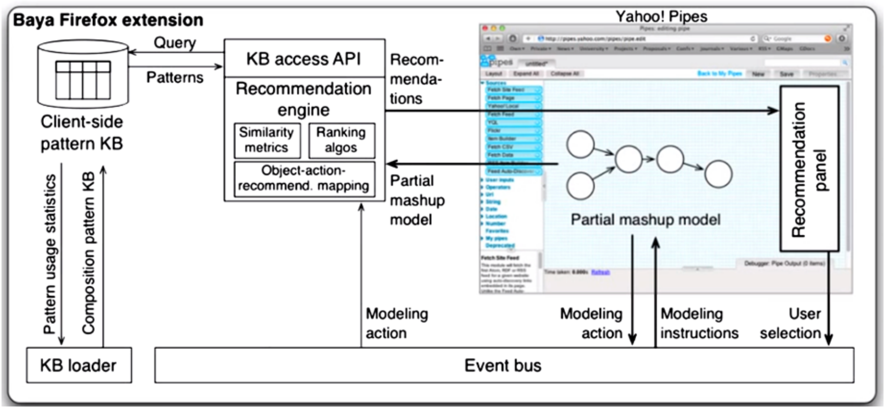



**Fig. 1.38** Baya in recommendation activity

server harvests community composition knowledge and integrates it into a canonical mashup model so that good patterns can be reused.

Figure 1.38 illustrates a recommendation event: user interactions are fed into an event bus (at the bottom of the drawing) which delivers them to the recommendation engine. If the user touches an object, the recommendation engine gets possibly matching patterns from the knowledge base. They are ranked and entered into the recommendation panel.

#### 1.6.2.2 Spreadsheets and Programming by Demonstration

**On Programming by Demonstration:** [11, 75, 76, 99, 104, 142, 210, 216, 227, 235]

One option in interface design is meeting users in surroundings they are used to. For good reasons, (Excel) spreadsheets are popular. So why not put a mashup in a spreadsheet design? With their integration of data and programming, spreadsheets inspire the above-mentioned MashMaker [75, 76], Marmite [235], the Mashroom system [227], and the distributed mashups of [11].

Programming by demonstration is offered by Karma [216], d.mix [99], query by example is used in the UQBE [210] and MashMaker mashup tools.

#### 1.6.2.3 Widgets on the User Interface: WIPLE

**On Widgets:** [6, 15, 56, 103, 205–207, 228, 233, 234]

Widgets are data elements drawn from incoming web resources/services with an accompanying presentation format. They give users an integrated unit that easily accepts their parameters. Widgets as used in Yahoo Pipes have widely conquered the mashup interface, so that they are first-class citizens of the mashup ecosystem. Mashup widgets may comply with the W3C widgets recommendation[37] (so in [56, 206, 234]):

> …widgets as specified in this document are full-fledged client-side applications that are authored using technologies such as HTML and then packaged for distribution. Examples range from simple clocks, stock tickers, news casters, games and weather forecasters, to complex applications that pull data from multiple sources to be "mashed-up" and presented to a user in some interesting and useful way.

WIPLE [206, 207] is a widget-based personal learning environment for language learning.[38] On its interface widgetized mashups appear as boxes that display a mashup result, or as dashboards with a set of services to the disposition of the user. The authors emphasize widget interoperability (also a topic of [103]), so that users can transmit data from one widget to another. This is exemplified in Fig. 1.39. There the user taps the word "car" on the mediatic widget. The related Flickr widget is notified and displays car images. An ontology or vocabulary of the platform and some reasoning activity mediate the communication of the two widgets. RDFa attributes integrated into HTML content provide the semantic knowledge which triggers the form of the target widget.

The user behavior is recorded so that frequent moves can be supported. For behavior mining the authors apply ontology-based reasoning and colored Petri nets.

[37] http://www.w3.org/TR/widgets/.

[38] Extensive demo at http://www.ahmetsoylu.com/pubshare/medes2011/, for individual system components on the web see links in the papers.

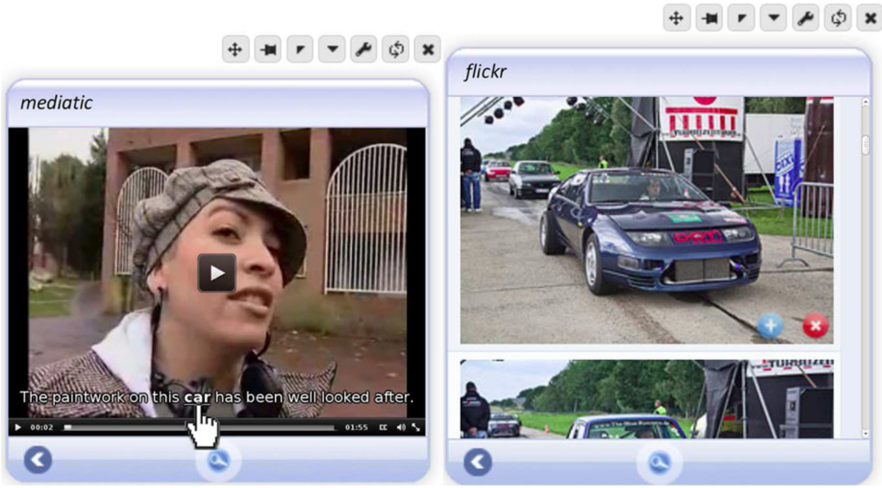



**Fig. 1.39** Widget triggering neighbor widget: tapping "car" makes display cars from Flickr (source: [207])

## 1.7 The Future of Mashups

The scientific papers count is a clear indicator for the degree of attention payed to mashups. The 'On X' sets of references under the previous section headings illustrate how unequally efforts are distributed in the mashup ecosystem. Nevertheless papers come from many areas where mashups cohabit with other types of web applications. Domains vary from genomics to bathroom appliances. Mashups empower experts in enterprises as well as consumers and smart home things or smart things elsewhere, for instance in logistics. With more IPs and more services on the web, web use is expected to keep steadily rising, engendering as side-effect a need for more organization and combined use/reuse of resources. Mashups fit that need. In short, the mashup ecosystem will go on growing and expanding into new application areas.

The references count does not explain why mashups are popular. Mashups are attractive because of their attractive properties:

- web connectivity, loose coupling
- web 2.0—style semantics, mostly annotation
- user activity and creativity
- simplicity, abstraction, delegation
- data resource reuse
- use of external services
- versatility
- accommodation to local use conditions

The listed features are by no means reserved to mashups. All web applications are entitled to integrate all sorts of data from web APIs, they can hand over processing to cloud servers, and so on. Consider a mobile app: as its local resources are limited it may be hard to avoid the use of web services, so that a mashup behavior is almost compulsory. Where sensor data are interpreted, the mashup situation is preset as well, because the sensor input must be interpreted somewhere, so that a second party is required. Those who want to impose clear frontiers between mashups and other web applications dare the impossible. Mashup behavior is promoted by mashups, but it is recommended wherever it is useful.

## References[39]


1. Abiteboul S, Greenshpan O, Milo T, Polyzotis N (2009) MatchUp: autocompletion for mashups. In: ICDE 2009, pp 1479–1482. http://www.cs.tau.ac.il/~milo/projects/mashups/papers/icde09.pdf
2. Aghaee S (2009) A quality-based framework for leveraging the process of mashup component selection. MS thesis, Univ of Gothenburg. https://gupea.ub.gu.se/bitstream/2077/21953/1/gupea_2077_21953_1.pdf
3. Aghaee S, Pautasso C (2012) EnglishMash: usability design for a natural mashup composition environment. In: ICWE 2012. https://sites.google.com/site/composableweb2012/program
4. Alani H, Hall W, O'Hara K, Shadbolt N et al (2008) Building a pragmatic semantic web. IEEE Intell Syst 23(3):61–68. http://eprints.soton.ac.uk/265787/1/alani-IEEEIS08.pdf
5. Alba A, Bhagwan V, Grace J et al (2008) Applications of voting theory to information mashups. In: IEEE Int Conf Semant Comput 2008, pp 10–17. http://www.almaden.ibm.com/cs/projects/iis/hdb/Publications/papers/VotingTheory-InfoMashups.pdf
6. Albinola M, Baresi L, Carcano M, Guinea S (2009) Mashlight: a lightweight mashup framework for everyone. In: WWW 2009. http://www.integror.net/mem2009/papers/paper9.pdf
7. Alistair A, Stincic S, Davies J et al (2010) Telecommunication mashups using RESTful services. In: IEEE Services 2010. http://oro.open.ac.uk/23089/1/, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.186.6275
8. Altinel M, Brown P, Cline S et al (2007) Damia—a data mashup fabric for intranet applications. In: VLDB 2007. http://www.vldb.org/conf/2007/papers/demo/p1370-altinel.pdf
9. Anjomshoaa A, Min Tjoa A, Hubmer A (2010) Combining and integrating advanced IT-concepts with semantic web: technology mashups architecture case study. In: ACIIDS 2010. LNAI, vol 5990, pp 13–22. http://publik.tuwien.ac.at/files/PubDat_187595.pdf
10. Ankolekar A, Krotzsch M, Tran T, Vrandecic D (2008) The two cultures: mashing up Web 2.0 and the semantic web. J Web Semant 6:70–75. http://www2007.org/papers/paper777.pdf
11. Baglietto P, Cosso F, Fornasa M et al (2010) Always-on distributed spreadsheet mashups. In: AM mashups 2009/2010. http://m.fornasa.it/baglietto2010.pdf
12. Barbagallo D, Cappiello C, Francalanci C, Matera M, Picozzi M (2012) A informing observers: quality-driven filtering and composition of Web 2.0 sources. In: EDBT-ICDT 2012, pp 1–8. http://www.edbt.org/Proceedings/2012-Berlin/papers/workshops/beweb2012/a1-barbagallo.pdf
13. Barnaghi P, Wang W, Henson C, Tayor K (2012). Semantics for the internet of things: early progress and back to the future. In: IJSWIS 2012. http://personal.ee.surrey.ac.uk/Personal/P.Barnaghi/doc/IJSWIS_SemIoT_CR_2.pdf


[39] All URLs were checked between Sept. 6 and Sept. 10, 2012 or later. In a few cases, an admission may be required.

14. Barros A, Dumas M, Bruza P (2005) The move to web service ecosystems. http://eprints.qut.edu.au/2979/1/WebServiceEcosystems.pdf
15. Batard F, Boudaoud K, Riveill M (2011) A middleware for securing mobile mashups. In: WWW 2011. http://www2011india.com/proceeding/companion/p9.pdf
16. Batty M, Crooks A, Hudson-Smith A et al (2010) Data mash-ups and the future of mapping. In: JISC technology & standards watch (TechWatch), Sept 2010. http://www.jisc.ac.uk/media/documents/techwatch/jisctsw_10_01.pdf
17. Batty M, Hudson-Smith A, Milton R, Crooks A (2010) Map mashups, Web 2.0 and the GIS revolution. Ann GIS 16(1):1–13. http://www.css.gmu.edu/andrew/pubs/aofgis_2010.pdf
18. Beemer B, Gregg D (2009) Mashups: a literature review and classification framework. Future Internet 2009(1):59–87. http://www.mdpi.com/1999-5903/1/1/59
19. Belleau F, Nolina MA, Tourigny N, Rigault P, Morissette J (2008) Bio2RDF: towards a mashup to build bioinformatics knowledge systems. J Biomed Inform 41(5):706–716. http://mashup.pubs.dbs.uni-leipzig.de/files/Belleau2008BioRDFTowardsamashuptobuildbioinformaticsknowledge.pdf
20. Benslimane D, Dustdar S, Sheth A (2008) Services mashups: the new generation of web applications. IEEE Internet Comput 12(5):13–15. http://www.infosys.tuwien.ac.at/staff/sd/papers/ServicesMashups_IC.pdf
21. Bentley F, Tollmar K, Viedma C (2012) Personal health mashups: mining significant observations from wellbeing data and context. In: CHI 2012
22. Bhattarai S, Zhao Z, Crespi N (2010) Consumer mashups: end-user perspectives and acceptance model. In: iiWAS 2010, pp 930–933. http://webspace.it-sudparis.eu/~crespi/ncpub/2010/IIWAS/MDC2010_camera_ready.pdf
23. Bishr M, Kuhn W (2007) Geospatial information bottom-up: a matter of trust and semantics. In: The European Information Society—leading the way with geo-information, pp 365–387. doi:10.1007/978-3-540-72385-1_22
24. Bizer C, Cyganiak R, Gauß T (2007) The RDF book mashup: from web APIs to a web of data. http://www.dfki.uni-kl.de/~grimnes/2007/06/SFSW07Papers/6.pdf
25. Bizer C, Lehmann J, Kobilarov G et al (2009) DBpedia—a crystallization point for the web of data. J Web Semant 7(3):154–165. http://w.websemanticsjournal.org/index.php/ps/article/viewFile/164/162
26. Bjerede M, Atkins K, Dede C (2010) Ubiquitous mobile technologies and the transformation of schooling. http://www.qualcomm.com/media/documents/files/ubiquitous-mobile-technologies-and-transformation-schooling.pdf
27. Blackstock M, Lea R (2012) WoTKit: a lightweight toolkit for the web of things. In: WOT 2012. http://www.webofthings.org/wot/2012/papers/Pervasive_WoT2012_Paper5.pdf
28. Bosch J (2009) From software product lines to software ecosystems. In: 13th international software product line conference (SPLC 2009). http://janbosch.com/Jan_Bosch/Publications_files/SPLC09-SoftwareEcosystems-Accepted.pdf
29. Boulos MNK, Resch B, Crowley DN et al (2011) Crowdsourcing, citizen sensing and sensor web technologies for public and environmental health surveillance and crisis management. Int J Health Geogr 10:67. http://www.ij-healthgeographics.com/content/10/1/67
30. Brodt A, Nicklas D (2008) The TELAR mobile mashup platform for Nokia internet tablets. In: EDBT 2008, pp 700–704. http://www.edbt.org/Proceedings/2008-Nantes/papers/p700-Brodt.pdf
31. Brodt A, Nicklas D, Sathish S, Mitschang B (2008) Context-aware mashups for mobile devices. In: WISE 2008. LNCS, vol 5175, pp 280–291. ftp://129.69.211.2/pub/library/ncstrl.ustuttgart_fi/INPROC-2008-50/INPROC-2008-50.pdf
32. Burnett MM, Scaffidi C (2011) End-user development. In: Soegaard M, Dam RF (eds) Encyclopedia of human–computer interaction. Interaction Design Foundation, Aarhus. http://www.interaction-design.org/encyclopedia/end-user_development.html
33. Butler D (2006) Mashups mix data into global service. Nature 439:6–7. http://www.antweb.org/nature_mashups_439006a1.pdf

34. Canini L, Benini S, Leonardi R (2010) Interactive video mashup based on emotional identity. In: Eusipco 2010. http://www.ing.unibs.it/~luca.canini/pub/pdf/CBL-10-EUSIPCO.pdf
35. Cao J, Rector K, Park TH, Fleming SD, Burnett M, Wiedenbeck S (2010) A debugging perspective on end-user mashup programming. In: IEEE VL/HCC 2010. ftp://cs.oregonstate.edu/pub/burnett/vlhcc10-genderDebuggingMashups.pdf
36. Cappiello C, Daniel F, Matera M (2009) A quality model for mashup components. In: ICWE 2009. https://www.fing.edu.uy/inco/cursos/caldatos/articulos/CappielloICWE09.pdf
37. Cappiello C, Daniel F, Matera M, Pautasso C (2010) Information quality in mashups. IEEE Internet Comput 14(4):14–22. http://www.floriandaniel.it/university/publicationdetails.php?publication_oid=63
38. Cappiello C, Daniel F, Koschmider A, Matera M, Picozzi M (2011) A quality model for mashups. In: Web engineering. LNCS, vol 6757, pp 137–151. http://www.aifb.kit.edu/images/6/6b/MashupQuality.pdf
39. Cappiello C, Daniel F, Matera M, Picozzi M, Weiss M (2011) Enabling end user development through mashups: requirements, abstractions and innovation toolkits. In: IS-EUD 2011, pp 9–24. http://www.floriandaniel.it/university/publicationdetails.php?publication_oid=79
40. Cappiello C, Matera M, Picozzi M, Daniel F, Fernandez A (2012) Quality-aware mashup composition: issues, techniques and tools. In: QUATIC 2012. http://www.floriandaniel.it/university/publicationdetails.php?publication_oid=109
41. Cappiello C, Matera M, Picozzi M, Caio A, Guevara MT (2012) MobiMash: end user development for mobile mashups. In: WWW 2012. http://www2012.wwwconference.org/proceedings/companion/p473.pdf
42. Cardillo D (2010) On the stage: story of a movie. PhD thesis, Turin. http://www.di.unito.it/~cardillo/PhD_thesis.pdf
43. Chaari T, Laforest F, Celentano A (2004) Design of context-aware applications based on web services. Univ Venice Tech Rep CS-2004-5. http://liris.cnrs.fr/Documents/Liris-1952.pdf
44. Challiol C, Fortier A, Gordillo SE, Rossi G (2008) Model-based concerns mashups for mobile hypermedia. In: MoMM2008, pp 170-177
45. Chang H, Lee K (2010) A quality-driven web service composition methodology for ubiquitous services. J Inf Sci Eng 26:1957–1971. http://www.iis.sinica.edu.tw/page/jise/2010/201011_02.pdf
46. Chatti MA, Jarke M, Wang Z, Specht M (2009) SMashup personal learning environments. http://ceur-ws.org/Vol-506/chatti.pdf
47. Chatti MA, Jarke M, Specht M, Schroeder U, Dahl D (2011) Model-driven mashup personal learning environments. Int J Technol Enhanc Learn 3(1):21–39
48. Chen H, Lu B, Ni Y et al (2009) Mashup by surfing a web of data APIs. In: VLDB 2009. http://www.vldb.org/pvldb/2/vldb09-1033.pdf
49. Cheung KH, Kashyap V, Luciano JS, Chen H et al (2008) Semantic mashup of biomedical data. J Biomed Inform 41:683–686. http://www.sciencedirect.com/science/article/pii/S1532046408001020
50. Cheung KH, Yip KY, Townsend JP, Scotch M (2008) HCLS 2.0/3.0: health care and life sciences data mashup using Web 2.0/3.0. J Biomed Inform 41(5):694–705. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2933816/?tool=pmcentrez
51. Chowdhury SR, Rodríguez C, Daniel F, Casati F (2010) In: Wisdom-aware computing: on the interactive recommendation of composition knowledge. In: WESOA 2010. http://www.floriandaniel.it/university/publicationdetails.php?publication_oid=74
52. Chowdhury SR, Birukou A, Daniel F, Casati F (2011) Composition patterns in data flow based mashups. In: EuroPLoP 2011. http://www.floriandaniel.it/university/publicationdetails.php?publication_oid=94
53. Chowdhury SR, Daniel F, Casati F (2011) Efficient, interactive recommendation of mashup composition knowledge. In: ICSOC 2011, pp 374–388. http://www.floriandaniel.it/university/publicationdetails.php?publication_oid=88

54. Chowdhury SR, Rodríguez C, Daniel F, Casati F (2012) Baya: assisted mashup development as a. service. In: WWW'12 companion, pp 409–412. http://www2012.org/proceedings/companion/p409.pdf
55. Chudnovskyy O, Weinhold F, Gebhardt H, Gaedke M (2011) Integration of telco services into enterprise mashup applications. http://vsr.informatik.tu-chemnitz.de/publications/2011/01/pdf
56. Chudnovskyy O, Nestler T, Gaedke M, Daniel F, Chepegin V (2012) In: End-user-oriented telco mashups: the OMELETTE approach. In: WWW 2012. http://www2012.wwwconference.org/proceedings/companion/p235.pdf, http://vsr.informatik.tu-chemnitz.de/publications/2012/04/pdf
57. Compton M, Henson C, Lefort L, Neuhaus H, Sheth A (2009) A survey of the semantic specification of sensors. In: SSN 2009. http://ceur-ws.org/Vol-522/p6.pdf
58. Cramer H, Rost M, Holmquist LE (2011) Services as materials: using mashups for research. In: UbiComp 2011. http://hcramer.files.wordpress.com/2011/07/servicesasmaterials.pdf
59. Dang MT, Dimitrova V, Djemame K (2007) Personalised mashups: opportunities and challenges. In: User modelling. LNCS, vol 4511. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.107.6722
60. Daniel F, Matera M (2008) Mashing up context-aware web applications: a component-based development approach. In: WISE 2008. LNCS, vol 5175, pp 250–263. http://www.floriandaniel.it/university/publicationdetails.php?publication_oid=33
61. Daniel F, Matera M (2009) Turning web applications into mashup components: issues, models, and solutions. In: ICWE 2009, pp 45–60. http://www.floriandaniel.it/university/publicationdetails.php?publication_oid=48
62. Daniel F, Casati F, Benatallah B, Shan MC (2009) Hosted universal composition: models. languages and infrastructure in mashArt. In: ER'09, pp 428–443. http://www.floriandaniel.it/university/publicationdetails.php?publication_oid=50
63. Daniel F, Matera M, Weiss M (2011) Next in mashup development: user-created apps on the web. In: IEEE IT Pro, Sept/Oct 2011, pp 22–29
64. De Angeli A, Battocchi A, Chowdhury SR, Rodriguez C, Daniel F, Casati F (2011) Conceptual design and evaluation of WIRE: a wisdom-aware EUD tool. Technical report DISI-11-353 1.0. http://eprints.biblio.unitn.it/2074/1/IS-EUD_2011_UNITN_Tech_Report.pdf
65. De Angeli A, Battocchi A, Chowdhury SR, Rodriguez C, Daniel F, Casati F (2011) End-user requirements for wisdom-aware EUD. In: IS-EUD 2011, pp 245–250. http://www.antonella_de_angeli.talktalk.net/files/Pdf/End-User Requirements for Wisdom-Aware EUD.pdf
66. De Keukelaere F, Bhola S, Steiner M, Chari S, Yoshihama S (2008) SMash: secure component model for cross-domain mashups on unmodified browsers. In: WWW 2008. http://researcher.watson.ibm.com/researcher/files/us-msteiner/fp332-dekeukelaere.long.pdf
67. Delin KA, Small E (2009) The sensor web: advanced technology for situational awareness. In: Handbook of science and technology for homeland security. Wiley, New York. http://www.sensorwaresystems.com/historical/resources/SensorWareSystems-Situational Awareness-Handbook.pdf
68. Di Fabbrizio G, Okken T, Wilpon JG (2009) A speech mashup framework for multimodal mobile services. In: ICMI-MLMI 2009, pp 71–78. http://www.difabbrizio.com/papers/icmi174-difabbrizio.pdf
69. Di Lorenzo G, Hacid H, Paik HJ, Benatalla B (2009) Data integration in mashups. SIGMOD Rec 38(1):59–66. http://www.sigmod.org/publications/sigmod-record/0903/p59.surveys.hacid.pdf
70. Dorn C, Schall D, Dustdar S (2009) Context-aware adaptive service mashups. In: APSCC 2009, pp 301–306. http://www.infosys.tuwien.ac.at/Staff/dorn/apscc_cameraready.pdf, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.156.4389
71. Dustdar S, Schreiner W (2005) A survey on web services composition. Int J Web Grid Serv 1(1). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.139.4827

72. Eliens A, Feldberg F, Konijn E, Compter E (2007) Mashup semantics in second life @ VU. http://www.cs.vu.nl/~eliens/media/@text-paper-mashups.pdf
73. Eliens A, Feldberg F, Konjin E, Compter E (2007) Mashups in second life @ VU. http://www.cs.vu.nl/~eliens/vip/papers/research/cv/media/paper-mashups.pdf
74. Elmeleegy H, Ivan A, Akkiraju R, Goodwin R (2008) Mashup advisor: a recommendation tool for mashup development. In: IEEE Int Conf Web Serv, pp 337–344. http://91-592-722.wiki.uml.edu/file/view/mashup_advisor.pdf
75. Ennals R, Garofalakis M (2007) MashMaker: mashups for the masses. In: SIGMOD 2007. http://www2.berkeley.intel-research.net/~rennals/pubs/sigmod035d-ennals.pdf
76. Ennals R, Gay D (2007) User-friendly functional programming for web mashups. In: ACM SIGPLAN 2007. http://ennals.org/rob/archive/intel/pubs/mashmaker_icfp.pdf
77. Falchuk B, Sinkar K, Loeb S, Dutta A (2008) Mobile contextual mashup service for IMS. In: IMSAA 2008. http://www1.cs.columbia.edu/~dutta/research/IMSAA-2008.pdf
78. Gagne D, Sabbouh M, Bennett S, Powers S (2006) Using data semantics to enable automatic composition of web services. MITRE technical paper 06_0814. http://www.mitre.org/work/tech_papers/tech_papers_06/06_0814/06_0814.pdf
79. Gamble MT, Gamble R (2008) Monoliths to mashups: increasing opportunistic assets. IEEE Softw 25(6):71–79. http://mashup.pubs.dbs.uni-leipzig.de/files/Gamble2008MonolithstoMashupsIncreasingOpportunisticAssets.pdf
80. Gaubatz P, Zdun U (2012) UML2 profile and model-driven approach for supporting system integration and adaptation of web data mashups. In: ComposableWeb 2012. http://eprints.cs.univie.ac.at/3458/1/PG_UML2_Profile_and_Model-Driven_Approach_for.pdf
81. Gebhardt H, Gaedke M, Daniel F et al (2012) From mashups to telco mashups: a survey. Internet Comput 16(3):70–76. http://www.floriandaniel.it/university/publicationdetails.php?publication_oid=104, http://vsr.informatik.tu-chemnitz.de/publications/2012/05/pdf
82. Gliet J, Krüger A, Klemm O, Schöning J (2008) Image geo-mashups: the example of an augmented reality weather camera. In: AVI 2008, pp 287–294. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.187.1359
83. Gluhak A, Bauer M, Montagut F et al (2008) Towards an architecture for a real world internet. In: Gluhak A, Bauer M, Montagut F et al (eds) Towards the future internet—a European research perspective. IOS Press, Amsterdam. http://conferences.telecom-bretagne.eu/data/asn-symposium/actes/10_keynote_gluhak.pdf
84. Göksel Canbek N, Hargis J (2011) Connecting augmented reality to higher education: mashup models in virtual worlds. In: EDULEARN11, pp 1297–1303. http://www.academia.edu/761408/Connecting_Augmented_Reality_to_Higher_Education_Mash-Up_Models_in_Virtual_Worlds
85. Goodchild MF (2007) Citizens as sensors: the world of volunteered geography. GeoJournal 69:11–221. http://www.ncgia.ucsb.edu/projects/vgi/docs/position/Goodchild_VGI2007.pdf
86. Grammel L, Storey MA (2012) An end user perspective on mashup makers. http://chiselgroup.files.wordpress.com/2012/07/grammel_mashup_makers.pdf
87. Gray AJG, Sadler J, Kit O et al (2011) A semantic sensor web for environmental decision support applications. Sensors 11(9):8855–8887. http://www.mdpi.com/1424-8220/11/9/8855
88. Gray AJG, García-Castro R, Kyzirakos K et al (2011) A semantically enabled service architecture for mashups over streaming and stored data. In: ESWC'11, vol II, pp 300–314. http://oa.upm.es/7413/1/A_SEMANTICALLY_.pdf
89. Greenshpan O, Milo T, Polyzotis N (2009) Autocompletion for mashups. In: VLDB 2009. http://www.vldb.org/pvldb/2/vldb09-95.pdf
90. Guinard D (2010) Mashing up your web-enabled home—enabling physical mashups in a web of things. In: ICWE 2010. http://de.slideshare.net/misterdom/touch-the-web2010physicalhomemashups
91. Guinard D (2010) Mashing up your web-enabled home. In: ICWE 2010. http://vs.inf.ethz.ch/publ/papers/dguinard-mashin-2010.pdf

92. Guinard D, Trifa V (2009) Towards the web of things: web mashups for embedded devices. In: MEM 2009. http://integror.net/mem2009/papers/paper4.pdf
93. Guinard D, Trifa V, Mattern F, Wilde E (2010) From the internet of things to the web of things: resource oriented architecture and best practices. In: Uckelmann D, Harrison M, Michahelles F (eds) Architecting the internet of things. Springer, Berlin, pp 97–129. http://vs.inf.ethz.ch/publ/papers/dguinard-fromth-2010.pdf
94. Guinard D, Trifa V, Wilde E (2010) Architecting a mashable open world wide web of things. http://www.vs.inf.ethz.ch/publ/papers/WoT.pdf
95. Guinard D, Floerkemeier C, Sarma S (2011) Cloud computing, REST and mashups to simplify RFID application development and deployment. In: WoT 2011. http://www.vs.inf.ethz.ch/publ/papers/guinard_epcCloud.pdf
96. Guo J, Han H, Tokuda T (2010) Towards flexible mashup of web applications based on information extraction and transfer. In: Web information systems engineering—WISE 2010. http://www.tt.cs.titech.ac.jp/~guo/WISE2010.pdf
97. Gupta S, Knoblock CA (2010) Building geospatial mashups to visualize information for crisis management. In: Int ISCRAM Conf 2010. http://www.isi.edu/integration/papers/gupta10-iscram.pdf
98. Hartmann B, Doorley S, Klemmer SR (2006) Hacking, mashing, gluing: a study of opportunistic design and development. http://hci.stanford.edu/cstr/reports/2006-14.pdf
99. Hartmann B, Wu L, Collins K, Klemmer SR (2007) Programming by a sample: rapidly prototyping web applications with d.mix. In: UIST 2007. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.75.3498, http://hci.stanford.edu/cstr/reports/2007-09.pdf
100. Hartmann B, Doorley S, Klemmer SR (2008) Hacking, mashing, gluing: understanding opportunistic design. IEEE Pervasive Comput 7(3):46–54. http://bjoern.org/papers/hartmann-pervasive2008.pdf
101. Hassan OAH, Ramaswamy, Miller JA (2010) CoMaP: a cooperative overlay-based mashup platform. In: OTM 2010. http://www.cs.uga.edu/~jam/home/theses/hassan_dissert/papers/hassan_comap.pdf
102. Hassenzahl M (2008) User experience (UX): towards an experiential perspective on product quality. In: IHM 2008. http://www.marc-hassenzahl.de/pdfs/hassenzahl-ihm08.pdf
103. Heil S, Chudnovskyy O, Gaedke M (2012) Supporting semantic interoperability in inter-widget-communication-enabled user interface mashups. http://vsr.informatik.tu-chemnitz.de/publications/2012/10/pdf
104. Hoang DD, Paik HJ, Benatallah B (2010) An analysis of spreadsheet-based services mashup. In: ADC 2010. http://unsw.academia.edu/DatHoang/Papers/660208/An_analysis_of_spreadsheet-based_services_mashup
105. Hoyer V, Stanoevska-Slabeva K (2009) Towards a reference model for Grassroots Enterprise mashup environments. http://is2.lse.ac.uk/asp/aspecis/20090189.pdf
106. Hsu IC (2011) An architecture of mobile Web 2.0 context-aware applications in ubiquitous web. J Softw 6(4):705–715. http://www.academypublisher.com/jsw/vol06/no04/jsw0604.pdf
107. Huang AFM, Yang SJH (2009) Situational mashups for ubiquitous learning. Int J Digit Learn Technol 1(3):245–265. http://ijdlt.org/download.php?dir=papers&filename=00052010-02-_ijdlt3-6_.pdf
108. Huang AFM, Yang SJH, Liaw SS (2012) A study of user's acceptance on situational mashups in situational language teaching. Br J Educ Technol 43(1):52–61
109. Huang K, Fan Y, Tan W (2012) An empirical study of ProgrammableWeb: a network analysis on a service-mashup system. In: IEEE ICWS 2012, pp 552–559. http://www.simflow.net/Publications/Papers/Year2012/HKM-1207.pdf
110. Hudson-Smith A, Crooks A (2008) The renaissance of geographic information: neogeography, gaming and second life. CASA working paper 142. http://www.bartlett.ucl.ac.uk/casa/pdf/paper142.pdf
111. Huynh DF, Miller RC, Karger DR (2007) Potluck: data mash-up tool for casual users. In: ISWC 2007. http://people.csail.mit.edu/dfhuynh/research/papers/iswc2007-potluck.pdf

112. ICT OMELETTE (2011) Initial specification of mashup description language and telco mashup architecture. Report D2.2. http://www.ict-omelette.eu/c/document_library/get_file?uuid=7743e659-57d2-4ce3-9ab9-acb5bc061aec&groupId=48739
113. Imran M, Kling F, Soi S et al (2012) ResEval mash: a mashup tool for advanced research evaluation. In: ICWE 2012, pp 291–298. WWW 2012 companion. http://www2012.wwwconference.org/proceedings/companion/p361.pdf, http://www.floriandaniel.it/university/publicationdetails.php?publication_oid=96
114. Imran M, Soi S, Kling F et al (2012) On the systematic development of domain-specific mashup tools for end users. In: ICWE 2012, pp 291–298. http://www.floriandaniel.it/university/publicationdetails.php?publication_oid=106
115. Iwazume M, Kaneiwa K, Zettsu K (2008) KC3 browser: semantic mash-up and link-free browsing. In: WWW 2008, poster paper. http://wwwconference.org/www2008/papers/pdf/p1209-iwazume.pdf
116. Jarrar M, Dikaiakos MD (2008) MashQL: a query-by-diagram topping SPARQL: towards semantic data mashups. In: ONISW 2008. http://www.jarrar.info/publications/onisw10-jarrar.pdf
117. Jarrar M, Dikaiakos MD (2009) A data mashup language for the data web. http://events.linkeddata.org/ldow2009/papers/ldow2009_paper14.pdf
118. Jie G, Bo C, Junliang C, Xiangtao L (2009) Applying recommender system based mashup to web–telecom hybrid service creation. In: IEEE GLOBECOM 2009. http://202.194.20.8/proc/GLOBECOM2009/DATA/PID967851.pdf
119. Kalou AK, Pomonis T, Koutsomitropoulos D, Papatheodorou T (2010) Intelligent book mashup: using semantic web ontologies and rules for user personalisation. In: ICSC 2010, pp 536–541. http://www.hpclab.ceid.upatras.gr/viografika/kotsomit/pubs/icsc10b.pdf
120. Kaltofen S (2010) Design and implementation of an end-user programming software system to create and deploy cross-platform mobile mashups. Linnaeus Univ Degree Project. http://lnu.diva-portal.org/smash/record.jsf?pid=diva2:369086
121. Kamilaris A, Pitsillides A (2010) Social networking of the smart home. In: PIMRC 2010. http://seacorn.cs.ucy.ac.cy/papers/files/Kamilaris_PIMRC2010.pdf
122. Kamilaris A, Trifa V, Pitsillides A (2011) HomeWeb: an application framework for web-based smart homes. In: Telecommunications (ICT) 2011, pp 134–139. http://vladtrifa.com/research/files/Kamilaris11.pdf
123. Kapitsaki GM, Prezerakos GN, Tselikas ND, Venieris TS (2009) Context-aware service engineering: a survey. J Syst Softw. doi:10.1016/j.jss.2009.02.026, http://www.ist-sms.org/Documents/J3.pdf
124. Keidl M, Kemper A (2004) Towards context-aware adaptable web services. In: WWW 2004. http://www.ambuehler.ethz.ch/CDstore/www2004/docs/2p55.pdf
125. Kenda K, Fortuna C, Fortuna B, Grobelnik M (2011) Videk: a mash-up for environmental intelligence. http://sensorlab.ijs.si/files/publications/Kenda_et_al_Videk_A_Mash_up_for_Environmental_Intelligence.pdf
126. Khadka R, Saeidi A, Jansen S et al (2011) An evaluation service framework for the management of service ecosystems. In: PACIS 2011. http://projects.business.uq.edu.au/pacis2011/papers/PACIS2011-087.pdf, http://www.staff.science.uu.nl/~khadk101/publication/Camera-Ready_version_PACIS.pdf
127. Kim JE, Boulos G, Yackovich J, Barth T, Beckel C, Mosse D (2012) Seamless integration of heterogeneous devices and access control in smart homes. In: IE 2012. http://vs.inf.ethz.ch/publ/papers/beckel-2012-seamless.pdf
128. Koskela T, Kostamo N, Kassinen O, Ohtonen J, Ylianttila M (2007) Towards context-aware mobile Web 2.0 service architecture. In: UBICOMM 2007, pp 41–48. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.131.6902
129. Kovatsch M, Mayer S, Ostermaier B (2012) Moving application logic from the firmware to the cloud: towards the thin server architecture for the internet of things. In: IMIS 2012. http://vs.inf.ethz.ch/publ/papers/mkovatsc-2012-esiot-thinserver.pdf

130. Kuka C, Busemann C, Nicklas D, Boll S (2011) Mashups for community aware sensor processing with SCAMPI. http://lgis.informatik.uni-kl.de/btw2011/fileadmin/paper/DSEP/26SCAMPI-btw-dsep.pdf
131. Lathem J, Gomadam K, Sheth A (2007) SA-REST and (S)mashups: adding semantics to restful services. In: IEEE Int Conf Semant Comput, pp 469–476. http://knoesis.wright.edu/library/download/lathemSAREST.pdf
132. Lee K (2010) Intelligent geo-web services based on hybrid-mashup using open source geospatial software. In: WebMGS 2010. http://www.isprs.org/proceedings/XXXVIII/4-W13/ID_48.pdf
133. Lee Y, Oh S, Shin C, Woo W (2009) Ubiquitous virtual reality and its key dimension. In: IWUVR 2009, pp 5–8. http://icserv.gist.ac.kr/mis/publications/data/2009/iwuvr-ylee.pdf
134. Lefort L (2009) Review of semantic augmentation techniques used in geospatial and semantic standards for legacy and opportunistic mashups. In: 5th Australian ontology workshop (AOW 2009), pp 17–26. http://crpit.com/Vol112.html
135. Le-Phuoc D (2009) SensorMasher: publishing and building mashup of sensor data. In: I-Semantics 2009. http://triplify.org/files/challenge_2009/sensormasher.pdf
136. Le-Phuoc D, Hauswirth M (2009) Linked open data in sensor data mashups. http://ir.library.nuigalway.ie/xmlui/bitstream/handle/10379/1113/p3.pdf?sequence=1, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.157.3507
137. Le-Phuoc D, Polleres A, Tummarello G, Morbidoni C (2008) DERI pipes: visual tool for wiring web data sources. http://www.cs.vu.nl/~pmika/swc-2008/DERI Pipes-DERI Pipes.pdf
138. Le-Phuoc D, Polleres A, Hauswirth M, Tummarello G, Morbidoni C (2009) Rapid prototyping of semantic mash-ups through semantic web pipes. In: WWW 2009, pp 581–590. http://www2009.eprints.org/59/1/p581.pdf
139. Le-Phuoc D, Parreira JX, Hausenblas M, Han J, Hauswirth M (2010) Live linked open sensor database. In: I-Semantics 2010. http://blog.semantic-web.at/wp-content/uploads/2010/09/a46_LePhuoc.pdf
140. Li S, Gong J (2008) Mashup: a new way of providing web mappings/GIS services. In: ISPRS 2008. http://www.isprs.org/proceedings/XXXVII/congress/4_pdf/114.pdf
141. Lin H, Batty M (2009) Virtual geographic environments: a primer. In: Lin H, Batty M (eds) Virtual geographic environments. Science Press, Beijing. http://downloads2.esri.com/ESRIpress/images/206/126840_VGE_samplechapter.pdf
142. Lin J, Wong J, Nichols J, Cypher A, Lau TA (2009) End-user programming of mashups with Vegemite. In: IUI 2009, pp 97–106. http://luci.ics.uci.edu/websiteContent/weAreLuci/biographies/faculty/djp3/LocalCopy/p97-lin.pdf
143. Liu D, Li N, Pedrinaci C et al (2011) An approach to construct dynamic service mashups using lightweight semantics. In: ComposableWeb 2011. http://people.kmi.open.ac.uk/ning/papers/cw2011.pdf
144. López-de-Ipiña D, Vázquez I, Abaitua J (2007) A context-aware mobile mash-up for ubiquitous web. http://paginaspersonales.deusto.es/dipina/publications/SentientGraffitiIE07.pdf
145. Lorey J, Naumann F, Forchhammer B, Mascher A, Retzlaff P, ZamaniFarahani A (2011) Black Swan: augmenting statistics with event data. In: CIKM 2011. http://blackswanevents.org/data/BlackSwan.pdf
146. Maaradji A, Hacid H, Daigremont J, Crespi N (2010) Towards a social network based approach for services composition. In: IEEE ICC 2010. http://webspace.it-sudparis.eu/~crespi/ncpub/2010/ICCAR/PID1017804.pdf
147. Maaradji A, Hacid H, Skraba R, Lateef A, Daigremont J, Crespi N (2011) Social-based web services discovery and composition for step-by-step mashup completion. In: ICWS 2011, pp 700–701. http://servicearchitecture.wp.it-sudparis.eu/files/2012/04/PID1842125.pdf
148. Mäkelä E, Viljanen K, Alm O et al (2007) Enabling the semantic web with ready-to-use mash-up components. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.100.2468
149. Maleshkova M, Pedrinaci C, Domingue J (2009) Supporting the creation of semantic RESTful service descriptions. In: ISWC 2009. http://oro.open.ac.uk/23106/1/paper6.pdf

150. Malki A, Benslimane SM (2012) Building semantic mashup. In: ICWIT 2012, pp 40–49. http://ceur-ws.org/Vol-867/Paper5.pdf
151. Maximilien EM, Wilkinson H, Desai N, Tai S (2007) A domain-specific language for web APIs and services mashups. In: ICSOC 2007. LNCS, vol 4749, pp 13–26. http://www.cs.uga.edu/~kochut/Teaching/8350/Papers/WS/WS_API_Mashups.pdf
152. Maximilien EM, Ranabahu A, Gomadam K (2008) An online platform for web APIs and service mashups. IEEE Internet Comput 12(5):32–43. http://knoesis.wright.edu/library/download/MRG-sharable_code-IC08.pdf
153. Mayer S (2010) Deployment and mashup creation support for smart things. MS thesis, ETH Zurich. http://www.vladtrifa.com/research/files/Mayer2010.pdf
154. Mayer S, Guinard D (2011) An extensible discovery service for smart things. In: WoT 2011. http://vs.inf.ethz.ch/publ/papers/mayer-extensiblediscovery-2011.pdf
155. Mayer S, Karam DS (2012) A computational space for the web of things. In: 3rd international workshop on the web of things (WoT 2012). http://vs.inf.ethz.ch/publ/papers/mayer-computationalmarketplace-2012.pdf
156. Mayer S, Guinard D, Trifa V (2010) Facilitating the integration and interaction of real-world services for the web of things. http://www.webofthings.org/urban-iot/2010/pdfs/Mayer.pdf
157. McIlraith SA, Son TC, Zeng H (2001) Semantic web services. IEEE Intell Syst 16(2):46–53. http://o2k.stanford.edu/people/sam/ieee01.pdf
158. Mehandjiev N, Lécué F, Wajid U, Namoun A (2010) Assisted service composition for end users. In: IEEE European Conf Web Serv. http://130.88.6.246/serviceDesign/files/Papers/AssistedServiceCompositionforEndUsers.pdf
159. Miles A (2009) News from the FlyWeb project (Milestone FM3). http://lists.w3.org/Archives/Public/public-lod/2009May/0052.html
160. Morbidoni C, Polleres A, Tummarello G, Le-Phuoc D (2007) Semantic web pipes. DERI technical report 2007-11-07. http://sw.deri.ie/fileadmin/documents/DERI-TR-2007-11-07.pdf
161. Mtsweni JS, Ngassam EK (2009) Seamless access to data and e-services via mobile phones: a Web2.0 approach for SMMEs in South Africa. In: WWW 2009. http://www.zaw3.co.za/index.php/ZA-WWW/2009/paper/viewFile/53/37
162. Nakano T, Murofushi S, Goto M, Morishima S (2011) DandeReproducer: an automatic mashup music video generation system by reusing dance video clips on the web. In: SMC 2011. http://staff.aist.go.jp/m.goto/PAPER/SMC2011nakano.pdf
163. Neely S, Stabeler M, Nixon P (2008) SensorMash: exploring system fidelity through sensor mashup. Pervasive 2008:83–86. http://www.mattstabeler.co.uk/publications/papers/SensorMash.pdf, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.140.3685
164. Nestler T, Feldmann M, Preussner A, Schill A (2009) Service composition at the presentation layer using web service annotations. In: ComposableWeb 2009. http://ceur-ws.org/Vol-470/CW09Proc.pdf#page=71:63-68
165. Neto JB, Hirata CM, Marques HC (2010) An architecture based on SOA, RESTful and mashup for C2 mobile applications. In: ICCRTS 2012. http://www.dodccrp.org/events/17th_iccrts_2012/post_conference/papers/094.pdf
166. Ngu AHH, Carlson MP, Sheng QZ, Paik H (2010) Semantic-based mashup of composite applications. IEEE Trans Serv Comput 3(1):2–15. http://alphasoftz.in/ws/paper.pdf
167. Outtagarts A, Martinot O (2009) iSSEE: IMS sensors search engine enabler for sensors mashups convergent application. Int J Comput Sci Issues 6(1). http://arxiv.org/pdf/0912.1805.pdf, http://130.203.133.150/viewdoc/similar;jsessionid=6F8EC9EE4ABE8FC79121C126F29724DA?doi=10.1.1.214.9584&type=ab
168. Passant A (2010) dbrec—music recommendations using DBpedia. In: ISWC 2010. http://iswc2010.semanticweb.org/pdf/431.pdf
169. Passant A (2010) Measuring semantic distance on linking data and using it for resources recommendations. http://swl.slis.indiana.edu/files/ldrec.pdf

170. Patel A, Na L, Latih R, Wills C, Shukur Z, Mulla R (2010) A study of mashup as a software application development technique with examples from an end-user programming perspective. J Comput Sci 6(12):1406–1415
171. Pathak J, Basu S, Honavar V (2007) Assembling composite web services from autonomous components. In: Emerging artificial intelligence applications in computer engineering: real word AI systems with applications in eHealth, HCI, information retrieval and pervasive technologies, pp 394–405. http://www.cs.iastate.edu/~honavar/Papers/Pathak-Bookchapter.pdf
172. Pedrinaci C, Domingue J (2010) Toward the next wave of services: linked services for the web of data. J Univers Comput Sci 16(13):1694–1719. http://www.jucs.org/jucs_16_13/toward_the_next_wave/jucs_16_13_1694_1719_pedrinaci.pdf, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.186.5572, http://oro.open.ac.uk/23073/1/services-for-the-web-of-data.pdf
173. Pedrinaci C, Domingue J, Sheth A (2010) Semantic web services. In: Domingue J, Fensel D, Hendler J (eds) Handbook of semantic web technologies, vol II—Semantic web applications. Springer, Berlin. http://open.academia.edu/CarlosPedrinaci/Papers/1016343/Semantic_web_services
174. Pedrinaci C, Liu D, Maleshkova M et al (2010) iServe: a linked services publishing platform. In: ORES 2010. http://people.kmi.open.ac.uk/carlos/wp-content/uploads/downloads/2010/09/iserve-ORES2010.pdf
175. Peng Z, Chen H, Rao J, Liu Y, Wang L, Chen J (2010) Semantic-based mobile mashup platform. In: ISWC 2010. http://ceur-ws.org/Vol-658/paper479.pdf, http://iswc2010.semanticweb.org/pdf/479.pdf
176. Picozzi M, Rodolfi M, Cappiello C, Matera M (2010) Quality-based recommendations for mashup composition. In: Current trends in web engineering. LNCS, vol 6385, pp 360–371. doi:10.1007/978-3-642-16985-4_32, http://mashart.org/composableweb2010/papers/paper3.pdf
177. Pietschmann S (2009) A model-driven development process and runtime platform for adaptive composite web applications. Int J Adv Internet Technol 2(4):277–288. http://thinkmind.org/index.php?view=article&articleid=inttech_v2_n4_2009_2
178. Pietschmann S, Mitschick A, Winkler R, Meissner K (2008) CROCO: ontology-based, cross-application context management. In: SMAP 2008, pp 88–93. http://pietschmann.cc/uploads/publications/Pietschmann.2008.SMAP.pdf
179. Pietschmann S, Nestler T, Daniel F (2010) Application composition at the presentation layer: alternatives and open issues. In: iiWAS 2010, pp 461–468. http://www.floriandaniel.it/university/publicationdetails.php?publication_oid=71
180. Pietschmann S, Tietz V, Reimann J et al (2010) A metamodel for context-aware component-based mashup applications. In: iiWAS 2010, pp 413–420. http://pietschmann.cc/uploads/publications/Pietschmann.2010.iiWAS.Metamodel.pdf
181. Pietschmann S, Radeck C, Meissner K (2011) Facilitating context-awareness in composite mashup applications. In: ADAPTIVE 2011. http://www.thinkmind.org/index.php?view=article&articleid=adaptive_2011_1_10_50008
182. Ponce-Medellín R, Gonzalez-Serna G, Vargas R, Ruiz L (2009) Technology integration around the geographic information: a state of the art. Int J Comput Sci Issues 5:17–26. http://arxiv.org/pdf/0911.0909.pdf
183. Prutsachainimmit K, Chaisatien P, Tokuda T (2012) A mashup construction approach for cooperation of mobile devices. In: ComposableWeb 2012. https://sites.google.com/site/composableweb2012/program
184. Radeck C, Lorz A, Blichmann G, Meissner K (2012) Hybrid recommendation of composition knowledge for end user development of mashups. In: ICIW 2012. http://www.thinkmind.org/index.php?view=article&articleid=iciw_2012_2_10_20180iciw_2012_2_10_20180
185. Raza M, Hussain FK, Chang E (2008) A methodology for quality-based mashup of data sources. In: iiWAS 2008, pp 528–533. http://mashup.pubs.dbs.uni-leipzig.de/files/Raza2008Amethodologyforqualitybasedmashupofdatasources.pdf

186. Riabov AV, Bouillet E, Feblowitz MD, Liu Z, Ranganathan A (2008) Wishful search: interactive composition of data mashups. In: WWW 2008. http://wwwconference.org/www2008/papers/pdf/p775-riabovA.pdf, http://www2008.org/papers/pdf/p775-riabovA.pdf
187. Riedl C, Böhmann T, Leimeister JM, Krcmar H (2011) A framework for analysis service ecosystem capabilities to innovate. http://www.theseus.joint-research.org/wp-content/uploads/2011/07/RiedlEtAl2009b1.pdf
188. Rizzotti S, Burkhart H (2010) useKit—lightweight mashups for the personalized web. In: WWW 2010. http://integror.net/mem2010/papers/paper98.pdf
189. Rümpel A, Baumgärtel K, Meissner K (2010) Platform adaptation of mashup UI components. In: ADAPTIVE 2010, pp 64–169. http://www.thinkmind.org/index.php?view=article&articleid=adaptive_2010_7_20_40111
190. Sabbouh M, Higginson J, Semy S, Gagne D (2007) Web mashup scripting language. In: WWW 2007. http://www2007.org/posters/poster972.pdf
191. Sabbouh M, Higginson J, Wan C, Semy S, Gagne D (2007) The web mashup scripting language profile. http://ceur-ws.org/Vol-248/paper1.pdf
192. Sabbouh M, Higginson J, Wan C, Semy S, Gagne D (2007) The web mashup scripting language profile. MITRE technical paper 07_0393. http://www.mitre.org/work/tech_papers/tech_papers_07/07_0393/07_0393.pdf
193. Sahoo SS, Bodenreider O, Rutter JL, Skinner KJ, Sheth AP (2008) An ontology-driven semantic mash-up of gene and biological pathway information: application to the domain of nicotine dependence. J Biomed Inform 41(5):752–765. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2766186/?tool=pubmed
194. Salim FD (2011) The challenges of designing effective persuasive technology on the road: towards adaptive mobile mashups. In: WAINA 2011, pp 7–11. http://www.nicta.com.au/__data/assets/pdf_file/0005/21596/salim.pdf
195. Salminen A, Kallio J (2010) Technical report: mobile multimedia mashups. https://www.cloudsoftwareprogram.org/results/delivrables-and-other-reports/i/27547/1941/d1-3-1-technical-report-multimedia-mashups
196. Salminen A, Mikkonen T (2012) Mashups—software ecosystems for the web era. In: IWSECO 2012, pp 18–32. http://ceur-ws.org/Vol-879/paper2.pdf
197. Sheth A (2009) Citizen sensing, social signals, and enriching human experience. IEEE Internet Comput 13(4):87–92. http://knoesis.wright.edu/library/publications/S09-citizen-sensor-IC.pdf
198. Sheth A, Verma K, Gomadam K (2006) Semantics to energize the full services spectrum: ontological approach to better exploit services at technical and business levels. Commun ACM 49(7):55–61. http://knoesis.wright.edu/library/download/SS8.pdf
199. Sheth A, Henson C, Sahoo SS (2008) Semantic sensor web. IEEE Internet Comput 12(4):78–83. http://knoesis.wright.edu/library/publications/SHS08-IC-Column-SSW.pdf
200. Sheth AP, Gomadam K, Lathem J (2007) SA-REST: semantically interoperable and easier-to-use services and mashups. IEEE Internet Comput 11(6):91–94. http://knoesis.wright.edu/library/download/SGL07-IEEEIC-SAREST-Dec07.pdf
201. Shih F, Kagal L (2009) Policy-aware pipes: support accountability in mashup service of linked data. In: ISWC 2009. http://dig.csail.mit.edu/2009/Papers/ISWC/policy-aware-pipe/paper.pdf
202. Shrestha P (2009) Automatic mashup generation of multiple-camera videos. PhD thesis, Eindhoven. http://alexandria.tue.nl/extra2/200613048.pdf
203. Shrestha P, De With PHN, Weda H, Barbieri M, Aarts EHL (2010) Automatic mashup generation from multiple-camera concert recordings. In: MM 2010, pp 541–550. http://www.dcg.ethz.ch/lectures/fs11/seminar/paper/samuel-2-2.pdf
204. Soon CJ, Roe P, Tjondronegoro DW (2008) An approach to mobile collaborative mapping. In: SAC 2008, pp 1929–1934. http://eprints.qut.edu.au/15185/1/15185.pdf
205. Soylu A, Wild F, Mödritscher F, De Causmaecker P (2010) Toward developing a semantic mash-up personal and pervasive learning environment: SMupple. In: MUPPLE 2010. http://www.moedritscher.com/papers/paper_soylu_et_al_smuppleapproach_2010.pdf

206. Soylu A, Wild F, Mödritscher F et al (2011) Mashups and widget orchestration. In: MEDES 2011. http://www.ahmetsoylu.com/wp-content/uploads/2011/05/50.pdf
207. Soylu A, Mödritscher F, Wild F et al (2012) Mashups by orchestration and widget-based personal environments: key challenges, solution strategies, and an application. Program 46(3):155–200. https://lirias.kuleuven.be/bitstream/123456789/345188/2/Program_Soylu_et_al_2012.pdf, http://www.ahmetsoylu.com/wp-content/uploads/2011/05/PhDThesis_Soylu_2012.pdf
208. Spivack N (2007) Making sense of the semantic web. http://74.43.177.87/digitalnow/upload/Nova-Spivack-Semantic-Web.pdf
209. Taivalsaari A (2009) Mashware: the future of web applications. Sun Microsystems report SMLI TR-2009-181. http://labs.oracle.com/techrep/2009/smli_tr-2009-181.pdf
210. Tatemura J, Chen S, Liao F, Po O, Candan KS, Agrawal D (2008) UQBE: uncertain query by example for web service mashup. In: SIGMOD 2008. http://www.public.asu.edu/~candan/papers/de694-tatemura.pdf
211. Thébault P, Decotter D, Richir S (2011) Towards the design of intelligible object-based applications for the web of things. In: WoT 2011. http://www.webofthings.org/wot/2011/papers/Pervasive2011.Workshop09.Paper10.pdf
212. Tietz V, Rümpel A, Liebing C, Meissner K (2012) Towards requirements engineering for mashups: state of the art and research challenges. In: ICIW 2012. http://www.thinkmind.org/index.php?view=article&articleid=iciw_2012_4_50_20164
213. Tollmar K, Bentley F, Viedma C (2012) Mobile health mashups: making sense of multiple streams of wellbeing and contextual data for presentation on a mobile device. In: Pervasive Health 2012. http://web.mit.edu/bentley/www/papers/mashupsPH.pdf
214. Treiber M, Kritikos K, Schall D, Plexousakis D, Dustdar S (2010) Modeling context-aware and socially-enriched mashups. In: Mashups 2009/2010. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.156.6531
215. Truong HL, Dustdar S (2010) Context coupling techniques for context-aware web service systems—an overview. http://www.infosys.tuwien.ac.at/Staff/sd/papers/Buchbeitrag Linh revised paper.pdf, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.189.2089
216. Tuchinda R, Szekely P, Knoblock CA (2008) Building mashups by example. In: IUI 2008, pp 139–148. http://www.isi.edu/integration/papers/tuchinda08-iui.pdf
217. Tuulos VH, Scheible J, Nyholm H (2007) Combining web, mobile phones and public displays in large-scale: Manhattan story mashup. In: PERVASIVE 2007, pp 37–54. http://www.mlab.uiah.fi/~jscheib/myweb/msm-pervasive07.pdf
218. Udell S (2008) Beginning Google maps mashups with mapplets, KML, and GeoRSS. Apress, New York
219. Väänänen-Vainio-Mattila K, Wäljas M (2011) Towards user-centered mashups: exploring user needs for composite web services. In: CHI 2011. http://www.cs.tut.fi/~kaisavvm/CompositeUX_WIP_final_update_KVVM_060311.pdf
220. Vallejos J, Huang J, Costanza P, De Meuter W, D'Hondt T (2009) A programming language approach for context-aware mashups. http://soft.vub.ac.be/lambic/files/contextawaremashups.pdf
221. Vancea A, Grossniklaus M, Norrie MC (2008) Database-driven web mashups. In: ICWE 2008. http://pdx.academia.edu/MichaelGrossniklaus/Papers/244818/Database-Driven_Web_Mashups
222. Verma K, Sheth A (2007) Semantically annotating a web service. IEEE Internet Comput 11(2):83–85. http://knoesis.wright.edu/library/download/VS07-IC-Sheth-Column-SAWSDL-Published.pdf
223. Vermeulen J, Luyten K, Coninx K (2007) Tangible mashups: exploiting links between the physical and virtual world. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.86.4244
224. Viedma C, Tollmar K (2010) Mobile web mashups: the long tail of mobile applications. http://www.ict.kth.se/courses/ID2216/wiki/lib/exe/fetch.php?media=mobilewebmashups.pdf

225. Voulodimos AS, Patrikakis CZ (2008) Using personalized mashups for mobile location based services. In: IWCMC 2008, pp 321–325. http://202.194.20.8/proc/IWCMC2008/DATA/01-08-05.pdf
226. Waltsgott J, Meissner K (2012) Towards mobile energy-adaptive rich internet applications. In: ICIW 2012. http://www.thinkmind.org/index.php?view=article&articleid=iciw_2012_6_40_20075
227. Wang G, Yang S, Han Y (2009) Mashroom: end-user mashup programming using nested tables. In: WWW 2009. http://www2009.eprints.org/87/1/p861.pdf
228. Weber N, Nelkner T, Schoefegger K, Lindstaedt SN (2010) SIMPLE—a social interactive mashup PLE. In: MUPPLE 2009. http://know-center.tugraz.at/download_extern/papers/mupple10_mature_final.pdf
229. Weiss M, Gangadharan GR (2010) Modeling the mashup ecosystem: structure and growth. R & D Manag 40(1):40–49. http://www.sce.carleton.ca/faculty/weiss/papers/2010/weiss-rnd-2010.pdf
230. Weiss M, Sari S (2010) Evolution of the mashup ecosystem by copying. In: ACM Mashups 2009/2010. http://www.sce.carleton.ca/faculty/weiss/papers/2010/weiss-mashups-2010-draft.pdf
231. Weiss M, Sari S (2011) Diversity of the mashup ecosystem. In: International conference on web information systems (WEBIST). http://www.sce.carleton.ca/faculty/weiss/papers/2011/sari-diversity-2011-draft.pdf
232. Wilde E (2006) Knowledge organization mashups. TIK report 245, ETH Zurich. http://dret.net/netdret/docs/wilde-tikrep245.pdf
233. Wilson S (2012) Design challenges for user-interface mashups: user control and usability in inter-widget communications. http://scottbw.wordpress.com/2012/03/07/design-challenges-for-user-interface-mashups-user-control-and-usability-in-inter-widget-communications/
234. Wilson S, Daniel F, Jugel U, Soi S (2011) Orchestrated user interface mashups using W3C widgets. In: ICWE 2011. http://www.floriandaniel.it/university/publicationdetails.php?publication_oid=85
235. Wong J, Hong JI (2007) Making mashups with Marmite: towards end-user programming for the web. In: IEEE VL/HCC 2007, pp 270–271. http://repository.cmu.edu/cgi/viewcontent.cgi?article=1063&context=hcii
236. Yamada I, Yamaki W, Nakajima H, Takefuji Y (2010) Ousia Weaver: a tool for creating and publishing mashups as impressive web pages. In: WWW 2010. http://integror.net/mem2010/papers/paper14.pdf
237. Yoon H, Woo W (2009) CAMAR mashup: empowering end-user participation in U-VR environment. In: ISUVR 2009, pp 33–36. http://icserv.gist.ac.kr/mis/publications/data/2009/CAMAR_Mashup_isuvr.pdf
238. Yoon H, Woo W (2011) Concept and applications of in-situ AR mashup content. In: SCI 2011. http://icserv.gist.ac.kr/mis/publications/data/2011/p25.pdf
239. Yu J, Benatallah B, Casati F, Daniel F (2008) Understanding mashup development and its differences with traditional integration. Internet Comput 12(5):44–52. http://mashart.org/ER09/YuIEEEC08.pdf
240. Yu S, Woodard CJ (2008) Innovation in the programmable web: characterizing the mashup ecosystem. In: Feuerlicht G, Lamersdorf W (eds) ICSOC 2008. LNCS, vol 5472. Springer, Heidelberg, pp 136–147. http://mashup.pubs.dbs.uni-leipzig.de/files/fulltext.pdf
241. Yue KB (2010) Experience on mashup development with end user programming environment. J Inf Sci Eng 21(1):111–119. http://prtl.uhcl.edu/portal/page/portal/SCE/COMPUTING_MATHMATICS_DIV/CM_Documents/YahooPipe_JISE_Paper_Published.pdf
242. Zahariadis T, Daras P, Bouwen J et al (2010) Towards a content-centric internet. In: Tselentis G et al (eds) Towards the future internet. IOS Press, Amsterdam. http://www.coast-fp7.eu/public/Zahariadis_FCCI_Camera_Ready.pdf
243. Zang N (2008) Mashups for the web-active end user. MS thesis, Pennsylvania State University. https://etda.libraries.psu.edu/paper/9241/4574

244. Zang N (2009) Mashups on the web: end user programming opportunities and challenges. http://ecs.victoria.ac.nz/twiki/pub/Events/PLATEAU/2009Program/plateau09-zang.pdf
245. Zang N, Rosson MB (2008) What's in a mashup? And why? Studying the perceptions of web-active end users. In: IEEE VL/HCC 2008. http://webremix.org/readings/zang.pdf
246. Zang N, Rosson MB, Nasser V (2008) Mashups: Who? What? Why? In: CHI 2008, pp 3171–3176
247. Zeng L, Benatallah B, Dumas M, Kalagnanam J, Sheng QZ (2003) Quality driven web services composition. In: WWW 2003. http://eprints.qut.edu.au/1778/1/p358-zeng.pdf
248. Zeng Z, Zhang X (2010) Research on mobile e-commerce information search approach based on mashup technology. Int J Bus Manag 5(5):89–96. http://ccsenet.org/journal/index.php/ijbm/article/view/5930
249. Zhang H, Zhao Z, Sivasothy S, Huang K, Crespi N (2010) Quality-assured and sociality-enriched multimedia mobile mashup. EURASIP J Wirel Commun Netw. doi:10.1155/2010/721312, http://jwcn.eurasipjournals.com/content/pdf/1687-1499-2010-721312.pdf
250. Zhang R, Arpinar IB, Aleman-Meza B (2003) Automatic composition of semantic web services. In: ICWS 2003, pp 38–41. http://www.cs.uga.edu/~budak/papers/composition_short_icws.pdf
251. Zhao Z (2012) User-generated service: a user-centered design approach in web mashups through social experience. PhD thesis, SudParis. http://tel.archives-ouvertes.fr/docs/00/71/94/21/PDF/Thesis_ZhenzhenZhao.pdf
252. Zhao Z, Bhattarai S, Liu J, Crespi N (2011) Mashup services to daily activities—end-user perspective in designing a consumer mashups. In: iiWAS 2011. http://servicearchitecture.wp.it-sudparis.eu/files/2012/04/67_iiWAS2011_Full_Zhao.pdf
253. Zhao Z, Bhattarai S, Crespi N (2012) An event-based functionality integration framework. http://servicearchitecture.wp.it-sudparis.eu/files/2012/04/PID1841337.pdf
254. Zhovtobryukh D (2006) Context-aware web service composition. PhD thesis, Jyväskylä. https://jyx.jyu.fi/dspace/bitstream/handle/123456789/13238/951392727X.pdf?sequence=1
255. Zou J, Pavlovski CJ (2007) Towards accountable enterprise mashup services. In: ICEBE 2007, pp 205–212. http://mashup.pubs.dbs.uni-leipzig.de/files/30030205.pdf

# Chapter 2
# Mashups Live on Standards

**Brigitte Endres-Niggemeyer**

**Abstract** Mashups integrate web services from pre-existing resources, often submitting them to an unforeseen secondary use. This can only succeed if standards, specifications or other types of regulation are maintained. A good understanding of standards helps mashups and their developers along. The chapter illustrates how web standards are organized and takes its readers through a parade of selected standards. Since standardization is a moderately organized environment, standards and their relatives (specifications, APIs, guidelines) are grouped from a users' point of view, depending on their usages in mashups: web communication, data organization, content types, web queries, and interfaces. As one may assume standards thrive in cultured web land and are poor or lacking in widely uncharted areas. Application examples are included from time to time. The standards review remains open-ended, but it draws the attention of its readers/users to an important and often neglected issue of their work. Some readers may find hints that directly help during mashup creation. All discussed entities come with their URL references (checked on 19/20-04-2012, a few ones on 20/23-09-2012).

## 2.1 Mashups and Their Underpinnings

Mashups live on standards. By definition they integrate existing web resources and draw new services out of them. We expect the resources to comprise things like data, procedures for their combination, and user interfaces that present the mashup output. Clearly, the data sources must conform to their specifications, web services and their APIs as well, web communication is a must, and the user interfaces of mashups have to comply with good practice. Semantic mashups may require specific formats for instance for their metadata and rules. In general mashups have to conform to the local standards and specifications of their application domains. There are many of them.

B. Endres-Niggemeyer (✉)
Heidegrün 36, 30179 Hanover, Germany
e-mail: brigitteen@googlemail.com

B. Endres-Niggemeyer (ed.), *Semantic Mashups*,

DOI 10.1007/978-3-642-36403-7_2, 

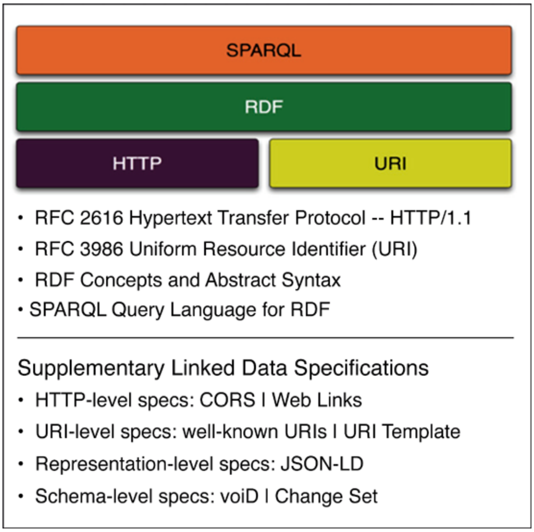



**Fig. 2.1** Specification underpinnings of linked data (shortened)

Let linked data authors explain what this means in practice. They display their specification underpinnings in a drawing that appears in Fig. 2.1 in an abridged version.[1] As a mashup combines several resources, even more standards from the application areas will walk in! In short, one cannot ignore the cohorts of applicable standards, specifications and the like when dealing with mashups.

Naive minds may expect the standards scene to be thoroughly organized, but beware: you may find standards in the wild instead. The cartoon[2] in Fig. 2.2 makes fun of how standards propagate. As a consequence of opportunistic growth, standard bodies cover their own areas of interest and may even rival each other. This makes life difficult for developers. In other words, in spite of often being blamed as under-regulated, the Internet appears as a highly standardized environment, albeit inhabited by rampant standards.

For dealing with web standards and their relatives, keeping up with some confusion factor is obligatory. The confusion resides inside the area. A description can map it, but this will not change the way agents behave in the field. As a continuous empirical structure is missing, readers of this chapter are the more entitled to ask what they care about when being confronted with regulations in the web, in particular with those related to mashups.

Regulation power clearly is a point. Standards and their kinship may regulate behavior to different degrees. Guidelines tend to be soft. They often propose best practices. Offenses may be worse or better than the guideline demands. In other cases a standard or protocol transgression will malfunction.

Less important is how a regulation is named. Depending on their emitters and their purpose, standards are often labeled specifications, protocols or recommen-

[1] Full version at http://linkeddata-specs.info/.

[2] From http://imgs.xkcd.com/comics/standards.png.

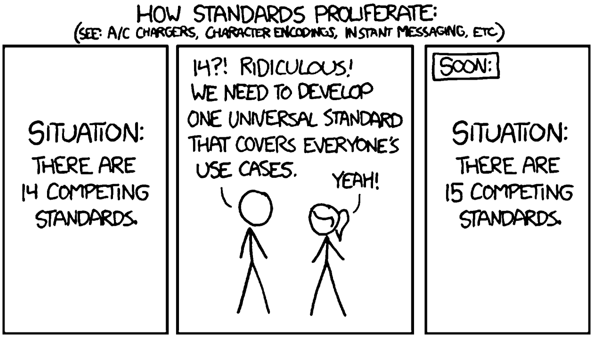



**Fig. 2.2** How standards proliferate

dations. Internet Engineering Task Force (IETF)[3] standards are called RFCs ("Requests for Comment"). Famous examples are RFC 791, the Internet Protocol (IP) and RFC 1738, the URL.[4] The long RFC lists also illustrate the high number of Internet standards. Nota bene that IETF is not the only standard body on the net.

Although standards or specifications are assumed to be set top-down by an authoritative body, web reality knows authority-free bottom-up standardization, too. Companies introduce de facto standards with their product or service APIs, or users massively prefer one practice so that crowds set standards. Pragmatic readers with mashups in mind will accept de facto and de jure standards and apply them.

For use in this chapter, the terminology can be simplified: the term standard is used as long as no discrimination from different types of standard-like regulations is required. Less-than-clear facts will inevitably blur terms, so that retracting towards everyday usage like in "be up to standard" is inevitable. WaSP, the Web Standards Project[5] fights for standards in the W3C-regulated functional core of the web. Why should the degree of organization and clear-cut definition rise with distance from the center?

Probably ISO, the International Standards Organization[6] is the most prominent provider of standards in a wide range of applications: ISO 26000 deals with social responsibility, ISO/IEC DIS 17826 with CDMI, which is a cloud data management interface, and ISO 11760:2005 specifies a classification system for coals. According to ISO/IEC Guide 2:2004 a standard is

> a document, established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context [19].

Readers will notice the comprehensive definition of ISO standards. It includes all sorts of regulations for common reuse. This book chapter follows it.

---

[3] http://www.ietf.org.

[4] This and more on http://www.faqs.org/rfcs/.

[5] http://www.webstandards.org/.

[6] http://www.iso.org/iso/home.htm.

ISO standards are developed in working groups of experts. Draft standards go through an approval procedure. If a draft is accepted by two-thirds of its developers and three-fourth of all ISO members, it is recognized as a standard and published. After some time of use, standards often need a revision. HTML may serve as an example. Revisions run from version 1.0 in 1991 to version 5.0 of today.[7]

Other standard emitters follow more or less the same procedure of standard development and agreement by vote/certification. However, most standardizing organizations are less global than ISO, restricting their domain to a country like DIN[8] in Germany, a discipline like ITU[9]—the International Telecommunication Union—or a specific environment as W3C[10]—the World Wide Web Consortium—does.

Standard bodies may cooperate, but this is far from being the rule. A special mention is, however, due to working groups of different standard bodies who cooperate. WebCGM[11] exemplifies a recommendation that W3C and OASIS publish together.

Many bodies offer their standards ("open" standards) for free. This is for instance true for ECMA,[12] OSGi,[13] OASIS[14] and for W3C. Many Apache implementations distribute open standards, too.[15] Open standards may be connected with the open source software movement.

Open (public) standards are preferred.[16] Proprietary standards are less accessible when their content is not known, and if patents or other regulations impose costs. ISO standard documents and those of other providers such as IEEE[17] or US ANSI[18] are sold. Their prices cover standard development costs. That they are paid emphasizes the manifest practical value of standards—agreed specifications—in many technical domains. Standards reduce costs.

Mixed-mode open and licensed standards exist as well. MP3[19] is a common example.

Standards can support mashups straightforwardly, but just as frequent is a mediated use via an API (Application Programming Interface) that applies standards.

On the value of web standards, one can rephrase the argumentation of the WaSP project cited above: Where specifications and standards reign the web, communica-

---

[7] http://en.wikipedia.org/wiki/HTML, brief overview at http://www.yourhtmlsource.com/starthere/historyofhtml.html.

[8] http://www.din.de/.

[9] http://www.itu.int/en/Pages/default.aspx.

[10] http://www.w3.org/.

[11] http://www.w3.org/TR/webcgm/.

[12] http://www.ecma-international.org/default.htm.

[13] http://www.osgi.org/About/HomePage.

[14] http://www.oasis-open.org/.

[15] See Apache list on http://projects.apache.org/indexes/standards.html.

[16] http://www.openformats.org/main.

[17] See standards store at http://www.techstreet.com/ieeegate.html.

[18] http://www.ansi.org/.

[19] http://en.wikipedia.org/wiki/MP3.

tion runs better. Since communication is a main feature of the web, web users and developers more and more appreciate the adherence to standards, for well-known common-sense reasons:

- If you follow a best-practice standard, this eases your own development work. You avoid reinventing the wheel.
- Access to your work is eased. Who knows the standard has better chances to reuse your work. This is the well-known accessibility aim for web data and services.
- Software and data conforming to standards can be shared over different hardware and software platforms. This feature is often called interoperability.

Mashup developers feel the advantage of standard compliance even more than other netizens, because their fate is combining resources that may come from far away in the web and that never were anticipated to meet in a mashup. These secondary uses are hard to achieve unless the integrated resources keep to their specifications. As essential underpinnings of mashups, web standards merit a closer look.

## 2.2 Grouping Mashup-Relevant Standards

Organizing standards is a problem because one cannot refer to an inherent order of the domain. Even big players may present their own standards output in a partially adequate order. W3C is a good example. They are sorting their standards by technology, status, date, title, working/interest group, or editor, and alphabetically in second sort position.[20] This is helpful, but far away from a systematic order. When mashups and mashup-relevant standards are of interest, one would also prefer an order that adapts to the newly set focus, in this case to mashups.

Mashups are widely distributed on the web. Especially innovative evolving web ecosystems tend to be less penetrated by standardization. For instance, look at the intermediate regularization state in the clouds.[21] The author uses cloud services in iPad apps without a need for specific cloud standards. When the authors of [12][22] describe their Web of Things experience with cloud services, they focus on the EPC (Electronic Product Code) network problems.

The W3C incubator group[23] sees social media in a pre-standardization data-silos state, because the bulk of specifications comes from the proprietary APIs of Facebook, Twitter, Google, LinkedIN and others, with their separated networking sites. Nevertheless the group lists many approaches with standards background, e.g. XMPP and OAuth mentioned below, so that social media standardization appears

[20] http://www.w3.org/TR/.

[21] Cloud standards list on http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/StandardsInventory. Active cloud standard committees are found on http://cloud-standards.org/wiki/index.php?title=Main_Page.

[22] http://www.vs.inf.ethz.ch/publ/papers/guinard_epcCloud.pdf.

[23] http://www.w3.org/2005/Incubator/socialweb/XGR-socialweb-20101206/.

to be in progress. Both XMPP and OAuth are shared with other application areas. Social media mashups combining two independent social networks are coming up.

In the same vein the AR (augmented reality) community is currently setting up standards for their field, repurposing many existing standards and following a systematic development procedure.

At least until the main standardization bodies offer authoritative arrangements, web users are summoned to order standards from their own viewpoint. Many options are possible; there is all reason for pragmatism and none to find out "was die Welt im Innersten zusammenhält". Grouping mashup-relevant standards for mashup developers and users must reflect the use situation: What do we use a standard for? In which context, with which regulation goal does it appear? In short the question is: *Which* standard *when* for doing *what*? Thus the ordering follows function:

1. *Web communication.* Almost all mashups draw resources from the web or write data to some external server. A web transport is needed, the standards/protocols for web transport apply.
2. *Data organization.* Data on the web must be organized according to preset schemes and syntaxes. This is a precondition for interoperability, i.e. assuring that resources can be interpreted elsewhere in the web as intended by the data producer or the producing system. The regulations may affect small units like characters, but also large ones like whole documents or texts. In user eyes they may appear as formatting schemes although developers may see them e.g. as protocols.
3. *Content types.* As demonstrated by XML formats or Java programs, the web is widely text-based. Life conditions change when one dares into content. Dealing therein with text, images, videos, audio and so on is a key task of semantic mashups. The relevant standards are considered starting with the highly common RDF format and ending with much more complex cases such as augmented reality standardization.
4. *Web queries.* Search is a frequent mashup activity. Especially linked data mashups connect via SPARQL searches. Query standards and specifications are needed.
5. *Human interfaces.* Designing helpful human interfaces is a core effort in mashup development and user-own mashup configuration. In contrast with other tasks, interface design is mainly regulated by guidelines. Companies (Google, Apple, etc.) provide them and put big investments in. Guidelines prescribe features and often support developers with ready-made templates. Since interfaces are near to users, maintaining a local look-and-feel is of evident interest.

The upcoming web standards parade requests some flexibility for avoiding common dead ends. Two of them:

- *The one type only trap.* While German Wikipedia[24] describes MP3 as a procedure of data compression ("Verfahren zur verlustbehafteten Kompression digital

[24] http://de.wikipedia.org/wiki/MP3.

gespeicherter Audiodaten”), English Wikipedia[25] sees it as an encoding format (“a patented encoding format for digital audio which uses a form of lossy data compression”). As MP3 is a German invention (the author met the main researcher Hans-Georg Musmann), the German version may be more authentic. Anyhow users are invited to switch types to the roles required in their task environment, without further discussion.

- *The fixed surroundings trap.* In their standards organization, W3C present the XML Document Model DOM[26] as a subcategory of ‘Scripting and AJAX’. Why? To the author it is known from its Java implementation in JDOM.[27] Thus transferring items into new environments is approved—everybody can quote precedents.

There is nothing wrong when for instance a developer-side protocol like SOAP appears on the user side as a format for the protocol execution, according to the user’s tasks. Similarly cooperative users of standards look over the fence and reuse what was invented elsewhere on the web. Context-driven switches of type, affiliation and other properties are normal.

## 2.3 The Web Standards Parade

In the following main section of the chapter, mashup-relevant standards parade in groups as listed above. Because this chapter belongs to a book on semantic mashups, content-related standards get their due share of attention. Depending on the local situation, the discussion centers on specific areas (e.g. security) or on concrete standards (such as XML).

### *2.3.1 Web Communication*

#### 2.3.1.1 HTTP and TCP/IP

HTTP and TCP/IP manage web communication in a client–server environment. The Internet Protocol Suite TCP/IP [9] includes the Transmission Control Protocol (TCP) and the Internet Protocol (IP). It includes several layers (see Fig. 2.3 adapted from [18]) reaching from network transmission to application services such as FTP or SMTP. The basic truth is that every computer in the web owns an IP address. IPs are provided by the Domain Name System (DNS).[28]

---

[25] http://en.wikipedia.org/wiki/MP3.

[26] http://www.w3.org/DOM/.

[27] http://www.jdom.org/.

[28] http://en.wikipedia.org/wiki/Domain_Name_System.

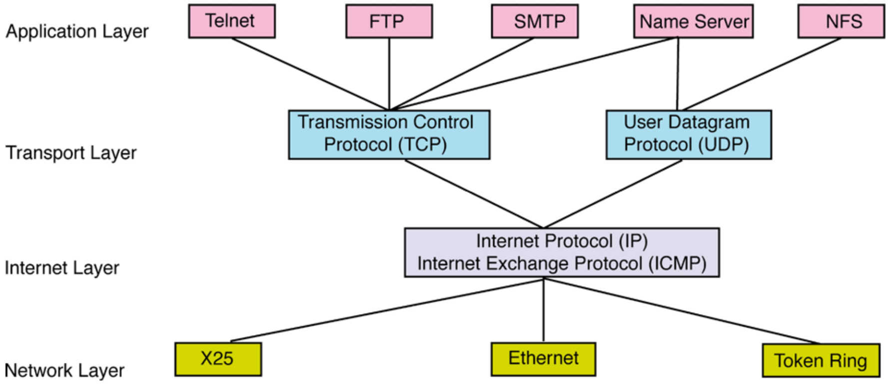



**Fig. 2.3** The TCP/IP architecture

HTTP[29] (Hypertext Transfer Protocol) is currently used in its HTTP 1.1 version. It opens reusable connections between two hosts. As well known, HTTP addresses show up in the headline of the browser when it is directed to a website. Everybody knows YouTube:

http://www.youtube.com/

An HTTP header may also contain some additional items. In the example

http://www.f5.com:80/path/time.jsp?h=13

the port 80 is specified, the data type is jsp (Java Server Pages), and after the question mark follows a short query. Web addresses are commonly called URLs (Unit Resource Locations). They verbalize numeric IP addresses, for instance

http://17.149.160.10/

is transformed into

http://www.apple.com/

The most common HTTP requests are GET and POST. In general, users will apply them from inside an HTML form.

#### 2.3.1.2 REST (Representational State Transfer)

The most popular style to address a web host is REST (Representational State Transfer) ([11],[30] [31]). Look at a REST example of my own use, a request string fetching web speech data that go straight to the audioplayer:

[29] Overview in http://www.w3.org/Protocols/, for original IETF RFC see http://tools.ietf.org/html/rfc2616.

[30] http://www.ics.uci.edu/~taylor/documents/2002-REST-TOIT.pdf.

```
cl_login=BrigitteEn&cl_app=NOspeech_EVAL&cl_pwd=8xxxxxxxx
&req_voice=klaus22k&req_text=
"Ein Stecker fuer die hoeheren Energien"
&prot_vers=2&cl_env=Mac OS X 10.7.2&req_asw_type=STREAM
```

REST[31] is built on web resources that are identified through URIs as usual on the web. It connects with HTTP as a service interface. In a typical REST request, the client discovers the URL of a service, it sends an HTTP call to this URL with a given HTTP command (GET, POST, PUT, etc.), a number of options (e.g., accepted format), and a payload in the negotiated format (e.g., in XML or JSON). In my example an audio file is returned.

Developers (the author included) like REST because its is easy to use. No official standard is needed to impose it as a standard approach [13].[32]

#### 2.3.1.3 SOAP (Simple Object Access Protocol)

SOAP is defined in a W3C standard.[33] According to a SOAP tutorial,[34]

> SOAP is a simple XML-based protocol to let applications exchange information over HTTP.

Sometimes this stance rises smiles in the web,[35] but SOAP has its defenders and users.[36] ProgrammableWeb of today (08-02-2012) records 904 Mashup APIs using SOAP while 3433 APIs use REST. A simple SOAP example points out why SOAP is less simple than REST use. For getting a Fahrenheit conversion of a Celsius degree one has to code and send a SOAP envelope:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<CelsiusToFahrenheit xmlns="http://tempuri.org/">
<Celsius>50</Celsius>"
</CelsiusToFahrenheit>"
</soap:Body>
</soap:Envelope>
```

The answer arrives in an XML formatted envelope:

---

[31] Good introduction: http://rest.elkstein.org/.

[32] http://www.vs.inf.ethz.ch/publ/papers/dguinard-rest-vs-ws.pdf.

[33] http://www.w3.org/TR/2000/NOTE-SOAP-20000508/.

[34] http://www.w3schools.com/soap/default.asp.

[35] Already in 2002 Amit Asaravala advocated giving SOAP a REST—http://www.devx.com/DevX/Article/8155.

[36] http://www.soapuser.com/index.html.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
<CelsiusToFahrenheitResponsexmlns="http://tempuri.org/">
<CelsiusToFahrenheitResult>122</CelsiusToFahrenheitResult>
</CelsiusToFahrenheitResponse>
</soap:Body>
</soap:Envelope>
```

After unpacking the XML, the value of 122 degrees Fahrenheit is available.

#### 2.3.1.4 Syndication: RSS (Really Simple Syndication) and Atom

All sorts of syndicated news, audio and video podcasts are streaming through the web, often in two popular feed formats in parallel: RSS 2.0.[37] and Atom.[38] RSS 2.0 is more frequently used, but the newer Atom format provides a larger hierarchy and more tags.[39] Under their XML head both formats organize a series of items in a hierarchy. The syntax prescribes the tag names and their sequence. Tools like parsers and validators are available.

RSS 2.0 defines information channels for its feeds. Channels have some header information followed by a sequence of items. For the sake of simplicity only one of them is shown in the following example:[40]

```
<?xml version="1.0"?>
<rss version="2.0">
<channel>
    <title>Real's HowTo</title>
    <link>http://www.rgagnon.com</link>
    <description>Useful code snippets for Java</description>
    <language>en</language>
    <copyright>Copyright by Real Gagnon</copyright>
    <pubDate>Mon, 23 Jun 2008 11:15:31 -0400</pubDate>
<item>
    <title>The PDF are updated</title>
    <description>Java (756 pages), Powerbuilder (197),
    Javascript (99) and VBS (32)</description>
    <link>http://64.18.163.122/rgagnon/download/index.htm</link>
    <guid>http://64.18.163.122/rgagnon/download/index.htm</guid>
    <pubDate>Mon, 23 Jun 2008 11:15:31 -0400</pubDate>
</item>
```

[37] http://www.rssboard.org/rss-specification.

[38] http://tools.ietf.org/html/rfc4287.

[39] More detail at http://www.rgagnon.com/javadetails/java-0556.html.

[40] From http://www.rgagnon.com/javadetails/java-0608.html, see also http://www.w3schools.com/rss/.

```
</channel>
</rss>
```

The following sample Atom feed from the same source exemplifies the more detailed Atom feed format:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
<title>Example Feed</title>
 <subtitle>A subtitle.</subtitle>
 <link href="http://example.org/"/>
 <updated>2003-12-13T18:30:02Z</updated>
 <author>
   <name>John Doe</name>
   <email>johndoe@example.com</email>
 </author>
 <id>urn:uuid:60a76c80-d399-11d9-b91C-0003939e0af6</id>
<entry>
   <title>Atom-Powered Robots Run Amok</title>
   <link href="http://example.org/2003/12/13/atom03"/>
   <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
   <updated>2003-12-13T18:30:02Z</updated>
   <summary>Some text.</summary>
 </entry>
</feed>
```

#### 2.3.1.5 Extensible Messaging and Presence Protocol (XMPP)

The Extensible Messaging and Presence Protocol (XMPP) [32][41] is an open technology for real-time communication, using XML as the base format for exchanging information. In essence, XMPP provides a way to send small pieces of XML from one entity to another in close to real time.[42]

XMPP builds on a series of IETF RFCs[43] and a list of extensions[44] to them.

XMPP core services provide channel encryption, authentication, presence check for web entities, messaging, service discovery, structured data forms, and workflow management.

XMPP is widely deployed for instant messaging, chat, data syndication and other application types. Main sponsors are Google, the US Secure Hosting Center, and Voxeo.

---

[41] http://xmpp.org.

[42] http://fyi.oreilly.com/2009/05/what-can-you-do-with-xmpp.html.

[43] http://xmpp.org/xmpp-protocols/rfcs/.

[44] http://xmpp.org/xmpp-protocols/xmpp-extensions/.

#### 2.3.1.6 Security: Encryption and Authentication

Internet security has evolved to an own field of investigation [20].[45] It may be a very serious concern in web communication. Among other things, hash encryption methods have been developed for secure Internet interaction. In critical tasks such as authentication they may even be obligatory.

The common encryption standards family is SHA (secure hash algorithm).[46] The SHAs are US Federal Information Processing Standards (FIPS).[47] SHA-1 and SHA-2 are in use, SHA-3 is currently being developed. MD5[48] was standardized by the Internet Engineering Task Force IETF in 1992.[49] It is frequently applied, but it has been found to be unsafe.

Authentication means identifying a computer user, authorization decides what the user is permitted to do. For users it is an evident advantage being easily admitted to resources of others on different sites, in particular in social networks. Just as evidently the security concerns of the resource owner must be protected.

Using OpenID[50] or OAuth,[51] users can authorize others to access their private resources without handing over their credentials to a foreign site. As often an image says more than a thousand words. Figure 2.4[52] explains how the two competing authentication/authorization services OpenID and OAuth ensure that only entitled users can access a resource. To admitted users, OpenID hands over a certificate, whereas OAuth provides a time-restricted key. Practical how-to advice for developers is offered e.g. by Google.[53]

### *2.3.2 Organizing Data*

Depending on web sectors different data formats are prevailing. JSON is a popular choice. The CSV (Comma-Separated Values) format is used frequently. Everybody agrees on them, but the main formats of the semantic web are different: HTML, RDF, and XML are the basic big three. They are all specified by a specific markup language. In contrast with HTML and XML, RDF is conceptual, with no implementation of its own.

---

[45] http://code.google.com/intl/de/edu/submissions/daswani/index.html.

[46] http://en.wikipedia.org/wiki/SHA-2.

[47] http://www.nist.gov/itl/fips.cfm.

[48] http://en.wikipedia.org/wiki/MD5.

[49] http://tools.ietf.org/html/rfc1321.

[50] http://openid.net.

[51] http://oauth.net.

[52] http://en.wikipedia.org/wiki/OAuth.

[53] https://developers.google.com/+/api/oauth.

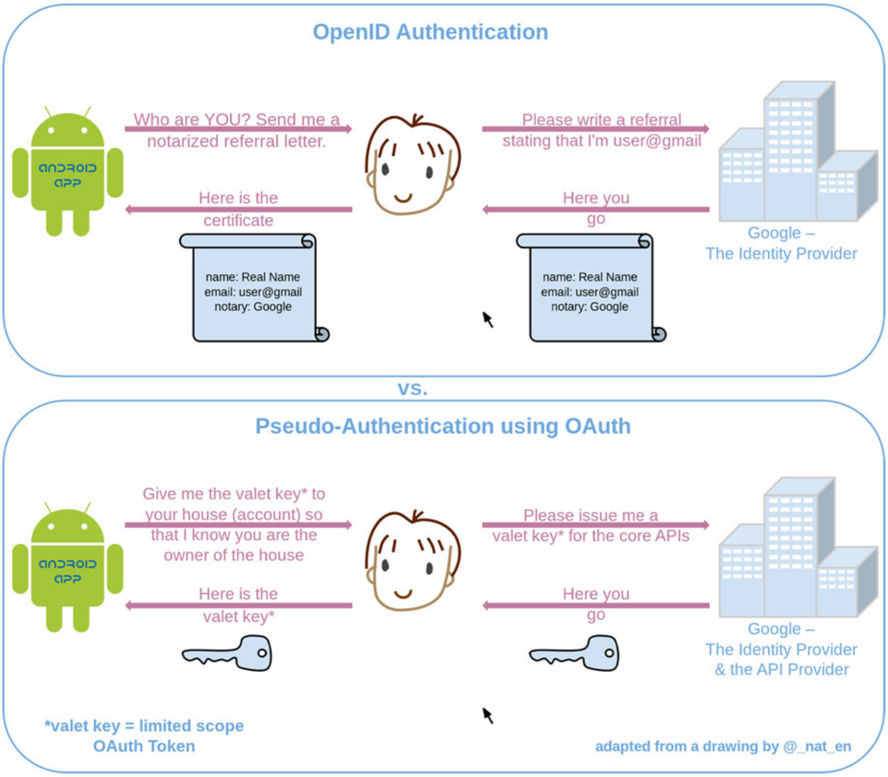



**Fig. 2.4** OpenID and OAuth authorize a user

Data standards of interest for the semantic web and semantic mashups are discussed advancing from simpler to more complicated formats. HTML comes first, whereas the recent and much more comprehensive HTML5 is in the end group.

#### 2.3.2.1 HTML

The most visible items on the web are the websites. Basically they are coded in HTML, the Hypertext Markup Language. HTML is specified and standardized by the W3C consortium.[54] Till HTML 4, style sheets such as CSS (Cascading Style Sheet)[55] or JavaScript[56] help to adapt HTML web content to the target presentation. For normal use, browsers (there are many of them: Chrome, Firefox, Internet Explorer, Opera, Safari, etc.) display HTML-style web content fetched from web

[54] http://www.w3.org/TR/html4/.

[55] http://www.w3.org/TR/CSS/.

[56] http://www.w3schools.com/js/, specification in ECMA-262 and ISO/IEC 16262.

servers. Web pages may be static or dynamic, i.e. being generated at question time from a database with the help of some code (often in PHP or AJAX). Browsers, document editors and so on may host plugins for handling special data types, e.g. video or PDF. HTML tutorials are legion on the web.

#### 2.3.2.2 XML

The XML family[57] is at least as important on the web as HTML and company. XML is a main carrier format for structuring/markup of web data. Today's XHTML[58] illustrates a common XML specialization. It is XML-based, i.e. it complies to the XML syntax, but it specializes it for its own HTML-style purposes. There are many application-specific XML markup languages behaving like this. For instance XML codes SVG graphics as well as the widely used syndication formats RSS 2.0 and Atom.

The XML example below reminds you of the basic syntax of XML: an XML declaration in the header line followed by a hierarchy of tagged entries with tags of your choice.

```
<?xml version="1.0" encoding ="utf-8"?>
<contact-info>
<name>Jane Smith</name>
<company>AT&amp;T</company>
<phone>(212) 555-4567</phone>
</contact-info>
```

In XSL,[59] the extensible stylesheet branch of the XML family, the most well known member is XSLT[60] for modifying XML structures.

XML Schema (XSD)[61] defines the structure of XML documents. Who wants to get along with less effort tries Relax NG[62] or the older DTD (Document Type Definition).

A human-friendly alternative to RDF/XML is offered by Turtle (Terse RDF Triple Language)[63] now in W3C draft status.

In the semantic web, much data representation builds upon the XML grounding. Indeed XML is—together with RDF—a backbone of the semantic web.

---

57 http://www.w3.org/TR/xml/, http://www.w3.org/XML/, http://www.w3.org/XML/1999/XML-in-10-points.html.en.

58 http://www.w3.org/TR/xhtml1/.

59 For an overview see http://www.w3.org/Style/XSL/.

60 http://www.w3.org/TR/xslt20/.

61 http://www.w3.org/XML/Schema, http://www.w3.org/TR/xmlschema-0/.

62 http://relaxng.org/, ISO/IEC STANDARD 19757-2 from 2008.

63 http://www.w3.org/TR/turtle/.

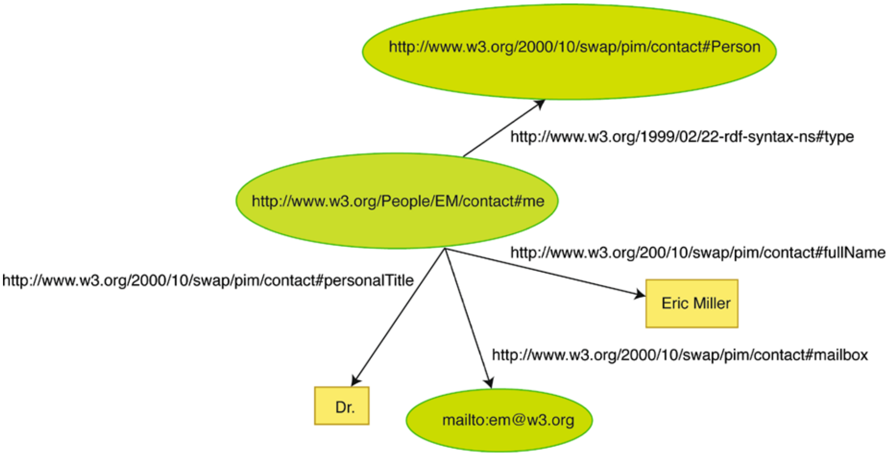



**Fig. 2.5** Remake of the classical RDF demo graph: Eric Miller

#### 2.3.2.3 RDF (Resource Description Format)

RDF[64] models concepts in a subject–relation–object pattern. The assumption is that all information can be broken down into triples of this format, corresponding to simple sentences. According to the assumption, all objects can be characterized with combined RDF-based descriptions. The basic RDF format is accompanied by RDFS schema for setting up simple concept relationships, and by RDFa for putting and finding RDF triples in running HTML text.

A classical drawing (Fig. 2.5) from the RDF Primer[65] explains what RDF contributes. Eric Miller is first characterized by a graph, then an XML-based description.[66] In the drawing, yellow squares represent strings, green ovals URLs/URIs. Remark that links are defined by URIs, too. In the XML format this information is mapped to tags and content items. A concept is characterized by as many RDF descriptions as needed.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
<contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>
```

64 http://www.w3.org/TR/rdf-concepts/, validator at http://www.w3.org/RDF/Validator/.

65 http://www.w3.org/TR/rdf-primer/.

66 Do no worry about the bothering length of URIs/URLs therein. CURIES (compact URIs—http://www.w3.org/TR/curie/) have been developed in the meantime. They are currently available for SPARQL, RDFa and XHTML2.

#### 2.3.2.3.1 RDFS—RDF Schema

A set of RDF triples or concepts is a vocabulary. Therein some organization is required in order to locate basic RDF triples or concepts built from them. The relations for that are provided by RDF Schema (RDFS),[67] the RDF Vocabulary Description Language. Again a standard example[68] (abridged) shows what happens. Descriptions of classes are located in the hierarchy with relations such as 'subClassOf'. In the example the passenger vehicle is specified as a subconcept of the vehicle because a 'subClassOf' link attaches it to its hypernym.

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xml:base="http://example.org/schemas/vehicles">
<rdf:Description rdf:ID="Vehicle">
 <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>
<rdf:Description rdf:ID="PassengerVehicle">
 <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
 <rdfs:subClassOf rdf:resource="#Vehicle"/>
</rdf:Description>
</rdf:RDF>
```

#### 2.3.2.3.2 RDFa

RDFa[69] is about RDF attributes in HTML or XML documents. So far RDF has been presented as mounted on XML. This is common, but not compulsory. The following RDFa[70] code fragment is thought to sit in an XHTML/HTML5[71] file. It enters RDFa-own tags like properties or attributes into XHTML items, here into a 'div' element. With its prefix 'dc' the property attribute refers to DublinCore.[72]

```
<div xmlns:dc="http://purl.org/dc/elements/1.1/">
   <h2 property="dc:title">The trouble with Bob</h2>
   <h3 property="dc:creator">Alice</h3>
   ...
</div>
```

[67] http://www.w3.org/TR/rdf-schema.

[68] http://ruleml.org/oojdrew/exa/exa4RDFS.html.

[69] Nice video by Manu Sporny at http://www.youtube.com/watch?v=ldl0m-5zLz4.

[70] http://www.w3.org/TR/2008/REC-rdfa-syntax-20081014/, http://www.w3.org/TR/2008/WD-xhtml-rdfa-primer-20080620/, current state of the RDFa call at http://rdfa.info/.

[71] http://dev.w3.org/html5/rdfa/.

[72] http://dublincore.org/.

RDFa attributes can be recovered from XML/HTML code. One major approach is gleaning them.[73] Practical applications are Google rich data using RDFa and the related microformats.[74]

#### 2.3.2.4 DOM (Document Object Model)

W3C DOM is a standard for accessing HTML and XML documents:[75]

> The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document.

The DOM recommendations are separated into three different parts/levels: core, HTML and XML.[76] Best known is the XML DOM API. It accesses the hierarchical XML document tree with all its nodes for parsing, writing and restructuring. As many XML documents run through the web, there is a considerable demand for XML DOM parsers. Off-the-shelf tools in several programming languages are available, and also helps for own parser development.

#### 2.3.2.5 JSON (Java Script Object Notation)

JSON is a very common data format. Its language-independence and simplicity enhances its popularity. JSON is a subset of the JavaScript Object notation. JavaScript is supported by virtually all browsers. ECMA standardizes JavaScript under the official name ECMAScript.[77]

JSON works with name/value pairs. Values may be numbers, strings, booleans, arrays, objects, and nil. In the introductory W3C example three employee objects are included in a larger object named "employees":

```
{"employees": [
{ "firstName":"John", "lastName":"Doe" },
{ "firstName":"Anna", "lastName":"Smith" },
{ "firstName":"Peter", "lastName":"Jones" }
] }
```

JSON objects can be accessed in JavaScript and in other programming/scripting languages.

---

[73] http://www.w3.org/TR/grddl/.

[74] http://microformats.org/.

[75] http://www.w3.org/DOM/.

[76] http://www.w3.org/DOM/DOMTR.

[77] http://www.ecmascript.org/.

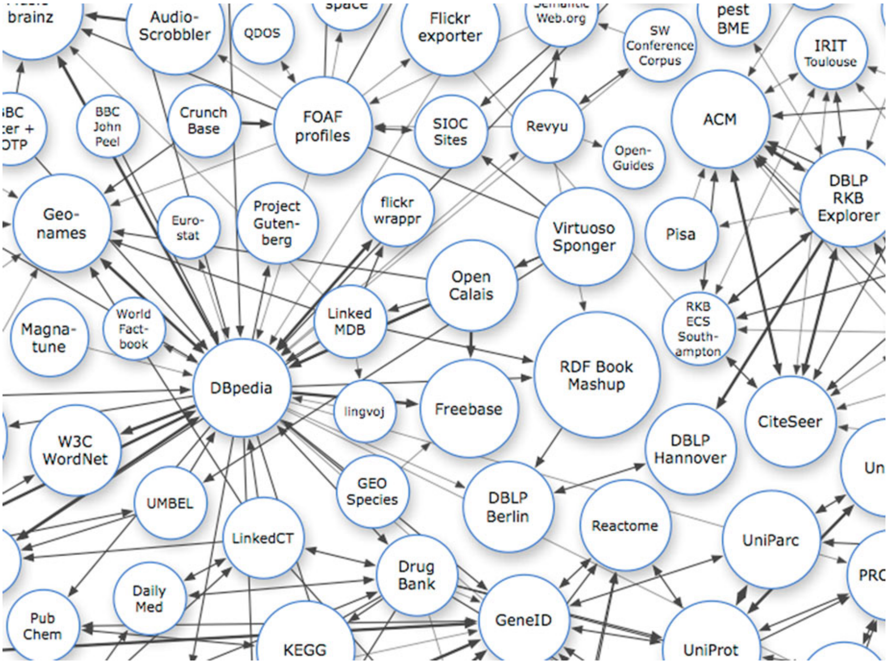



**Fig. 2.6** Linked Open Data—small sample of the data network

### 2.3.2.6 Linked Data [15]

Linked Data[78] are RDF triples from anywhere with a link in between. Because the triples and their links are identified by URIs, the linked data structure is a well-defined graph in the web. The triples represent concepts or things (not only documents), so that the web switches to a web of data or a web of things (WoT). Figure 2.6[79] shows a small subset of the global linked data space [14][80] with the main player DBpedia [4][81] and some resources it refers to. With their impressive size in the web linked data set standards by fact. Linked data are recognized by W3C without explicit standard/recommendation.

According to the advice of Berners-Lee[82] you support the linked data movement by conforming to four simple rules:

---

[78]http://linkeddata.org/home.

[79]http://linkeddata.org/static/images/lod-datasets_2009-07-14_cropped.png.

[80]http://linkeddatabook.com/book.

[81]http://www.wiwiss.fu-berlin.de/en/institute/pwo/bizer/research/publications/Bizer-etal-DBpedia-CrystallizationPoint-JWS-Preprint.pdf, http://tomheath.com/papers/bizer-heath-berners-lee-ijswis-linked-data.pdf.

[82]http://www.w3.org/DesignIssues/LinkedData.html.

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL).
4. Include links to other URIs, so that they can discover more things.

For more you may want to watch a video lecture[83] of Tom Heath and colleagues.

#### 2.3.2.7 OWL (Web Ontology Language)—A Biomedicine-Focused View

Ontologies are structured vocabularies. They reflect the semantics of their application domain. Basic links like the generic 'is-a' relation are borrowed from traditional thesauruses and library classification schemes. These also received an RDF-based web implementation and specification: the Simple Knowledge Organization System SKOS.[84] Metadata often refer to DublinCore specifications.[85]

Readers who doubt the impact of ontologies are invited to the National Center for Biomedical Ontology (NCBO).[86] Their BioPortal[87] gives access to a wealth of ontologies in biomedicine, from the Adult Mouse Brain (ABA) to the Zebrafish Anatomy and Development (ZFA). These and other ontologies set annotation standards for their scientific areas.

In biomedicine we find an environment of ontological tolerance. Methodological underpinnings differ. BioPortal supports ontologies in OBO format, OWL, RDF, Rich Release Format (RRF), Protégé frames, and LexGrid XML.[88] Look at a sample concept in the OBO format of Gene Ontology (GO):[89]

```
[Term]
id: GO:0002138
name: retinoic acid biosynthetic process
namespace: biological_process
def: "The chemical reactions and pathways resulting in the
biosynthesis of retinoic acid, one of the three components
that makes up vitamin A." [GOC:hjd]
synonym: "retinoic acid anabolic process" EXACT []
is_a: GO:0042573 ! retinoic acid metabolic process
is_a: GO:0044249 ! cellular biosynthetic process
```

[83] http://videolectures.net/iswc08_heath_hpldw/.

[84] http://www.w3.org/TR/2009/REC-skos-reference-20090818/, http://www.w3.org/TR/skos-primer/.

[85] http://dublincore.org/.

[86] http://www.bioontology.org/.

[87] http://bioportal.bioontology.org/ontologies.

[88] http://www.bioontology.org/bioportal.

[89] http://www.geneontology.org/.


**Fig. 2.7** OWL integration layers

OWL, the Web Ontology Language[90] is the most-cited ontology markup language of the semantic web. By matter of fact, Protégé[91] is the near-to standard editor for OWL ontologies. A large set of OWL ontologies is available from the Protégé Ontology Library.[92]

OWL 2 contains three degrees of specification: OWL DL and OWL Lite semantics are based on Description Logic, while OWL Full uses a more extended semantic model. Figure 2.7 illustrates the layered OWL setup. Currently, OWL Full waits for larger future applications.

OWL follows a concept import strategy. Constructs from RDF, RDFS and Description Logics (DL) [1] are reused. They are applied as-is or—in the case of DL—adapted to their new environment. Some subsidiary OWL constructs cover concepts for which no suitable import items are available.

OWL integrates the sophisticated concept building of Description Logics so that concepts can be constructed from their semantic features. The very popular example for demonstrating the effect is the grandmother (Fig. 2.8). Note that she is set up from highly recyclable concepts/features only: ‘female’, ‘male’, ‘person’, ‘parent’ and the relation ‘has.Child’. It is easy to imagine how such a reduction to basic concepts can simplify e.g. medical vocabularies.

**Fig. 2.8** Description Logic: Grandmother constructed from semantic features

$\text{Woman} \equiv \text{Person} \sqcap \text{Female}$
$\text{Man} \equiv \text{Person} \sqcap \text{Male}$
$\text{Mother} \equiv \text{Woman} \sqcap \text{hasChild.Person}$
$\text{Father} \equiv \text{Man} \sqcap \text{has.Child.Person}$
$\text{Parent} \equiv \text{Father} \sqcup \text{Mother}$
$\text{Grandmother} \equiv \text{Woman} \sqcap \text{hasChild.Parent}$

The following code sample taken from the OWL Primer[93] illustrates how constructs from RDF and RDFS work together with additional OWL tags: the OWL class ‘Father’ is built with imported ‘rdf:about’ and ‘rdfs:subClassOf’ links and an OWL-own ‘intersectionOf’ with roots in DL. Comparing the DL version of the ‘Father concept’ in Fig. 2.8 above with its notation below roughly illustrates how DL notation is transferred into an XML/RDF/OWL version.

[90] http://www.w3.org/TR/owl2-overview/.
[91] http://protege.stanford.edu/.
[92] http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library#OWL_ontologies.
[93] http://www.w3.org/2007/OWL/wiki/Primer.

```
<owl:Class rdf:about="Father">
        <rdfs:subClassOf>
                <owl:Class>
                        <owl:intersectionOf rdf:parseType="Collection">
                        <owl:Class rdf:about="Man"/>
                        <owl:Class rdf:about="Parent"/> </owl:intersectionOf>
                </owl:Class>
        </rdfs:subClassOf>
</owl:Class>
```

As mentioned before, the standard editor for OWL ontologies is Protégé. On the Protégé screen (Fig. 2.9), the minimal family first appears as hierarchy with additional relations that came over from DL. The second view shows a graphical visualization. The editor builds the RDF/XML code of OWL.

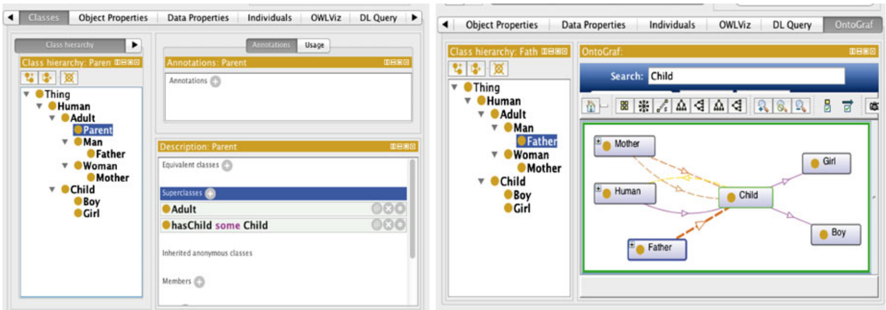



**Fig. 2.9** Protégé views of OWL ontology: Hierarchy on the *left*, visualization as graph on the *right*

## *2.3.3 Content Types*

The web is said to be text-based, as opposed to sound-based or image-based, although much web content uses media like image or audio. The text-based core of the web consists of organized character sequences (often encoded in ASCII) carrying core web information as in HTML pages, Java programs, Python scripts, etc. They serve technical purposes—their readers are in first line compilers and interpreters, humans reading them are mainly programmers.

Since earlier times, the contrasting non-ASCII MIME (Multipurpose Internet Mail Extensions) type covers attachments like pictures that emails transport as attachments. Iana[94] registers MIME media types like audio, text, image and 3D models.[95] The way how MIME types are declared is standardized as well, e.g. as "image/png" or "audio/mp4".

[94] http://www.iana.org/assignments/media-types/index.html.

[95] Listed on http://en.wikipedia.org/wiki/Internet_media_type, a sort of dictionary on http://www.myformatfactory.com/Format.

Graphics MIME types refer both to 2D and 3D dimensioning (overview in [22][96]). In 2D, raster graphics or bitmaps are composed of arrays of pixels while vector images describe the paths that will show up in the image. 3D graphic uses 3D models—3D representations of geometric data. Web 3D would refer to all technologies that enable 3D applications in the web. For them, the Web3D Consortium[97] is the first-line reference.

A contrasting view on how to manage web text as a content type of its own is held by the Text Encoding Initiative (TEI).[98] The Digital Library of Georgia[99] displays an extensive list of content-oriented media types. The Dublin Core DCMI Type Vocabulary[100] may also offer a suitable term.

In mashups, media types are used for transmitting content. In case of doubt, imagine mashups without the popular maps. Media transport content—this is what semantic mashups interpret. Without any claim for completeness, the following content media are considered:

1. PDF (text + image + more)
2. PNG (image)
3. SVG (image)
4. OpenGL (2D, 3D graphics)
5. X3D (interactive 3D)
6. MPEG's MP3 (audio) and MP4 (video)
7. Maps
8. HTML5
9. AR (augmented reality)

#### 2.3.3.1 PDF (Portable Document Format)

The great merit of PDF is the platform-independent representation of text and image documents. PDF was proposed by Adobe and still keeps features of a proprietary format. The PDF ISO standard 32000-1 released in 2008 refers to Adobe PDF 1.7.[101] Altogether the ISO Store[102] offers 13 separate PDF Standards.

---

[96] http://research.microsoft.com/pubs/69076/cga03.pdf.

[97] http://www.web3d.org/realtime-3d/.

[98] http://www.tei-c.org/index.xml.

[99] http://dlg.galileo.usg.edu/MediaTypes/.

[100] http://dublincore.org/documents/2012/06/14/dcmi-terms/?v=elements#H7.

[101] http://www.adobe.com/devnet/pdf/pdf_reference.html, http://partners.adobe.com/public/developer/en/pdf/PDFReference.pdf.

[102] http://www.iso.org/iso/store.htm.

PDF is widely used and has considerable industrial support.[103] Special purpose versions of PDF have their own standards background, as the PDF/Xs[104] that targets different tasks in the printing industry.

#### 2.3.3.2 PNG (Portable Network Graphics)

PNG (Portable Network Graphics) is a bitmapped image format.[105] It improves over its predecessors GIF and TIFF.[106] Images in PNG format are smaller than in TIFF and therefore more appropriate for web use. With respect to GIF, PNG performs better on alpha channels (variable transparency), gamma correction (cross-platform control of image brightness), and two-dimensional interlacing (a method of progressive display). PNG also compresses better than GIF, and in contrast to GIF it is available without fees because no patents are used. PNGs are processed almost everywhere where bitmaps are expected. They compress well without losses. However, PNG does not support non-RGB color spaces such as CMYK so that PNGs must be converted for printing.

#### 2.3.3.3 SVG (Scalable Vector Graphics)

Scalable vector graphics[107] are two-dimensional. All modern web browsers render them to different degrees. Graphics behavior is stated in XML files so that developers can specify and update them on a text editor.[108] The following code sample illustrates the SVG XML representation for the beginning of the TCP/IP drawing in Fig. 2.3.

```
<?xml version="1.0" encoding="utf-8"?>
<!--Generator: Adobe Illustrator 15.0.0. SVG Export Plug-In .
SVG Version: 6.00 Build 0) -->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://wwww.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" id="Ebene_1" xmlns="http://wwww.w3.org/2000/svg"
xmlns:xlink="http://wwww.w3.org/1999/xlink"
```

---

[103] http://www.callassoftware.com/callas/doku.php/en:products:pdftoolbox, http://www.pdf-tools.com/pdf/produkte-pdf-pdfa.aspx, http://www.helios.de/web/EN/solutions/FoDTP.html, and others.

[104] http://www.globalgraphics.com/technology/pdfx/.

[105] Joint specification of W3C and ISO/IEC: http://www.libpng.org/pub/png/spec/iso/, http://tools.ietf.org/html/rfc2083.

[106] http://www.libpng.org/pub/png/pngintro.html.

[107] http://www.w3.org/TR/SVG/.

[108] See for instance SVG-edit—http://www.youtube.com/watch?v=ZJKmEI06YiY.

```
x="0px" y="0px" width="728.5px" height="515.91px"
viewBox="0 0 728.5 515.91" enable-background="new 0 0 728.5 515.91"
xml:space="preserve">
<text transform="matrix(1 0 0 1 74.2256 70.8711)"
font-family="Helvetica" font-size="10">Application Layer</text>
<text transform="matrix(1 0 0 1 74.2256 135.7739)"
font-family="Helvetica" font-size="10">Transport Layer</text>
......
</svg>
```

#### 2.3.3.4 OpenGL (OpenGraphics Library)

OpenGL[109] is a royalty-free platform-independent industry standard serving both 2D and 3D graphics development. OpenGL is widely used in CAD, virtual reality applications, visualization and computer games, also on mobile devices. In the author's working context it appears on iPhone and iPad. As in earlier cases we observe on the Khronos site[110] a whole family of standards, this time of industry specifications for the web, serving also embedded systems as WebGL[111] does.

#### 2.3.3.5 X3D (Extensible 3D)

X3D[112] is an ISO markup language standard for defining and communicating real-time, interactive 3D content for visual effects and behavioral modeling. It is replacing the Virtual Reality Modeling Language (VRML). X3D uses XML and integrates with relevant other technologies (e.g. MPEG-4 or SVG). According to its authors X3D is royalty-free, scalable and open. The fit into HTML5 is explored by x3dom.[113]

X3D is on its way to current browsers. Serving the tiny piece of code in Fig. 2.10 to Chrome equipped with instantreality[114] shows you two blue triangles (on the right of the figure) that you can move around.

[109] http://www.khronos.org/opengl, http://www.opengl.org/documentation/specs/version1.5/glspec15.pdf.

[110] http://www.khronos.org/.

[111] http://www.khronos.org/webgl/.

[112] http://www.web3d.org/x3d/specifications/x3d/, http://www.web3d.org/realtime-3d/x3d/what-x3d.

[113] http://www.x3dom.org/.

[114] http://www.instantreality.org/downloads/.

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile='Immersive'>
<Scene>
<Shape>
<Appearance>
<Material diffuseColor="0 0 3" />
</Appearance>
<TriangleSet solid='false'>
<Coordinate point='
0 1 0, 1.5 0 0, 1.5 1 1.8,
1.5 1 1.8, 3 0 1.8, 3 1 0' />
</TriangleSet>
</Shape>
</Scene>
</X3D>
```

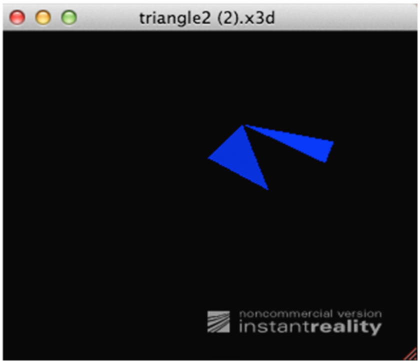



**Fig. 2.10** X3D code (*left*) and its result on the Chrome window (*right*)

#### 2.3.3.6 MPEG—Motion Pictures and Audio Standardization

The Moving Pictures Expert Group (MPEG)[115] is a major player on audio and multimedia specifications. According to their own mission statement,[116] their area of work is the

> Development of international standards for compression, decompression, processing, and coded representation of moving pictures, audio, and their combination, in order to satisfy a wide variety of applications.

Their list of standards[117]—delivered via ISO or under way—is just as impressive as those of other big providers.

The main influences on web audio and video practice come from the standards MPEG-1, MPEG-2 and MPEG-4. They are the technical background of MP3 audio and MP4 digital video. Television and radio rely on them. Internally the mentioned standards are bundles of specifications. Their inside view is displayed on the MPEG site. Figure 2.11[118] provides a minimal orientation of the relationships between the audio and video standards of our interest. VCD is video CD, DVD is optical disk, DVB is digital video broadcasting. DVB-S2 is its satellite-based follower of the second generation certified by the European Telecommunications Standard Institute ETSI[119] in 2005. PS is the program stream defined in MPEG-1, TS is the transport stream. An overall introduction to video, DVD, and digital sound is available at karbosguide.[120]

MPEG standards are charged with patents and engender license fees.[121]

[115] http://mpeg.chiariglione.org.

[116] http://mpeg.chiariglione.org/terms_of_reference.php.

[117] http://mpeg.chiariglione.org/standards.php.

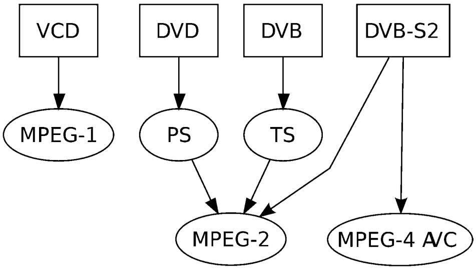
[118] From http://en.wikipedia.org/wiki/File:MPEG.svg.

[119] http://www.etsi.org/website/homepage.aspx.

[120] http://www.karbosguide.com.

[121] http://en.wikipedia.org/wiki/MPEG-4.

**Fig. 2.11** Main relationships of audio and video standards




MPEGLA[122] manages them, but not all. On their patent pools site[123] they tell that they stopped dealing with MPEG-4, but anyway they list many companies that are in with specific patents. MPEG-4 Visual is still managed by MPEGLA with "reasonable" (sic) royalties.

The situation of MP3 patents is just as unclear. MPEGLA asserts to manage them. However, from mp3-tech[124] one learns that only Fraunhofer/Thomson patents require royalties, with a link to their price list. It is easy to imagine that the many involved companies litigated on their deals and rights. Indeed US courts decided on huge indemnities.

Although MP3 is widely used, it has to share the market with some alternative options. A few of them:

- WMA—Windows Media Audio[125]
- WAV—Waveform Audio[126]
- AIFF—Audio Interchange File Format[127]
- MIDI—Musical Instrument Digital Interface[128]
- Ogg Vorbis—a free, open and un-patented music format[129]

Remarkable is the Chinese Audio Video Standard AVS.[130] According to its owners, it is faster than MPEG-2 and MPEG-4. It is owned by China and frees Chinese developers from MPEG royalties.

[122] http://www.mpegla.com/main/default.aspx.

[123] http://www.mpegla.com/main/programs/M4S/Pages/PatentList.aspx.

[124] http://www.mp3-tech.org/patents.html.

[125] http://en.wikipedia.org/wiki/Windows_Media_Audio.

[126] https://ccrma.stanford.edu/courses/422/projects/WaveFormat/, http://msdn.microsoft.com/en-us/windows/hardware/gg463006.aspx.

[127] http://www-mmsp.ece.mcgill.ca/documents/audioformats/aiff/aiff.html, http://muratnkonar.com/aiff/index.html.

[128] http://www.midi.org/.

[129] http://xiph.org/vorbis/doc/.

[130] http://www.avs.org.cn/en/index.asp.

#### 2.3.3.6.1 MP3

MP3 is a de facto standard for consumer digital audio. It is an audio compression technology based on the MPEG-1 (lossy audio and video compression) and MPEG-2 (update and extension) specifications, referring to the MPEG Audio Layer 3. MP3 compresses CD quality sound by a factor of 8–12, while maintaining almost the same high-fidelity sound quality. MP3 compression loses sound depending on the compression rate, but it loses intelligently:

- It keeps the audible audio data at the cost of less audible ones.
- Loud sounds drown softer ones, so that the softer sounds are removed.
- Human ears hear best in the range of 1–4 kHz. Sounds in this zone remain unchanged, while higher or lower ones may by reduced.
- In the marginal areas, common information of stereo sounds may be reduced to mono sound.

Systems apply many additional compression strategies, and often combine them. [26][131] reviews the digital audio standards. Who wants to learn more about the effort that lead to MP3 can read the account of a highly involved Hanover scientist [25].

An MP3 file is made up of multiple MP3 frames, which consist of a header and a data block. The MP3 data blocks contain the (compressed) audio information in terms of frequencies and amplitudes. Today most MP3 files contain ID3 metadata[132] that precede or follow the MP3 frames.[133] Frames are fed into each other in a Matryoshka style [30].[134]

#### 2.3.3.6.2 MP4

As an MP3 file extension makes think about music or audio, MP4 makes think about video. Again a drawing (Fig. 2.12[135]) conveys an initial orientation, this time of what works inside an MP4 file. Additionally it explains that MPEG-4 grew out of Quicktime. Although users perceive MP4 as an integrated multimedia technology, a whole standards family stands behind it. MP4 or MPEG-4 AVC (Advanced Video Coding)[136] is a container format. It can contain all kinds of multimedia

---

[131] http://erdos.csie.ncnu.edu.tw/~hychen/multimedia/mpeg%20audio%20coding.pdf.

[132] http://en.wikipedia.org/wiki/ID3, http://www.id3.org/.

[133] From http://en.wikipedia.org/wiki/MP3.

[134] http://www.mp3-tech.org/programmer/docs/mp3_theory.pdf, extended specification at http://www.multiweb.cz/twoinches/mp3inside.htm.

[135] Remake of a drawing from http://www.tansee.com/what-is-mp4-video.html.

[136] http://en.wikipedia.org/wiki/MPEG-4_Part_14.

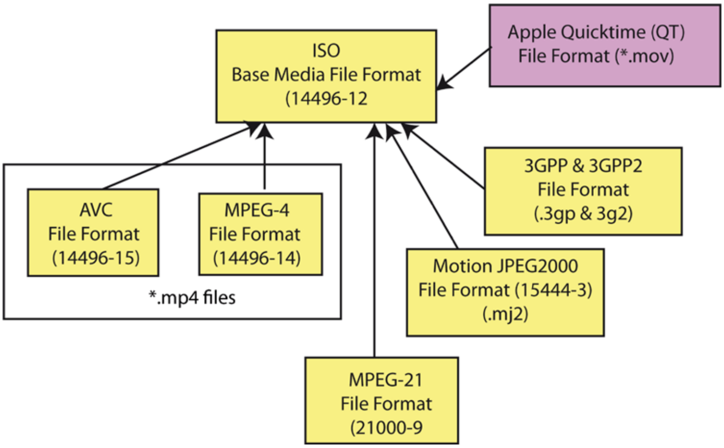



**Fig. 2.12** The standards background of MP4

content—like audio, video, 2D and 3D graphics and animated avatars, besides user interactivity features. MPEG-4 stores audio, scenes (described by the scene language BIFS), and other multimedia content using the ISO Base Media File Format, while AVC means the storage of AVC (ISO/IEC 14496-10/AVC) standard 4 data (which may be audio, video, or sub-scenes) within the files of the ISO Base Media File Format.[137] MP4 containers are box-structured files.[138] A box-structured file consists of a series of boxes that have a size and a type. The file structure is object-oriented. All box-structured files start with a file-type box (possibly after a box-structured signature) that defines the best use of the file, and the specifications to which the file complies.

The files have a logical structure, a time structure, and a physical structure:

- The logical structure of the file is that of a movie which in turn contains a set of time-parallel tracks.
- The time structure of the file is made of the temporally ordered samples in the tracks.
- The physical structure separates the data needed for logical, time, and structural de-composition from the media data samples themselves. The structural information is concentrated in a movie box.

Each media stream is contained in a track specialized for that media type (audio, video, etc.), and is further parameterized by a sample entry. The sample entry

[137] http://www.myformatfactory.com/MP4.

[138] http://thompsonng.blogspot.com/2010/11/mp4-file-format.html explains inside MP4 with sample code.


**Fig. 2.13** Theaters in Chicago—users see what is on stage in the district

contains the name of the exact media type (i.e., the type of the appropriate decoder) and any parameterization as needed by decoders.[139]

### 2.3.3.7 Maps

The most popular media to be mashed into an application are maps, possibly superimposed with location information on persons and all sorts of things, e.g. businesses. A recent map site reminds you of the information value of map media (Fig. 2.13). It displays tonight's program of Chicago theaters,[140] nicely exemplifying the attractivity and the information value of map-driven user services. On the hit list of digital map delivery, Google Maps[141] is most probably number one, followed by Yahoo

[139]Taken from http://www.tansee.com/what-is-mp4-video.html, more detail there.

[140]http://www.theatreinchicago.com/maps/mapTheatres.php.

[141]http://maps.google.com/, http://code.google.com/apis/maps/documentation/geocoding/.

maps[142] and other providers.[143] ProgrammableWeb knows of 2351 mashups using Google Maps.[144] Additional evidence for the popularity of digital maps comes from related listings[145] on the web.

Who wants to integrate a Google map into the own application, applies the Google Maps API for web services.[146] First results are easily achieved, so that complying with the Google Maps API specification is immediately rewarding. Please register that map standardization is driven by industrial deployment.

Correspondingly to the thriving application area the realm of map services standards is flourishing. The Open Geospatial Consortium (OGC) presents an impressive list,[147] containing the Web Map Service[148] and many other standards. The Web Map Service Interface Standard (WMS) therein provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. It returns geo-registered map images that can be displayed in a browser.

Keyholes are points on earth defined by longitude, latitude and altitude for the use in Google Earth and other applications. The respective Keyhole Markup Language (KML) is developed by Google and standardized by OGC.[149]

Points of interest (POI) are locations that may be important for somebody for some reason. Usually they are defined by their geographic longitude and latitude. They obtain scientific interest as well, but interestingly a standard for points of interest is still missing. GPS provides an industrial specification.[150] Commercial and free collections of POIs are available.

The Geography Markup Language (GML)[151] standardizes an XML-based representation of geographical features. A code sample from the GML 3.1 specification[152] displays how an XML/XSD configuration works for geographic observations:

```
<gml:DirectedObservationAtDistance>
<gml:validTime>
<gml:TimeInstant>
<gml:timePosition>2002-11-12T09:12:00</gml:timePosition>
```

142 http://uk.maps.yahoo.com/.

143 http://www.natgeomaps.com/, http://www.geocomm.com/, http://www.mapsolutions.com.au/.

144 http://www.programmableweb.com/api/google-maps/mashups.

145 http://mashable.com/2009/01/08/google-maps-mashups-tools/, http://googlemapsmania.blogspot.com/, search in http://mashable.com/.

146 http://code.google.com/intl/en-EN/apis/maps/documentation/webservices/index.html.

147 http://www.opengeospatial.org/standards/.

148 http://www.opengeospatial.org/standards/wms.

149 Currently KML 3.2, http://www.opengeospatial.org/standards/kml/, https://developers.google.com/kml/documentation/.

150 http://en.wikipedia.org/wiki/GPS_eXchange_Format.

151 http://www.opengeospatial.org/standards/gml, http://www.ogcnetwork.net/gml.

152 http://www.opengeospatial.org/standards/gml.

```
</gml:TimeInstant>
</gml:validTime>
<gml:using xlink:href="http://www.my.org/cameras/leica2"/>
<gml:subject xlink:href="http://www.tourist.org/sights/mountain3"/>
<gml:resultOf xlink:href="http://www.my.org/photos/landscape1.jpg"/>
<gml:direction>
<gml:CompassPoint>NW</gml:CompassPoint>
</gml:direction>
<gml:distance uom="#m">16500.</gml:distance>
</gml:DirectedObservationAtDistance>
```

#### 2.3.3.8 HTML5

After good old HTML, HTML5[153] is a big jump forward. HTML5 is still a W3C candidate recommendation,[154] but it is already widely applied in the web. There are galleries full of sites coded in HTML5.[155] Some big web companies support HTML5. Take a look at Apple's HTML5 show page,[156] or at the still more engaged Google HTML5Rocks resource.[157]

There we learn the basic HTML5 recipe:

HTML5 = HTML + CSS + JS

Cascading Stylesheets (CSS)[158] for presentation and JavaScript (JS)[159] for client-side scripting are integrated with the HTML functionality, so that with the HTML5 environment alone, users dispose of virtually all means for setting up a site. In addition, HTML5 is equipped with graphics tools—SVG and Canvas. It offers video and audio tags and plays video and audio files without additional software.

A tiny test of the HTML5 advances: Put the following piece of SVG-including code into a file and pass it to your browser. It will draw a green ball.

```
<!doctype html>
<title>SVG in text/html</title>
<p>
 A green circle:
 <svg> <circle r="50" cx="50" cy="50" fill="green"/> </svg>
</p>
```

Apple and Google are not alone in supporting HTML5 developers. HTML5 tutorials abound on the web.

[153] http://dev.w3.org/html5/spec/Overview.html, for good examples see http://html5demos.com/.

[154] http://www.w3.org/TR/html5/.

[155] http://html5gallery.com/; http://html5websites.net/.

[156] http://www.apple.com/html5/.

[157] http://www.html5rocks.com/en/.

[158] http://www.w3.org/TR/CSS/.

[159] http://en.wikipedia.org/wiki/JavaScript, specification in ECMA-262 and ISO/IEC 16262.

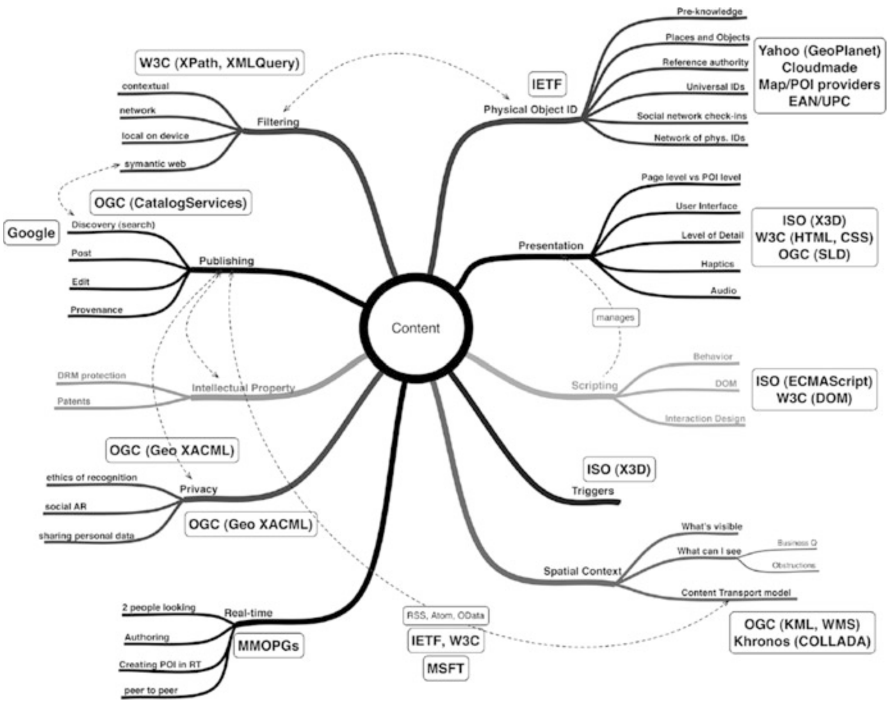



**Fig. 2.14** AR standardization with integration of existing standards from [29]

#### 2.3.3.9 Augmented Reality (AR)

Augmented reality (AR) adds some virtual information to a real-world environment. This can be helpful because it makes at a time visible on the same camera screen what otherwise must be looked up separately. Positive effects are evident for instance when a learner sees the real-world problem together with a virtual inset explaining how to solve it.

The field is recent and successful, so that a need for standards is felt. Remarkably standardization is discussed in a collaborative style with integration of existing standards that can be reused for AR: JSON, HTML5, KML, GML, CityGML, X3D can be repurposed. [29][160] focus on mobile AR. They propose to take over standards from several sources, for example Sensor Web Enablement (SWE) standards maintained by OGC.[161] The aim is to avoid mistakes and to speed up the standardization process. Figure 2.14 illustrates how much earlier work can be exploited. At the same time it makes manifest that realizing and standardizing AR is a complex task. Many subresources must be incorporated.

[160] http://www.perey.com/ARStandards/[Springer_Chapter]Standards_for_Mobile_Collaborative_AR.pdf.

[161] http://www.opengeospatial.org/projects/groups/sensorwebdwg.

[23][162] proposes a step-by-step procedure for standards creation:

1. Stakeholder analysis
2. Architecture analysis
3. Scope definition
4. Usecase definition
5. Breaking it down
6. Analyze existing standards and standard development organizations (SDOs)
7. Fill the gaps
8. Feasibility check

In [24][163] the XML-based Augmented Reality Markup Language (ARML) is explained. Its specification is available on the Wikitude developer page.[164]

## *2.3.4 Web Queries*

While elsewhere in this chapter web (pre)-history is almost disregarded, this is less defendable for retrieval technologies, simply because the web inherits so much from earlier approaches. After all Edgar F. Codd proposed the relational data model in prehistoric times (around 1970), together with some relational algebra and the retrieval algorithms which were called SQL. In information retrieval the first implemented approaches appear already in the late 1940s.[165] When ANSI and ISO began standardizing SQL for search in relational databases in 1986/1987 they encountered a good state of earlier work. Today SQL development is going on, currently a new working draft is discussed.[166]

A timeline of newer web history (from [3]—Fig. 2.15) shows that only after 1995 pre-existing retrieval and query strategies were reworked for use on the web. Web search for structured data adapted the predefined SQL search techniques: SQLite is popular for web databases, XQuery searches XML data, and SPARQL searches RDF graphs. Since RDF data most of the time serializes in XML format, both XQuery and SPARQL apply to RDF data.

### 2.3.4.1 SQLite SQL

SQLite ([28], [21][167]) is an SQL database engine that embeds into applications—because of its small footprint also on cellphones, PDAs, and MP3 players. It is

[162] http://www.navteq.com/outdoor_mar2011/ar_standards_position_paper.pdf.

[163] http://www.perey.com/MobileARSummit/Mobilizy-ARML.pdf.

[164] http://openarml.org/wikitude4.html.

[165] http://en.wikipedia.org/wiki/Information_retrieval.

[166] http://www.wiscorp.com/SQLStandards.html.

[167] http://www.sqlite.org/.

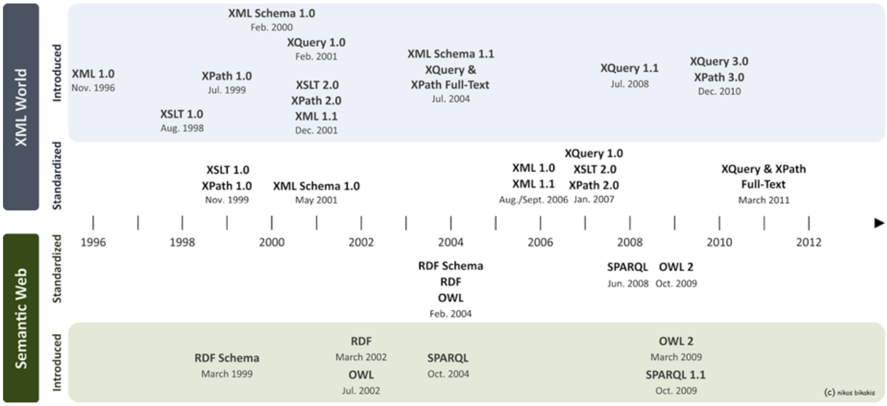



**Fig. 2.15** W3C standards for XML and RDF—recent timeline. Web query development starts around 1995

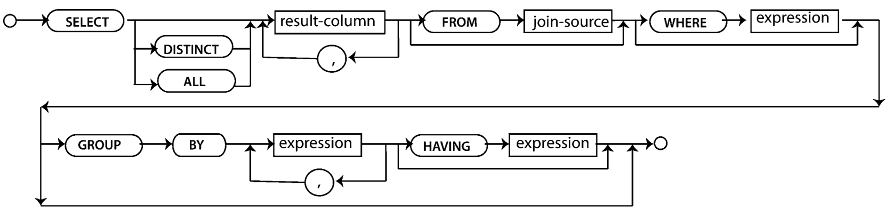



**Fig. 2.16** SQL select as understood by SQLite

platform independent, binds to many programming languages and adapts to low memory constraints. SQLite is heavily used as embedded database engine in the web. Wikipedia qualifies it as "arguably the most widely deployed database engine". Interestingly SQLite achieves this diffusion with a mixed standards behavior—implementing most of the SQL standard, but also specific features of its own.

The author's moderate experience was acquired on iPhone/IOS5.[168] A select command "as understood by SQLite"[169] (Fig. 2.16) recalls core SQL query syntax and shows how SQLite conforms to it.

### 2.3.4.2 XQuery

XQuery[170] is a W3C-specified query language for XML data sources. The whole XQuery family including XPath, XSLT and others is strongly supported by W3C.[171]

168 Up-to-date tutorial at http://klanguedoc.hubpages.com/.

169 From http://sqlite.org/lang_select.html.

170 http://www.w3.org/TR/xquery/.

171 http://www.w3.org/XML/Query/#specs.

**Table 2.1** An XQuery expression selects a bike from XML data

| itemsBase.xml | XQuery expression |
|---|---|
| <items><br><item_tuple><br><itemno>1001</itemno><br><description>Red Bicycle</description><br><offered_by>U01</offered_by><br><start_date>1999-01-05</start_date><br><end_date>1999-01-20</end_date><br><reserve_price>40</reserve_price><br></item_tuple><br><item_tuple><br><itemno>1004</itemno><br><description>Tricycle</description><br><offered_by>U01</offered_by><br><start_date>1999-02-25</start_date><br><end_date>1999-03-08</end_date><br><reserve_price>15</reserve_price><br></item_tuple><br></items> | for $item in<br>document("itemsBase.xml")/items/item_tuple<br>where $item/reserve_price > 30<br>order by $item/description<br>return <item>{$item/description, $item/reserve_price}</item> |

New query expressions and processing models are defined in the XQuery Update Facility 1.0 recommendation of March 2011.[172] Many XQuery tutorials are available.

XQuery is running on web XML databases.[173] A code example from ipedo[174] (Table 2.1) illustrates the use of XQuery expressions. From a tiny XML bike resource (left), the query statement (on the right) retrieves the red bicycle because its price meets the query condition.

#### 2.3.4.3 SPARQL

SPARQL (Protocol and RDF Query Language) is a W3C recommendation since 2008.[175] It searches RDF-coded data resources, so that one finds RDF triples and their conjunctions, disjunctions and additional patterns. SPARQL has considerable industry support. It is implemented for many programming languages.[176] SPARQL queries can be transferred into SQL and XQuery formats.[177] The opposite direction—exploiting and rewriting non-RDF relational databases in RDF format—is in the implementation state.[178]

[172] http://www.w3.org/TR/xquery-update-10/.

[173] See e.g. http://exist-db.org/exist/index.xml.

[174] http://www.ipedo.com/.

[175] http://www.w3.org/TR/rdf-sparql-query/.

[176] Listing of implementations at http://www.w3.org/wiki/SparqlImplementations.

[177] http://en.wikipedia.org/wiki/SPARQL, http://www.dblab.ntua.gr/~bikakis/SPARQL2XQuery.html.

[178] http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/.

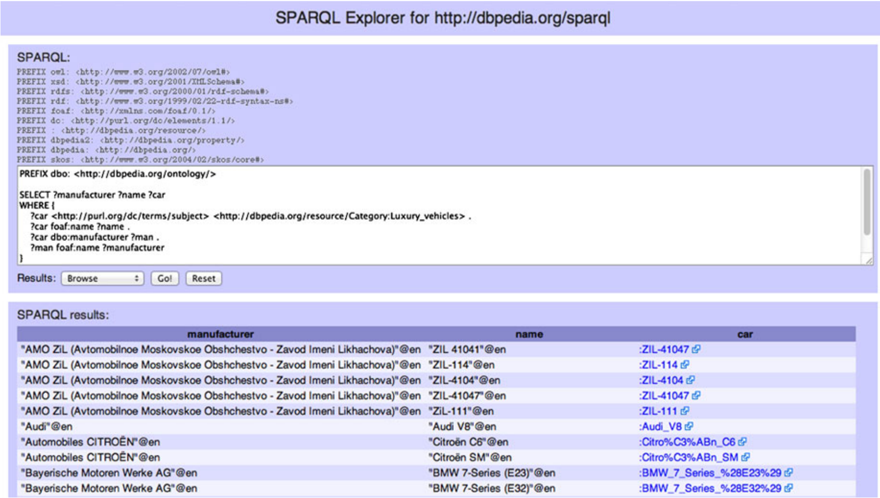



**Fig. 2.17** SPARQL luxury cars example—remark prefixes and SQL query expression

Protégé includes a SPARQL retrieval technique for OWL ontologies, with the restriction that availability in Protégé 4 is still unclear.[179]

DBpedia offers a SPARQL endpoint.[180] The luxury cars example from there (Fig. 2.17) explains all we need for instant purposes. The prefixes show which resources are used. The SQL-style select query uncovers the SQL roots. Who wants to see XML and JSON formats changes the presentation format.

SPARQL entailment standards for expanding queries to implicit results are under way.[181]

### *2.3.5 Human Interfaces*

Human interfaces are the realm of Human Computer Interaction (HCI). As most mashups are made to serve human users, interfaces are a main issue for mashup developers. Long-standing research and HCI classics like [27] and [7] let us assume that developers have a good grasp of interfaces, in particular of Graphical User Interfaces (GUIs). However, today's user interfaces have advanced over their classical ancestors by being interactive, touch-sensitive, including auditory components, and so on. Media they present to users may vary—from songs to maps, mathematical formulas, soundscapes, webcams or surgical cut videos.

---

[179] http://protege.stanford.edu/doc/sparql/.

[180] Reached from http://wiki.dbpedia.org/OnlineAccess#h28-2.

[181] http://www.w3.org/TR/2012/WD-sparql11-entailment-20120105/.

Long-standing HCI knowledge[182] also might entail the assumption that a common understanding of interfaces resulted in generally accepted specifications or standards. According to available evidence this is not the case. Some standards are indeed available[183] but they are a minority group. [2] presents an overview of the majority inside the minority group: the ISO-based standards.

Interfaces are a guideline country. One reason for the predominance of guidelines is that interfaces communicate with users, so that flexibility, style and esthetics play an important role. Interfaces mediate between users and applications, they must adapt to both sides—the system's services must be handled so that users succeed in working with them without unnecessary clashes. Design strives for functionality and usability as for recognition value, esthetics and corporate style (the "look and feel"). Users interact with the interface, they have to feel at home with it.

Interface guidelines for general audiences are available, scaling from rule-of-thumb rules to very sophisticated and recommended work.[184] Among newer (e)books one can hint at [33] and [10].

Companies invest massively into web guideline development. This can be observed with IBM: "Good design is good business."[185] Apple can serve as an example for specializing guidelines to tasks and devices such as Mac, Bluetooth, USB device, iPhone, iPad, etc. Thirty interface-oriented guidelines and references come up from the OS X Developer Library.[186] SAP[187] helps companies to run better, their user-centered design guidelines[188] target business tasks and users. Many other enterprises follow the same strategy for their own businesses or devices.[189]

The impact of map services visualization is evident enough to motivate a discussion of geographic information presented according to a few guidelines in different styles [16].[190]

Digital map developers are not alone in thinking of specific user interfaces for their media. Audio interfaces ([17],[191] [8][192])—why not? 3D interfaces ([5], [6][193])—why not? The sequence might go on.

---

[182] Best overview found: http://hcibib.org/hci-sites/GUIDELINES.html.

[183] W3C-based example: http://www.w3.org/TR/UAAG10/.

[184] http://www.webstyleguide.com/wsg3/index.html.

[185] https://www-01.ibm.com/software/ucd/designconcepts/designbasics.html.

[186] https://developer.apple.com/library/mac/navigation/.

[187] http://www.sap.com/corporate-en/index.epx.

[188] http://www.sapdesignguild.org/resources/ucd_paper.asp.

[189] For example, http://developer.android.com/guide/practices/ui_guidelines/index.html, http://wiki.eclipse.org/User_Interface_Guidelines.

[190] http://icaci.org/files/documents/ICC_proceedings/ICC2009/html/nonref/20_6.pdf.

[191] http://ncomprod.nokia.com/library/files/docs/A_Spatial_Audio_User_Interface_for_Generating_Music_Playlists.pdf.

[192] http://ldt.stanford.edu/~ejbailey/02_FALL/ED_147X/Readings/CohenExcerpt.Winograd.pdf.

[193] http://www.ijvr.org/issues/issue2/02!ijvr_bowman_formatted.pdf.

## 2.4 In Lieu of a Conclusion

The standards parade is over. Space has been overstretched without really bringing the topic to an end. Standards are important, there are many of them, they are more or less stringent, some solid acquaintance with them helps. This may appear as a truism, but it is particularly true for mashup developers and users.

Observing standards, specifications, and guidelines that standardize/populate/ underpin the web has surprised the author often enough. What standard counts, what a standard wilderness! The chapter had to tame this area to traceability in order to bring it home to its audience. If to some readers the standards parade has brought concrete hints for own system design—much the better.

Almost all readers will have objections against how the standards were chosen, left out or described in this chapter. The author shares many of them.

As long as the web keeps expanding, more and new regulation is expected to come up, and more existing standards are available for reuse. Mashup users and developers need standards, but they should ask which ones they need for their own purposes.

## References


1. Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider P (eds) (2007) The description logic handbook. Cambridge University Press, Cambridge
2. Bevan N (2001) International standards for HCI and usability. Int J Hum-Comput Stud 55:533–552
3. Bikakis N, Tsinaraki C, Gioldasis N, Stavrakantonakis I, Christodoulakis S (2012) Semantic hyper/multi-media adaptation: schemes and applications. Springer, Berlin
4. Bizer C, Lehmann J, Kobilarov G (2009) DBpedia—a crystallization point for the web of data. J Web Semant 7(3):154–165
5. Bowman DA, Kruijff E, Laviola JJ (2004) 3D user interfaces: theory and practice. Addison-Wesley Longman, Reading
6. Bowman DA, Chen J, Wingrave CA, Lucas J, Ray A, Polys NF, Li Q, Haciahmetoglu Y, Kim JS, Kim S, Boehringer R, Ni T (2006) New directions in 3D user interfaces. Int J Virtual Real 5(2):3–14
7. Carroll JM (2000) Making use. Scenario-based design of human–computer interfaces. MIT Press, Cambridge
8. Cohen MH, Giangola JP, Balogh J (2004) Voice user interface design. Addison-Wesley, Reading
9. Comer D (2006) Internetworking with TCP/IP. Volume 1: Principles protocols, and architecture, 5th edn
10. Cooper A, Reimann R, Cronin D (2007) Face3. The essentials of interaction design. Wiley, New York
11. Fielding RT, Taylor RN (2002) Principled design of the modern web architecture. ACM Trans Internet Technol 2(2):115–150
12. Guinard D, Floerkemeier C, Sarma S (2011) Cloud computing, REST and mashups to simplify RFID application development and deployment. In: WoT 2001
13. Guinard D, Ilon I, Mayer S (2011) Search of an internet of things service architecture: REST or WS-*? A developers' perspective. http://www.vs.inf.ethz.ch/publ/papers/dguinard-rest-vs-ws.pdf

14. Heath T, Bizer C (2011) Linked data: evolving the web into a global data space. Morgan and Claypool
15. Heath T, Hepp M, Bizer C (eds) (2009) Special issue on linked data. Int J Semant Web Inf Syst 5(3)
16. Heil B, Reichenbacher T (2009) The use of guidelines to obtain usability for geographic user interfaces. In: ICC_proceedings. http://www.zora.uzh.ch/29900/
17. Hiipakka J, Lorho G (2003) Spatial audio interface for generating music playlists. In: Int Conf on Auditory Display. http://library.graphicon.ru/pubbin/view_paper.pl?paper_id=932
18. Holtkamp H (1997) Introduction to TCP/IP. Univ Bielefeld. http://www.rvs.uni-bielefeld.de/~heiko/tcpip/tcpip.pdf
19. ISO/IEC (2011) Directives. Part 2: Rules for the structure and drafting of international standards, 6th edn
20. Kern C, Kesavan A, Daswani N (2007) Foundations of security: what every programmer needs to know. Apress, New York
21. Kreibich JA (2010) Using SQLite. O'Reilly Media, Sebastopol
22. Lau Rynson WH, Li F, Kunii TL, Guo G, Zang B, Magnenat-Thalmann N, Ksirsagar S, Thalmann D, Gutierrez M (2003) Emerging web graphics standards and technologies. IEEE Comput Graph Appl 23(1):66–75
23. Lechner M (2011) A proposed step-by-step guide to an AR standard. http://www.perey.com/ARStandards/[Wikitude]ARMLSWG_Background.pdf
24. Lechner M, Tripp M (2010) ARML—an augmented reality standard. http://www.perey.com/MobileARSummit/Mobilizy-ARML.pdf
25. Musmann HG (2006) Genesis of the MP3 audio coding standard. IEEE Trans Consum Electron 52(3):1043–1049
26. Noll P (2000) MPEG digital audio coding standards. CRC Press, Boca Raton
27. Norman DA, Draper SW (1986) User-centered system design. Erlbaum, Hillsdale
28. Owens M, Allen G (2010) The definitive guide to SQLite. Apress, New York
29. Perey C, Engelke T, Reed C (2011) Current status of standards for augmented reality. http://www.perey.com/Standards_for_Augmented_Reality_March_28.pdf
30. Raissi R (2002) The theory behind MP3. MP3' Tech
31. Richardson L, Ruby S (2007) RESTful web services. Web services for the real world. O'Reilly Media, Sebastopol
32. Saint-Andre P, Smith K, Tron R (2009) XMPP: the definitive guide. O'Reilly Media, Sebastopol
33. Scott B, Neil T (2009) Designing web interfaces. O'Reilly Media, Sebastopol

# Chapter 3
# Mashups for Web Search Engines

**Ioannis Papadakis and Ioannis Apostolatos**

**Abstract** Current practice in looking for information on the web states that searchers rely on large-scale web search engines to get assistance. The quality of the search results is analogous to the ability of the searchers to accurately express their information needs as keywords in the search engine's input box. In this chapter, an attempt is made to explore the various efforts that have been made regarding the query construction/refinement phase of a search session on the web. Along these lines, a number of cases are presented that are based on intuitively created mashups for the underlying web search engine. Particular attention is given to two query construction/refinement mashups that integrate various DBpedia datasets with the web search engine provided by Google.

## 3.1 Introduction

The crucial role of search engines in locating meaningful information on the web is justified from the fact that currently, three out of the top-5 sites on the web (according to Alexa[1]) are search engines. However, despite their popularity, such services do not always succeed in retrieving the right results for their users. Sometimes, queries suffer from intrinsic features of natural languages such as ambiguity and synonymy of words. Things get even worse when a top-ranked resource squeezes out from the search results list other, less popular resources with the same name. Thus, it appears that search engines should not only pay attention to efficiently

[1] http://alexa.com/topsites, accessed: 29.6.2012.

I. Papadakis (✉)
Department of Archives and Library Science, Ionian University, Ioannou Theotoki 72, Corfu, 49100, Greece
e-mail: papadakis@ionio.gr

I. Apostolatos
Department of Informatics, University of Piraeus, Karaoli and Dimitriou 80, Piraeus, 18534, Greece
e-mail: japost@unipi.gr

B. Endres-Niggemeyer (ed.), *Semantic Mashups*,
DOI 10.1007/978-3-642-36403-7_3, © Springer-Verlag Berlin Heidelberg 2013

ranking their results, but also to provide the necessary tools that will enable their users issuing the right queries.

In this chapter we explore mashups for the query construction/refinement phase of web searches. Such mashups often refer to the collaborative knowledge accumulated in Wikipedia. As will be shown in this work, query construction/refinement mashups for web search engines have been reported prior to the advent of linked data.

This chapter presents in detail two query construction/refinement mashups that are based on DBpedia [4] and targeted towards the users of major search engines on the web. More specifically, the first approach takes advantage of a centralized mashup that is based on various DBpedia datasets that are integrated into a relational database schema. Users may interact with the provided GUI to issue queries to the underlying database that will, in turn, respond with the corresponding information. Such information is represented as recommendations for constructing and/or refining an initial query. The second approach takes advantage of a contextual mashup that is dynamically created each time a search session is initiated. Users are able to select words from the mashup and formulate a query that is ultimately redirected to Google. Both mashups explore ways to integrate large-scale web search engines with linked data to produce semantically rich queries.

The rest of this chapter is structured as follows: the next section describes the different information seeking modes a searcher may exhibit during a search session. Section 3.3 discusses the problems that web searchers encounter when they try to express their information needs to a search engine. Section 3.4 describes the various approaches that have been followed by search engines over time to aid their users in providing meaningful queries. Section 3.5 outlines some mashups that are targeted towards the query construction and refinement phase of a search session. The following section is devoted to two query construction/refinement mashups that are based on linked data and especially DBpedia. Both of them are discussed in detail. Finally, in the last section, among others is concluded that DBpedia-based mashups are excellent candidates for integration with web search engines that provide programmable access to their search box.

## 3.2 Web Querying Behaviors

According to the sense-making model for information seeking [14], searchers find themselves in various information seeking situations when looking for information. Such situations reflect their information needs, defined as the perceived needs for information that lead to using an information retrieval system [28]. Thus, it is the information retrieval system's responsibility to bridge the knowledge gap between the searcher's needs and the actual resources that satisfy such needs.

Web search engines have to overcome many problems to succeed in providing searchers with the right information. Thus, a major issue they have to deal with is the fact that average searchers do not exhibit a standard behavior during their information seeking process. According to the corresponding literature [7, 21], there are various modes a searcher may adopt when participating in such a process. Most recently, Spencer [30] identified four intersecting modes for information seeking.

- The 'known item' information seeking mode, in which searchers know exactly what they want and also what terms to use to verbalize it. This particular mode is straightforward and easy to comprehend. In this mode, web search engines experience various difficulties imposed mainly by the very nature of natural languages. As will be discussed later in this chapter, the way a search query is formulated provides insights not only about the searchers' understanding of the problem, but also of the information necessary to address it [35].
- The 'exploratory' mode for information seeking states that searchers have a general idea of what they need to know but they are not sure how to express it as a set of terms for the search query. Usually, they recognize the right information when they come across it, but they also may not know whether they have found enough information. Search engines have to deal with the fact that searchers are not able to properly formulate queries, since they do not know how to phrase such queries [8]. As the searchers cannot express what they are looking for within the environment of the search engine, they have to perform an initial search. They have to exhaustively run through the search results in order to familiarize themselves with the domain they are after, and eventually get some ideas for relevant terms and their synonyms [8]. This is a rather tedious process that involves running through a lot of useless information to find comparatively smaller pieces of useful information.
- The "don't know what I need to know" information seeking mode refers to searchers that may think they need one thing but they actually need another. Or, they may be looking at an information resource without a specific goal in mind. This mode consists of searches occurring in a complex and/or unknown domain (e.g. legal, medical, financial) as well as searches addressing the need of keeping up-to-date. As stated in [23], searchers are asked to confront the paradox of describing something they do not know without any help from the search engine. Especially when it comes down to major web search engines, searchers are offered regrettably few options to reformulate their initial query.
- Finally, the 're-finding' mode addresses information seeking situations where searchers are looking for things they have already seen. From the search engine's point of view, this mode is addressed by personalized search, where users have to sign-in to the personalized search engine [32]. There is also the option of bookmarking the search query, but since bookmarking can be achieved away from the search engine, it will not concern us further in this work.

## 3.3 Issuing Effective Queries to Search Engines

In their quest for satisfying their information needs through the employment of a search engine, users participate in search sessions. A search session initiates when a user starts expressing his information needs according to the specific dialect of the search engine (commonly, by typing some keywords to the appropriate search box) and concludes when a user stops interacting with the search engine. From a certain point of view, a search session is composed of two phases, namely the query construction and query refinement phases. The query construction phase refers to the attempt of a user to articulate the initial query that will bring the first results from the search engine whereas the query refinement phase refers to any subsequent attempts to 'fine tune' the query that has been initially issued to the underlying search engine, to improve the quality of the corresponding search results.

The query construction phase of a search session is crucial to the fulfillment of the information needs of the searcher. During the query construction phase, a searcher has to adopt his information needs to the specific dialect of the underlying search engine. However, human languages have certain features that should also be taken under consideration during the query construction phase. Thus, queries expressed as consecutive terms suffer from the possible polysemy of the corresponding words. Polysemy occurs when a word has more than one sense [22]. Consequently, a query consisting of an ambiguous word without further information that correctly disambiguates such word, may result in a search results list with completely useless information. Current practice in major web search engines' ranking tactics states that the risk of cluttering the search results list with useless resources is greater when searchers are seeking for the less popular meaning of a polysemous word.

Provided that a query is correctly disambiguated, it could be further refined to provide search results of better quality to its users. The query refinement phase of a search session should take under consideration another important feature of natural languages: synonymy of words. Synonymy occurs when two or more words share the same meaning [22]. Thus, two web pages using two different words to express the same concept are indexed differently by the search engine. The magnitude of synonymy's influence to search engines can be further realized by taking under consideration the fact that the probability of two persons using the same term in describing the same thing is less than 20 % [6].

As users depend more and more on the web for finding information, it becomes more crucial that search engines assist them to set up their varying queries in a manner that the engine can distinguish them and interpret them correctly.

## 3.4 Aiding Searchers During Their Search Session

To address the difficulties that web users encounter while searching for information about a topic of interest, a number of techniques have emerged such as clustering, human-powered search engines, personalization, relevance feedback, etc. Such techniques will be discussed in the following sections.

### 3.4.1 Clustering

Perhaps the most commonly known clustering search engine on the web is Yippy.[2] Yippy is a prevailing meta-search engine that adds a sidebar containing clusters (labeled as 'clouds') next to the search results list. Each cluster corresponds to a topic and contains one or more items appearing to the search results list. Thus, searchers are able to filter out results belonging to a specific topic. Such clusters derive from the short descriptions that accompany each item returned from the underlying search engines. Due to the automated nature of the service, Yippy finds it difficult to provide semantically distinguished labels corresponding to the actual content of the clusters. For example, consider the query 'semantics', which yields semantically overlapping labels for many clusters.

### 3.4.2 Human-Powered Search Engines

The phrase 'human-powered' refers to a search engine which has its results list affected by human intervention, usually by people rating individual results further up or further down [12]. The rationale behind human-powered search engines is, as a bit exaggeratively mentioned in [33], the fact that machines are excellent in executing code very fast but they have no real intelligence, they do not share, they cannot judge, they have no appreciation. So, a search engine that enables its users to determine the position of an information resource in a search results list (capturing this way 'the wisdom of the crowd' [33]) will always be better than the most algorithmically efficient web search engine.

In this line of thoughts, a number of human-powered search engines have emerged. Some of them rely exclusively on their users to build search results lists (e.g. stumpedia[3]), but most of them do not try to 'reinvent the wheel', in the sense that they behave as hybrid search engines, applying human intervention to re-rank results that are initially machine-generated. For example, Anoox[4] and iRazoo[5] depend on a voting system on affecting the position of an information resource within a search results list. According to the founders of such systems, people's opinion always outweighs machine's algorithms.

Another kind of human-powered search engines (also called social search engines) tries to improve its quality by applying social networking logic to the underlying workflow. For example, MySidekick[6] is a human-powered search engine that allows people to find and submit information resources that are then automatically tagged with terms used during the search session. These tagged pages are then

[2] http://yippy.com.

[3] http://www.stumpedia.com.

[4] http://www.anoox.com.

[5] http://www.irazoo.com.

[6] http://www.mysidekick.com.

anonymously shared within the MySidekick community to provide better results for all. Another effort, based on the concept of Wikipedia, wikia[7] (founded by Jimmy Wales, the founder of Wikipedia) allows users to edit, annotate, comment, delete and expand the search results.

As a common ground, it is argued that the above solutions are mainly targeted to the problem of providing better quality of search results by improving the ranking and/or the topical context of the retrieved information resources. Moreover, human-powered search engines suffer from the fact that they are still just as easy to 'game' as more traditional engines [12] and from the fact that they have to persuade their users to provide feedback (implicit or explicit) to succeed.

### *3.4.3 Relevance Feedback*

According to relevance feedback, queries are reformulated based on previously retrieved relevant and not relevant information resources [27]. This technique provides a controlled query alteration process designed to emphasize some terms and to deemphasize others, as required in particular search environments. However, such a technique cannot be easily applied to large-scale web search engines, where authentication is difficult to impose and diversity prevails. Moreover, explicitly asking searchers to contribute to the overall information seeking process simply will not work. The average web user employing a search engine has no motivation in spending time this way [12]. On the contrary, financially motivated stakeholders could take advantage of the impact of relevance feedback to the search engine to promote business web sites higher in the search results lists. Even in the case of automatic (blind) relevance feedback, where terms from the top few information resources returned are automatically fed back into the query [10], success is by no means self-evident. This particular technique may possibly be effective in the 'known-item' information seeking mode (see Sect. 3.2), where web search engines in any case seem to perform well.

### *3.4.4 Ontology-Driven Approaches*

Ontologies can either derive from the document collection at hand or from other sources [16]. However, integrating ontologies with web search engines has proved to be less successful in the past [20]. This is due to the fact that (a) domain ontologies are expensive to produce; (b) they are only available for a small proportion of document sets, and (c) they appeal only to expert users. Even in the case where generic ontologies are employed, shortcomings in any specific technical domain have been identified [23]. However, attempts in large-scale linguistic-oriented ontologies on the web like EDR [37], CYC [18] and WordNet [22], together with the

[7] http://search.wikia.com.

evolution of the web as a collaborative environment where contributions of any kind are greatly endorsed by the overall web community, may turn things around, as far as ontology-powered query formulation is concerned.

As will be shown later in this chapter, the advent of linked data poses another argument in favor employing ontologies and/or other external sources of information to the query construction/refinement phase of a search session.

### *3.4.5 Search Personalization*

Search personalization is the process of incorporating information about the user needs in the query processing phase. One approach to personalization is to have users describe their general search interests, which are stored as personal profiles [26]. Recent search personalization approaches on the web involve integration with some kind of external semantic structure to identify the context of each search session [2, 13, 29]. More specifically, the authors of [2] describe a profile representation using Internet domain features extracted from URLs. In [29], an effort is made to model the user context as an ontological profile by assigning implicitly derived interest scores to existing concepts deriving from the Open Directory Project (ODP) ontology.[8] Another search personalization technique based on the ODP ontology is defined in [13]. More specifically, a user profile is built by accumulating graph-based query profiles in the same search session. In contrast to [2], the user profile is represented as a graph of the most relevant concepts of an ontology in a specific search session and not as an instance of the entire ontology.

### *3.4.6 Google's Approach*

During the past few years, major web search engines and especially Google, which seems to be the most popular one,[9] have evolved their provided functionality. Although the mechanics of their approaches has not officially been published, it is evident that some of the above techniques (i.e. search personalization, relevance feedback, human intervention in ranking) are finding their way into the provided searching process.

More specifically, Google's "+1"[10] (successor of discontinued 'Stars' and 'SearchWiki') approach takes advantage of Google Account authentication services to identify the searchers and consequently log their personal search tactics. Such information is also available to each user as his search history. Moreover, quite recently, Google introduced the 'knowledge graph' in an attempt to provide disambiguation functionality and crowd-sourcing recommendations to its users through the employment of an underlying graph that is based on collaborative knowledge.

---

[8] http://www.dmoz.org.

[9] http://www.alexa.com, accessed: 29.6.2012.

[10] http://www.google.com/psearch.

As far as the query construction phase of a search session is concerned, major web search engines have made considerable progress. Auto-suggest functionality within the search box is currently provided by default; according to Google's approach, upon issuing a query, a list of query suggestions is displayed. Consequently, users can rapidly express their initial query by selecting and promoting the suggestion that best suits their information needs. Moreover, Google provides the option to fetch search results while users type their query (i.e. Google Instant[11]).

## 3.5 Integrating Search Engines with the Semantic Web: Mashup-Based Approaches

In the context of web search, mashups could be described as the efforts that are targeted towards the provision of extended functionality to the traditional paradigm of a search engine through the combination of data deriving from multiple sources. Such functionality is often realized as innovative query construction/refinement services aiming at enhancing users' search experience.

The idea of enhancing the performance of query construction/refinement methods during a search by creating mashups using external resources is not new. As a matter of fact, many works suggest the utilization of large corpora [15] or ontologies [9] for making rich integrations between user queries and document collections in the hope of improving the quality of search results. Recently, researchers suggested that Wikipedia may serve as the external resource to support query construction and refinement. In this context, Wikipedia-based querying methods have been reported in [17, 19, 23]. Here, we briefly discuss the most important mashups and underline their methodology and effectiveness.

Thus, in [19] a query refinement method based on Wikipedia articles is introduced, which aims at improving retrieval effectiveness of poorly articulated queries for which automatic relevance feedback is not helpful. The authors in [19] have created their own machine-readable version of Wikipedia, which they employ to expand the initial queries. The corresponding evaluation against the TREC Robust Track [34] revealed that the Wikipedia corpus provides significant expansions to the initial query, as compared to automatic relevance feedback, provided of course that the topic of the initial query exists within the Wikipedia corpus. Finally, it should be noted that the work in [19] is intended for use in controlled repositories, not for integration with web search engines where important issues arise [5].

In the work of [23], Koru is introduced as a mashup consisting of a digital library and a query construction/refinement interface that is based on Wikipedia (Fig. 3.1).

According to Koru's query topics panel, a typical search box receives the searcher's query. The provided functionality dismantles the query to a list of related

[11] http://www.google.com/instant.

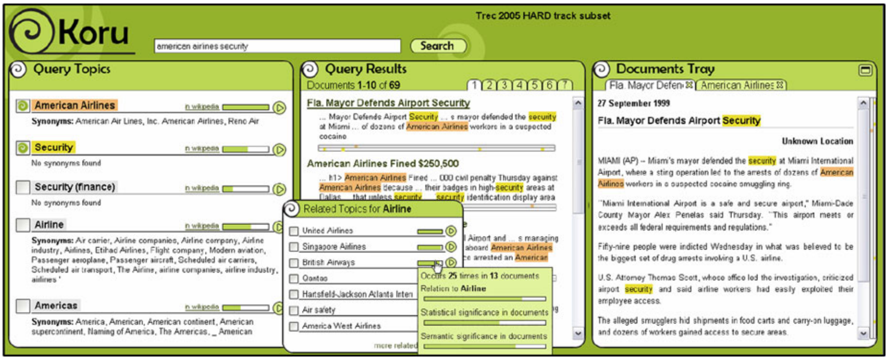



**Fig. 3.1** Koru UI (figure is taken from [23])

topics, as they have been identified in the underlying thesaurus. Such a list also contains possible disambiguations of the containing topics according to the thesaurus. The list is ranked according to the likelihood that each contained topic is a relevant, significant topic for the current query. Such relevance is measured in terms of statistical and semantic significance of the topics within the underlying document collection. Searchers are able to fine-tune their query by manually excluding and/or including topics from their query in an intuitive, user-friendly manner. Each topic is also accompanied with a paragraph containing possible synonyms. Such information does not explicitly participate in the query. The purpose is to help searchers understand the topic. Moreover, searchers are able to progressively expand their initial knowledge concerning their information needs by interacting with the interface. Such interaction is realized through the employment of non-movable pop-up windows containing topics that are related to the topic they selected from the previous list. In the case users find significant information during their interactions, they can incorporate the corresponding terms within the query by selecting them. Similar to [19], the authors of [23] have created their own machine-readable version of Wikipedia. Koru is a well-suited approach for controlled collections such as digital libraries, but practically unsuitable for web search engines; the enormous size of the information resources within web search engines together with their highly dynamic and complex nature demand for completely decoupling the query interface from the underlying information resources to avoid scalability and performance problems [5]. This is not the case in Koru, where query terms are ranked according to their relevance to the document collection. Additionally, Koru's GUI introduces new metaphors and thus entails a high learning curve to its users. According to the inductive user interfaces technology proposed by "Microsoft Inductive User Interface Guidelines",[12] learning curves on the web must necessarily be quick. Koru's provided functionality incorporates gauges, multiple sliding panels, pop-ups that dominate the screen and require a certain degree of familiarity from its users, which is by no means self-evident.

[12] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwui/html/iuiguidelines.asp.

Quite recently, Tweetnews[13] emerged as a mashup that integrates search engines with the popular social media service Twitter. Tweetnews was created in 2009 by Vik Singh, the architect of Yahoo! Search BOSS API.[14] Essentially, Tweetnews uses this API to bring Yahoo's news search results[15] and reorders them based on how similar they are to the emerging topics found on Twitter for the same query. The results that the API returns, many times are irrelevant with the user's original query. This happens because some news items come higher in rank since they are ordered only by timestamp.

## 3.6 Integrating Web Search Engines with the Semantic Web: DBpedia-Based Mashups

The advent of Linked Open Data—LOD and especially DBpedia—facilitated the creation of many mashups in various disciplines, especially in the field of web engineering [1]. LOD is essentially the building block of the so-called 'Web of Data' in which anyone can publish information, link to existing information, follow links to related information, and consume and aggregate information without necessarily having to fully understand its schema nor to learn how to use a proprietary Web API [11]. The flagship of LOD is DBpedia, which contains structured content from the information created as part of the Wikipedia project. In August 2012, DBpedia contained descriptions of more than 3.5 million things, many of which are also available in 18 languages.[16]

However, when it comes down to web search, LOD-mashups are not so easy to find. In this section, two mashups are presented that are both targeted towards the exploitation of crowd-sourced information provided by DBpedia, in favor of web search engines. As it will be shown later, the two approaches converge in their ultimate goal and diverge in their technical solutions.

The first mashup [24, 25] replicates specific DBpedia datasets in a traditional relational database schema and integrates such data with the search box of two major web search engines, namely Google and Yahoo!, in favor of a query construction/refinement service for their users. The service is based on a novel GUI that manages to hide the inherent complexity of semantic web terminology from its end-users.

GContext [3], the second mashup, facilitates a DBpedia-based query construction/refinement service for Google. GContext follows a decentralized approach. It creates appropriate SPARQL queries to DBpedia's endpoint[17] and thus recommends its users how to construct or refine their initial query.

[13] http://tweetnews.appspot.com.

[14] http://developer.yahoo.com/search/boss/.

[15] http://news.yahoo.com/.

[16] DBpedia ver. 3.8: http://blog.dbpedia.org/2012/08/06/dbpedia-38-released-including-enlarged-ontology-and-additional-localized-versions/.

[17] http://dbpedia.org/sparql.

The following section outlines the advantages of integrating linked data with web search engines, particularly during the query construction and refinement phase of a search session.

### *3.6.1 Integrating LOD with Web Search Engines*

The aforementioned mashups underpin query construction/refinement services on top of web search engines in order to help information seekers formulating queries that express their search intentions well. This way, search engines can propose queries which adapt to the user needs. Such services are particularly helpful when users are unable to formulate accurate and specific queries (e.g. "don't know what I need to know" mode of information seeking).

One thing that makes search engines so popular is that they enable users to query the web in an intuitive yet simple manner, i.e. by submitting a few keywords to the engine's search box. Despite the intended simplicity associated with querying the web via a web search engine, web searchers may spend too much time reformulating queries, without being able to satisfy their information needs. Search engines provide little help to users with vague knowledge of the terminology employed within relevant documents. Even if searchers succeed in locating the information sought, they often realize that their successful queries differ significantly from their initial query.

The motive behind the aforementioned mashups is to bridge the semantic gap between the initial query and the ideal query that the search engine expects from a user provided that he knows in which terms pertinent documents answer his question.

Both of the query construction/refinement services extend the functionality of the traditional search box and act as an intermediate layer between searchers and web search engines.

The following section presents the first mashup, which follows a centralized approach in providing DBpedia-based information to the users of a search engine during their search sessions.

### *3.6.2 A DBpedia-Based, Centralized Mashup for Web Search Engines*

#### **3.6.2.1 Description**

In this section, a mashup is presented, which assists searchers to formalize their query and pick the most suitable terms. The mashup exploits crowd-sourced Wikipedia knowledge that is available via various DBpedia's datasets. Nevertheless, the described approach can be easily extended to incorporate other LOD such as YAGO [31]. Serving as a LOD-based mashup, the described approach provides an interactive GUI that seamlessly integrates the knowledge provided by web users

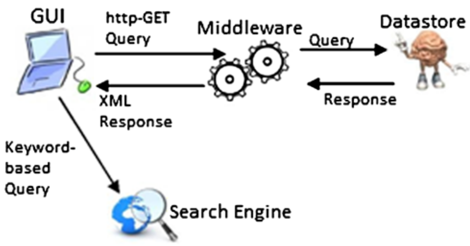



**Fig. 3.2** Architecture of DBpedia-based, centralized mashup

with web search engines. Such knowledge is modeled in a carefully designed, modular conceptual schema that supports querying against a large volume of linked data.

The underlying schema has been developed by making use of N3-formated[18] datasets provided by DBpedia. The details of the schema construction are given in Sect. 3.6.2.2.

Searchers are able to express their information needs by interacting with an accordingly designed LOD-browsing GUI. More specifically, interactions through the GUI are converted to query/response messages that are administered by a middleware (Fig. 3.2).

Queries are encapsulated in HTTP-GET requests and responses are expressed as XML-based strings. The middleware issues the queries to the underlying datastore, which, in turn, delivers the appropriate responses to the middleware. Such responses are converted to XML and channeled back to the GUI.

Finally, the GUI transforms responses to keyword-based queries and issues them to the underlying web search engine. A detailed description of the approach is provided in the following sections.

#### 3.6.2.2 A Wikipedia-Based Schema

As previously mentioned, the described system stores linked data originating from DBpedia into a datastore that is based on a conceptual schema. Several studies exist that rely upon DBpedia datasets for building highly expressive ontologies via the combination of Wikipedia and WordNet.[19] Two of the most widely known resources that have emerged from such efforts are the Kylin Ontology Generator (KOG) [36] and the YAGO ontology [31]. The above studies motivated researchers to build query construction/refinement services that mediate between searchers and web search engines with the help of knowledge drawn from various DBpedia datasets.

The service utilizes the following DBpedia datasets:[20] (i) the Wikipedia articles, (ii) the list of disambiguations that Wikipedia encodes for connecting generic articles to their specific interpretations, (iii) the categories under which the Wikipedia

[18]N3 notation (Berners-Lee T. Notation 3 (N3): a readable RDF syntax. http://www.w3.org/DesignIssues/Notation3.html).

[19]http://wordnet.princeton.edu.

[20]DBpedia's dataset dumps are readily available for downloading from: http://wiki.dbpedia.org/Downloads32.

**Table 3.1** Statistics of the DBpedia datasets

| Datasets | No. of items |
|---|---|
| Wikipedia articles | 2,866,994 |
| Disambiguation entries | 226,978 |
| Categories entries | 339,112 |
| WordNet classes | 124 |
| Articles linked to WordNet classes | 497,797 |
| Infobox records | 19,230,789 |

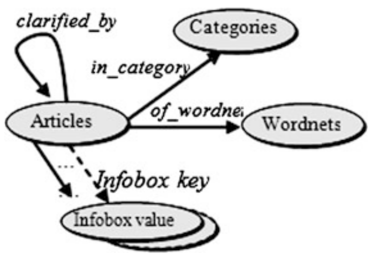



**Fig. 3.3** Conceptual schema

articles are classified, (iv) the WordNet classes to which Wikipedia articles correspond, and (v) the articles' infoboxes that contain semantically rich properties about the considered articles. Table 3.1 summarizes the statistics of the DBpedia datasets that are employed in this mashup.

These datasets are organized in a conceptual schema as Fig. 3.3 illustrates. The classes of the schema are: (i) *Articles* that correspond to the Wikipedia articles organized as class instances, (ii) *Categories* that correspond to the appropriate categories of the Wikipedia articles, (iii) *WordNets* that store the type of every Wikipedia article.

Moreover, the relations of the schema are: (i) "*clarified by*", a reflexive relation corresponding to the disambiguation pages of Wikipedia articles, which has the class 'Articles' as both domain and range, (ii) "*in category*" that connects every article to one or more appropriate categories. Another relation is (iii) "*of WordNet*" that connects articles associated with WordNet classes to an appropriate entity type. Finally, the Wikipedia infobox properties are key-value pairs that are expressed as datatype properties of their corresponding article instances.

Based on the above schema, a datastore is created that is incorporated into the query construction/refinement service in the hope of assisting searchers decipher the semantic orientations of their candidate queries before these are actually issued to the search engine. The datastore is serialized as a MySQL database, taking this way advantage of its fast indexing capabilities.

Given the highly dynamic nature of crowd-sourced knowledge on the web, the mashup is designed in such a way that it can be easily extended with existing and yet-to-appear datasets. Specifically, in case of an incoming dataset, a new class would be added to the schema (see Fig. 3.3) together with its corresponding relation.

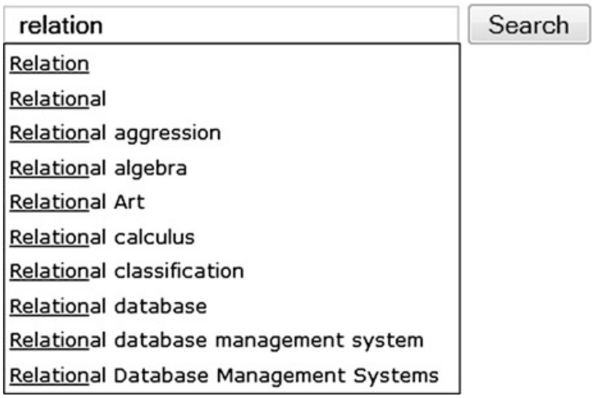



**Fig. 3.4** Automatic suggestions for query construction

Moreover, a new query-response pair would be defined together with the corresponding rendering of the response from the client-side GUI, as will be explained later in this chapter.

Next, the GUI and the middleware of the query construction/refinement service are accordingly illustrated through several examples.

#### 3.6.2.3 GUI and Middleware for Structuring/Refining Queries

In this section, the discussion turns to the GUI. It extends the traditional input box of a web search engine by (a) suggesting context-aware query formulations based on the first letters in the input box, and (b) visualizing the semantics of the initial query (Figs. 3.5a, 3.5b, 3.6, 3.7, 3.8). The design principles require that the GUI should be interactive, inductive, easy to use and fast to execute. Having such requirements in mind, the auto-suggest input box illustrated in Fig. 3.4 is presented.

As in web search engines like Google and Yahoo!, the search box accepts the input of the user and responds with a set of alternative query wordings. In the event of a few characters typed in, the box suggests a number of strings that can be attached to them. The auto-complete suggestions are leveraged from the titles of Wikipedia articles. The searcher can employ any of the suggested query alternatives, or ignore them and search with his own keywords.

Up to this point, the described functionality replaces the auto-suggest functionality that has been recently added to major web search engines such as Google and Yahoo! with auto-suggest functionality powered by Wikipedia.

If the searcher decides to select one of the offered suggestions, an HTTP-GET query is issued to the middleware, which, in turn, issues appropriate queries to the underlying datastore. Since the datastore is serialized in MySQL, such queries are transformed to SQL-select statements. The results of each statement are encoded by the middleware in XML-based strings that are routed back to the GUI. The GUI visualizes the responses as interconnected boxes located above the search engine's input box. Each box has a title corresponding to an article from Wikipedia and a

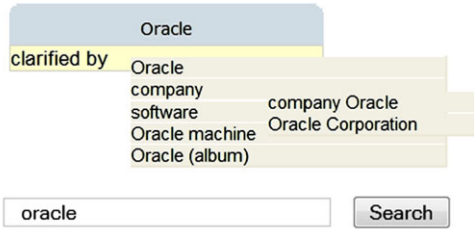



**Fig. 3.5a** Provided query disambiguations

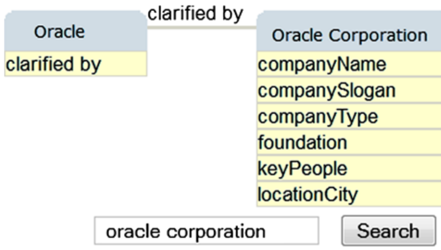



**Fig. 3.5b** Selected query disambiguations

number of labels beneath it pertaining to the article's possible semantic relations to other elements. Searchers are able to interact with the boxes by clicking on a relation. In that case, the initial query is refined. As stated earlier in this chapter, there are currently four different types of relations (i.e. disambiguations, categories, WordNet-classes and infoboxes) implemented, although the modularity of the mashup allows for further expansion with more relations.

#### 3.6.2.3.1 Disambiguations/WordNet Classes

If the searcher clicks on a "clarified by" relation, query disambiguation is performed as follows: at first, the searcher is presented with a list of all the corresponding disambiguations that match his selected suggestion (Fig. 3.5a). Such disambiguations could be grouped by WordNet classes, provided they share common WordNet meaning. In such a case, upon selecting the corresponding WordNet label, a second-level disambiguation list appears. By selecting either one of the first- or second-level disambiguations, a new box containing the disambiguated entity is sketched at the right (Fig. 3.5b), which is connected to the previous box with a line labeled "clarified by". Simultaneously, a search query that consists of keywords deriving from the two box titles (elimination of duplicates is applied) is directed to the underlying search engine.

Under the hood, requests for disambiguations equal to HTTP-GET requests containing parameters determining (a) the id of the request, (b) the type of the request (i.e. disambiguation) and (c) the name of the Wikipedia article for which the disambiguations are requested. The corresponding response is an XML-based string containing the possible disambiguations as a set of <instance> elements. Below, the request: id = "q0", type = "disambiguation", name = "Ferrara" results to the following XML-string:

```
<reply type='success' iid='Ferrara'
  rid='disambiguation' id='q0'>
<instance id='Ferrara'>
<label lang='en'>Ferrara</label> </instance>
<instance id='Ferrara_Fire_Apparatus'>
<label lang='en'>Ferrara Fire Apparatus</label>
</instance></reply>
```

In a similar manner, WordNet requests equal to HTTP-GET requests containing parameters determining (a) the id of the request, (b) the type of the request (i.e. WordNet category) and (c) the name of the Wikipedia article for which the WordNet categories are requested. The corresponding response is an XML-based string containing the possible WordNet categories as <instance> elements. Below, the request: id = "q0", type = "in_wordnet_category", name = "Ferrara" results to the following XML-string:

```
<reply type='success' iid='Ferrara'
  rid='in_wordnet_category' id='q0'>
<instance id='city'><label lang='en'>city
</label></instance>
<instance id='monument'>
<label lang='en'>monument</label>
</instance></reply>
```

#### 3.6.2.3.2 Categories

If the searcher clicks on an "in category" relation, a pop-up menu appears containing the corresponding categories. Upon selecting one, a new box named after the selected category is sketched at the right, which is connected to the previous box with a line labeled "in category" (Fig. 3.6).

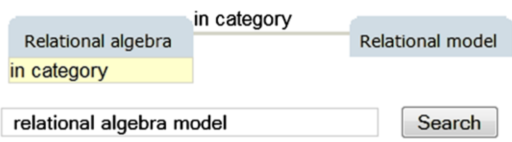



**Fig. 3.6** Selected category

Simultaneously, a query consisting of keywords from the two box titles is directed to the underlying search engine. Duplicates are eliminated.

A request for categories is realized as an HTTP-GET request containing parameters determining (a) the id of the request, (b) the type of the request (i.e. category) and (c) the name of the Wikipedia article for which the categories are requested. The corresponding response is an XML-based string containing the possible categories as <instance> elements. Below, the request: id = "q1", type = "in_category", name = "amade camal" results to the following XML-string:

```
<reply type='success' iid='Amade_Camal'
  rid='in_category' id='q1'>
<instance id='1956_births'><label lang='en'>
1956 births</label></instance>
<instance id='Living_people'><label lang='en'>
Living people</label></instance>
<instance id='Muslim_activists'><label lang='en'>
Muslim activists</label> </instance>
<instance id='Mozambican_politicians'>
<label lang='en'>Mozambican politicians
</label></instance></reply>
```

#### 3.6.2.3.3 Infoboxes

Finally, if the server's response consists of infobox properties realized as key-value pairs, the keys are displayed as labels. If the searcher clicks on a key, its corresponding value(s) appear(s) (Fig. 3.7). Upon selecting a value, a new box containing the selection is sketched, which is connected to the previous box with a line named after the infobox property's key.

Then, a search query featuring the keywords deriving from the two box titles (duplicate wipe out is applied) is send to the underlying search engine.

Each sketched box corresponds via its title to a part of the resulting query. The searcher controls the participation of each box in the search query by clicking on the checkbox that resides on top of each box (Fig. 3.8).

This way, the searcher is provided with information for determining and expressing the semantic orientation of his queries, before/while these are issued for search.

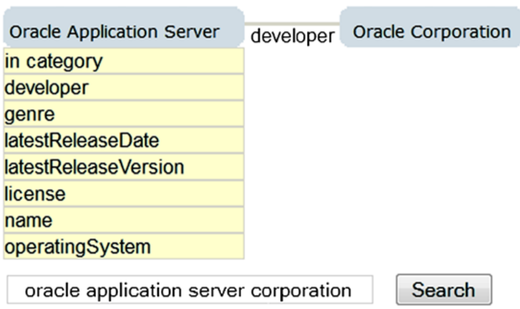



**Fig. 3.7** Infobox properties

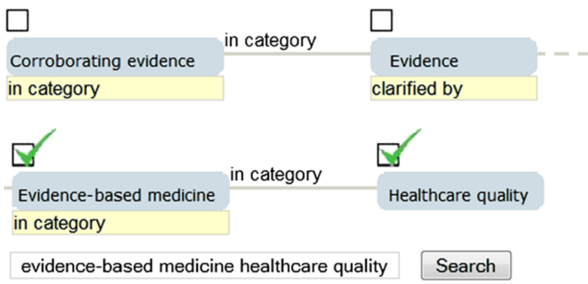



**Fig. 3.8** Selecting query terms

A request for infoboxes equals an HTTP-GET request containing parameters determining (a) the id of the request, (b) the type of the request (i.e. infobox) and (c) the name of the Wikipedia article for which the infoboxes are requested. The corresponding response is an XML-based string containing the possible key-value(s) pairs of the infobox as <dtprop> elements (the key of the infobox is the value of dtprop's attribute "id" and the value of the infobox is the value of the dtprop element). Below, the request: id = "q2", type = "infobox", name = "Zathras" results to the following XML-string:


```
<reply iid='Zathras' qtype='infobox' id='q2'
  type='success'>
<dtprop id='affiliation'>Great Machine </dtprop>
<dtprop id='finish'>War Without End</dtprop>
<dtprop id='name'>Zathras</dtprop>
<dtprop id='planet'>Unknown</dtprop>
<dtprop id='portrayer'>Tim Choate</dtprop>
<dtprop id='race'>Unknown</dtprop>
<dtprop id='start'>Babylon Squared </dtprop>
</reply>
```


#### 3.6.2.4 Discussion

As final notes, it should be stated that by employing the described mashup, searchers are instantly acquainted with query terms that otherwise would take them a lot of time to gather by exhaustively running through the search results of potentially vague queries. Thus, the corresponding service is particularly useful for the 'exploratory' and the "don't know what I need to know" information seeking modes that have been presented in Sect. 3.2.

Additionally, the provided GUI metaphors are smoothly integrated into the traditional search engine's GUI, since they occupy just a small portion of the screen on top of the input box, thus leaving plenty of room for the search results.

Furthermore, the simplicity of the underlying architecture not only renders the approach scalable to future enhancements with more semantically rich datasets, but also guaranties its rapid execution. The above features are very important for web search engines where time and space play a crucial role to their success.

The employment of common web widgets such as the auto-suggest box and interactive boxes as well as the absence of semantic web terminology from the GUI, renders the service fast to learn and easy to use.

Finally, it should be mentioned that if there is no available information about the user-typed terms, the overall search process does not break down and the query is transparently forwarded to the underlying search engine. Therefore, the worst case scenario is that searchers do not get any help from the service, but still their query is automatically submitted for search.

The query construction/refinement service has been integrated so far with two major web search engines (Google and Yahoo!) and can be accessed on-line.[21]

[21] Demo, available at: http://thalassa.ionio.gr/snh/entry/.

Thus, it is believed that the integration is doable for any search engine that gives programmable access to the input box.

The next section presents GContext, another mashup that provides query construction/refinement functionality to the users of web search engines. Although the origin of the underlying semantic data is common (i.e. DBpedia), this particular mashup follows a decentralized approach in providing the corresponding functionality. Finally, the GUI is largely based on textual metaphors such as lists.[22]

### *3.6.3 GContext: A DBpedia-Based, Decentralized Mashup for Web Search Engines*

GContext is another DBpedia-based mashup intending to provide query construction/refinement functionality to web search engines. More specifically, GContext dynamically suggests disambiguations during the query construction phase of a search session. Moreover, the service provides semantic refinements to the resulting query. The suggestions and/or the refinements originate from Wikipedia and are made available through DBpedia and other Linked Open Data—LOD providers. The GUI of the service aids searchers in selecting the appropriate context words that will ultimately reach Google, the search engine that lies beneath the service. The described approach explores ways to integrate web search engines with linked data to produce semantically rich queries. The service is presented through a number of case studies that correspond to real-world situations in information seeking on the web.

#### 3.6.3.1 Overview

The query construction/refinement service that is implemented through GContext is a two-step process: Initially, it provides auto-suggest functionality by reacting to the corresponding keystrokes of a searcher. Prefix search is performed to an index that comprises words and/or phrases originating from Wikipedia and made available through DBpedia. More specifically, the underlying index contains Wikipedia's titles. Then, upon selection of a suggestion, the information seeker is offered the chance to refine the initial query.

Every interaction (i.e. suggestion selection and/or query refinements) results in the construction of an appropriate query that reaches the underlying search engine, which, in turn, retrieves the corresponding search results. The underlying search engine is Google's Custom Search,[23] parameterized in a way that issues queries to the entire web site population.

The query construction/refinement service treats Google as a 'black box', which responds with search results to the corresponding queries. Thus, the principles that

[22]For more details about the specific mashup, the reader is prompted to read [25].

[23]http://www.google.com/cse/.


**Fig. 3.9** Query replacements

govern the described approach could be applied to any search engine that allows programmable access to its search box.

#### 3.6.3.2 Provided Functionality

As stated earlier, users start typing the first letters of the words they believe that best describe their information needs and receive suggestions that match against Wikipedia's titles. Such functionality facilitates query disambiguations, since Wikipedia's disambiguations follow a pattern (i.e. <ambiguous word> (disambiguation info)) that is promoted by prefix search.

Then, upon selection of a suggestion, SPARQL queries are directed to DBpedia's endpoint[24] in order to retrieve the suggestion's redirections, categories, wordnets, and infoboxes. Such information is parsed at the middleware and delivered to the GUI.

The GUI follows Google's search metaphor and creates a side bar at the left part of the screen, which contains the redirections (captioned as 'synonyms'), categories, wordnets and infoboxes (captioned as 'context words') of the selected suggestion. Since the mashup acts as a query construction/refinement service, elimination of duplicates is performed in the corresponding lists.

Each word/phrase within the lists of the side bar responds to the event of mouse-clicking. Thus, selecting a word/phrase from the lists triggers a corresponding

[24] http://dbpedia.org/sparql.

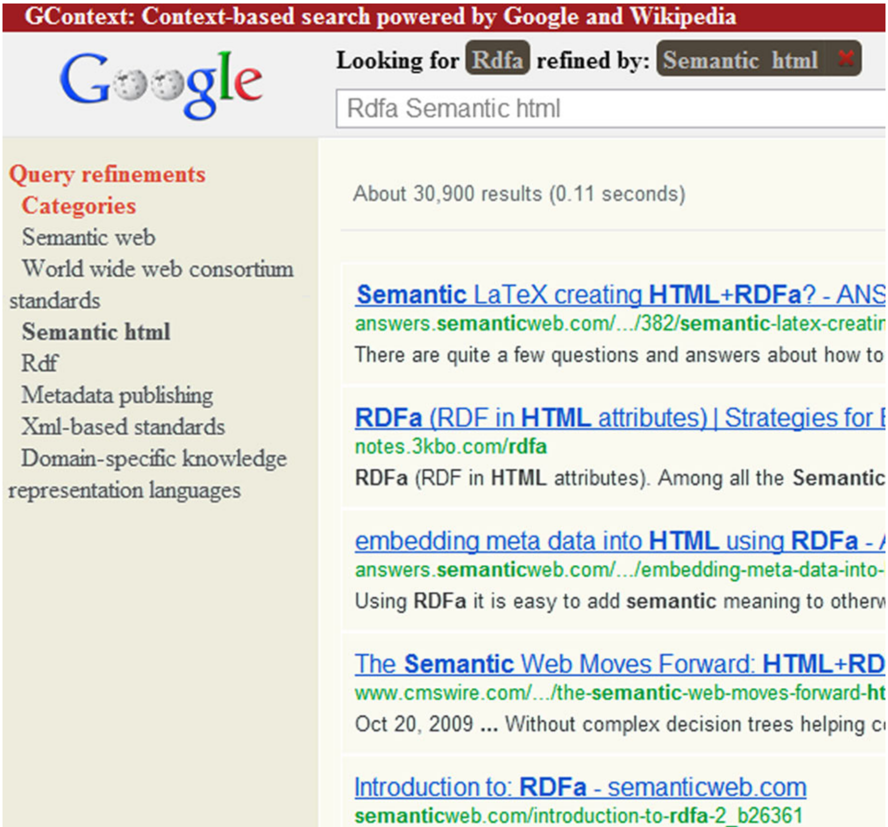



**Fig. 3.10** Query refinements

interaction. The GUI currently provides a number of interactions. Thus, upon selecting a word/phrase, the selection is:

- appended to the search box. Elimination of duplicates is applied to refine the initial query. Then, the refined query reaches the search engine and results are instantly retrieved. An exception to this behavior is imposed by selecting a synonym. In this case, the selection replaces the content of the search box instead of being appended to it (see Fig. 3.9).
- rendered as an interactive bubble on top of the search box. The user has the option to delete the refinement by clicking on the corresponding 'x' sign next to the bubble, or, truncate the bubble by clicking on an individual word of the bubble (in this case, the word appears as deleted (a 'strikethrough' is applied to the word) and is eliminated from the search box). The user has the option to restore the truncated bubble to its original state by clicking on the deleted word. Any interaction with the bubble on top of the search box refines the initial query. Then, the refined query is directed to the underlying search engine, which, in turn, delivers the corresponding search results (see Fig. 3.10).

If the searcher is not satisfied with the provided interactions and their corresponding search results, he may employ the search box once again to perform prefix search on Wikipedia's titles.

#### 3.6.3.3 Implementation Details

As stated earlier in this chapter, the auto-suggest control acts as an entrypoint to the provided service and suggests queries that derive from Wikipedia's titles. Such information is locally replicated and accordingly indexed. It could be argued that DBpedia's own auto-complete search[25] would be more suitable for the needs of the described service. However, DBpedia's auto-complete search index does not meet the requirements of GContext and any attempt to post-process the corresponding results would inevitably suffer from an intolerable time lag.

Moreover, the provided functionality evolves around issuing specific SPARQL SELECT queries to DBpedia's SPARQL endpoint. The information that exists for the provided GUI is dynamically gathered from the corresponding SPARQL responses. So far, the following SPARQL queries have been sketched:

1. 
```
SELECT DISTINCT ?subject WHERE {?subject
<http://dbpedia.org/ontology/wikiPageRedirects>
<uri_of_suggestion>} ORDER BY ?subject
```
2. 
```
SELECT ?object WHERE {{<uri_of_suggestion>
<http://dbpedia.org/ontology/wikiPageRedirects> ?x .
?x <http://purl.org/dc/terms/subject> ?object .}
UNION {<uri_of_suggestion>
<http://purl.org/dc/terms/subject> ?object}}
```
3. 
```
SELECT ?object WHERE {{<uri_of_suggestion>
<http://dbpedia.org/ontology/wikiPageRedirects> ?x .
?x <http://dbpedia.org/property/wordnet_type> ?object .}
UNION {<uri_of_suggestion>
<http://dbpedia.org/property/wordnet_type> ?object}}
```
4. 
```
SELECT ?predicate, ?object WHERE {{<uri_of_suggestion>
<http://dbpedia.org/ontology/wikiPageRedirects> ?x .
?x <http://dbpedia.org/property/> ?object .}
UNION {<uri_of_suggestion> ?predicate ?object}
FILTER ... }
```

The first query retrieves the URIs that correspond to Wikipedia's redirection pages. Such URIs are semantically considered as synonyms for the needs of the service.

The second and third queries retrieves the subjects and the wordnets, respectively, of the suggestion that the user has picked from the auto-suggest control. Such information is semantically considered as categories for the needs of the service.

Finally, the fourth query retrieves the key-value(s) pairs that constitute the infobox of the suggestion that the user has depicked from the auto-suggest control. Then

[25]http://dbpedia.org/lookup.

the information is filtered out to remove words that have low informative value as query terms. For example, the following filtering:

```
FILTER(!regex(str(?p), \"^http://dbpedia.org/property/filename\"))
```

filters out the filenames that possibly occur in an infobox.

#### 3.6.3.4 Evaluation

In order to evaluate the effectiveness of the approach, a comparative analysis against Google's query construction tool (i.e. auto-suggest search box) is performed. The aim of this analysis is twofold: Firstly, to highlight the occasions where web search engines do not seem to perform well, and, secondly, to demonstrate the capability of the approach to fill this gap.

Google's suggestions most likely derive from some kind of statistical analysis of the queries that have been issued to the search engine over time.[26] On the other hand, the described approach matches the user's input against Wikipedia's titles.

When an information seeker expresses his information needs to Google's auto-suggest service and consequently receives the corresponding suggestions, there are two possibilities that may occur:

- The service provides a suggestion corresponding to a query that returns useful search results.
- The service provides a suggestion corresponding to a query that returns either both useful and useless or just useless search results (within the scope of the first search results page).

The first possibility occurs when the auto-suggest service succeeds in aiding the information seeker finding useful information. From empirical studies, it seems that Google's auto-suggest service performs well when the information seeker picks suggestions that correspond to:[27]

- popular queries in general (e.g. "beatles"),
- popular queries within the geographic region of the client computer that invoked the search engine (e.g. "coupons uk" when the search is performed within the United Kingdom),
- queries that have been issued to the search engine before from the same user (i.e. personalization),
- queries that contain unambiguous words (e.g. "afghanistan").[28]

[26]The mechanics of a highly competitive service like Google's auto-suggest service are not formally described. Consequently, any attempt to interpret them is subjective.

[27]How Google Instant auto-complete suggestions work: http://searchengineland.com/how-Google-instant-autocomplete-suggestions-work-62592.

[28]Apart from the above factors there may be other, statistical factors that affect the quality of Google's suggestions.

GContext's auto-suggest service cannot possibly outperform Google's suggestions when information seekers need to compose queries that are affected by such factors.

The second possibility occurs when the suggested query corresponds to a word (or phrase) with various meanings (or 'senses') and the information seeker is interested in the least popular one. For example, an information need about 'jaguar' the animal (not the car) corresponds to the suggestion 'jaguar' from Google, which, in turn, corresponds to a search results list (at least within the scope of the first page) full of resources about the famous car and just one resource about the animal. Suggestions provided by Google lead to even more useless search results as the number of possible meanings of an ambiguous word rises. The situation gets even worse when a popular resource (e.g. movie) is named after a word that literally means something else. Consider, for example, the term 'ajax', which has more than 20 senses according to Wikipedia.[29]

The described service provides a suggestion that correctly expresses the implied information need (i.e. 'Jaguar (animal)') and consequently results in a list full of useful resources about such a need. The service benefits from the fact that disambiguated words/phrases in Wikipedia appear as article titles that provide contextual words within parentheses after the ambiguous word/phrase. Thus, an auto-suggest service that performs prefix search over an index of such literals results in a list of semantically disambiguated suggestions.

Moreover, the service provides query refinements originating from various repositories of the LOD-cloud.[30] The semantic nature of such refinements contradicts the statistical nature of the refinements that are currently provided by Google (i.e. time-based, location-based, type-based, language-based, etc.). Nevertheless, refinements of both services are considered useful and complementary to each other. Thus, they may co-exist in a yet-to-come, integrated environment.

## 3.7 Conclusions—Lessons Learned

In this chapter, the web search engines domain is presented in the context of the particular mashup efforts that have emerged the past few years. The chapter is especially focused on mashups integrating major web search engines and collaborative knowledge to provide value-added query construction and refinement services to their users.

Such knowledge is largely provided by Wikipedia and made readily available in linked data flavor from DBpedia, although a number of individual efforts that are based on Wikipedia have been reported prior to the advent of DBpedia.

Along these lines, a couple of DBpedia-based mashups have been presented (a centralized and a decentralized one), offering query construction/refinement functionality to major web search engines like Google and Yahoo!. The modularity of

[29] Wikipedia's 'ajax' disambiguation article: http://en.wikipedia.org/wiki/Ajax.

[30] http://lod-cloud.net.

the described approaches renders them applicable to any web search engine that provides programmable access its search box. A comparison between the two mashups concludes that the centralized one provides quicker response time to its users. On the other hand, the decentralized mashup eliminates the need to synchronize the information that is being employed by the service and its origin (i.e. DBpedia).

As final notes, it is believed that major web search engines should pay closer attention to the issues that arise at the query formulation phase of a search session. Without proper assistance, searchers may end up issuing queries that are not aligned with their information needs. As described throughout this chapter, query formulation issues are largely attributed to the hidden semantics of the corresponding queries. The inherent ability of semantic web technologies and LOD in particular to address such issues states that in the near future we will see the major web search engines incorporating LOD as part of their default functionality. Along these lines, the 'knowledge graph' announced by Google seems to endorse such thoughts by integrating the collaborative knowledge provided by FreeBase.[31]

## References

1. Aghaee S, Pautasso C (2012) End-user programming for web mashups. In: Haarth A, Koch N (eds) Current trends in web engineering. Lecture notes in computer science, vol 7059, pp 347–351. doi:10.1007/978-3-642-27997-3_38
2. Aktas M, Nacar M, Menczer F (2004) Using hyperlink features to personalize web search. In: Advances in web mining and web usage analysis. Proceedings of the 6th international workshop on knowledge discovery from the web, WebKDD 2004, Seattle
3. Apostolatos I, Papadakis I (2012) GContext: context-based search provided by Wikipedia. In: AI mashup challenge 2012, ESWC 2012. https://sites.google.com/site/aimashup12/home/gcontext/
4. Auer S, Bizer C, Lehmann J, Kobilarov G, Cyganiak R, Ives Z (2007) DBpedia: a nucleus for a web of open data. In: Proceedings of international semantic web conference, ISWC'07
5. Baeza-Yates R, Ciaramita M, Mika P, Zaragoza H (2008) Towards semantic search. In: Natural language and information systems. Lecture notes in computer science, vol 5039, pp 4–11
6. Bates M (1986) Subject access in online catalogs: a design model. J Am Soc Inf Sci 11:357–376
7. Beckers T, Fuhr N (2012) Search system functions for supporting search modes. In: Proceedings of the 2nd European workshop on human–computer interaction and information retrieval, EuroHCIR 2012. http://ceur-ws.org/Vol-909/paper8.pdf
8. Belkin NJ (1980) Anomalous states of knowledge as the basis for information retrieval. C J Inf Sci 5:133–143
9. Bhogal J, Macfarlane A, Smith P (2007) A review of ontology based query expansion. Inf Process Manag 43(4):866–886
10. Billerbeck B, Scholer F, Williams HE, Zobel J (2003) Query expansion using associated queries. In: Proceedings of the ACM CIKM conference
11. Bizer C, Cyganiak R, Gau T (2007) The RDF book mashup: from web APIs to a web of data. In: Proceedings of the 3rd workshop on scripting for the semantic web, SFSW 07. http://ceur-ws.org/Vol-248/paper4.pdf

[31] http://www.freebase.com.

12. Bradley P (2008) Human-powered search engines: an overview and roundup. Ariadne 54. http://www.ariadne.ac.uk/issue54/search-engines/
13. Daoud M, Tamine L, Boughanem M, Chebaro B (2009) A session based personalized search using an ontological user profile. In: ACM symposium on applied computing, Hawaii. ACM, London, pp 1031–1035
14. Dervin B (1982) Useful theory for librarianship: communication, not information. Drexel Libr Q 13:16–32
15. Diaz F, Metzler D (2006) Improving the estimation of relevance models using large external corpora. In: Proceedings of the SIGIR conference, pp 154–161
16. Foskett DJ (1997) Readings in information retrieval. In: Jones KS, Willet P (eds) Thesaurus. Morgan Kaufman, San Francisco, pp 111–134
17. Koolen M, Kazai G, Craswell N (2009) Wikipedia pages as entry points for book search. In: Proceedings of the 2nd ACM international conference on web search and data mining, WSDM'09, pp 44–53
18. Lenat D, Guha R, Pittman K, Pratt D, Shepherd M (1990) CYC: toward programs with common sense. Commun ACM 33(8):30–49
19. Li Y, Luk RWP, Ho EKS, Chung FL (2007) Improving weak ad-hoc queries using Wikipedia as external corpus. In: Proceedings of the SIGIR conference, pp 797–798
20. Mandala R, Tokunaga T, Tanaka H (1999) Combining multiple evidence from different types of thesaurus for query expansion. In: Proceedings of the 22nd annual international conference on research and development in information retrieval. ACM, Berkeley
21. Marchionini G (1995) Information seeking in electronic environments. Cambridge series on human–computer interaction. Cambridge University Press, New York
22. Miller GA (1995) WordNet: a lexical database for English. Commun ACM 38(11):39–41
23. Milne DN, Witten IH, Nichols DM (2007) A knowledge-based search engine powered by Wikipedia. In: Proceedings of the sixteenth ACM conference on information and knowledge management, CIKM'07, Lisbon, pp 445–454
24. Papadakis I, Stefanidakis M, Stamou S, Andreou I (2009) A query construction service for large-scale web search engines. In: Web intelligence and intelligent agent technologies. IEEE/WIC/ACM international joint conferences, WI-IAT'09, pp 96–99
25. Papadakis I, Stefanidakis M, Stamou S, Andreou I (2012) Semantifying queries over large-scale web search engines. J Internet Serv Appl. doi:10.1007/s13174-012-0068-9
26. Pazzani M, Muramatsu J, Billsus D (1996) Syskill & Webert: identifying interesting web sites. In: Proceedings of the 13th national conference on artificial intelligence, pp 54–61
27. Salton G, Buckley C (1990) Improving retrieval performance by relevance feedback. J Am Soc Inf Sci 41(4):288–297
28. Shneiderman B (1997) Direct manipulation for comprehensible, predictable and controllable user interfaces. In: Moore J, Edmonds E, Puerta A (eds) Proceedings of international conference on intelligent IJser interfaces (IUI97). ACM, Orlando
29. Sieg A, Mobasher B, Burke R (2007) Web search personalization with ontological user profiles. In: Proceedings of the ACM conference on information and knowledge management, CIKM'07. ACM, New York, pp 525–534
30. Spencer D (2006) Four modes of seeking information and how to design for them, boxes and arrows. http://www.boxesandarrows.com/view/four_modes_of_seeking_information_and_how_to_design_for_them
31. Suchanek FM, Kasneci G, Weikum G (2007) YAGO: a core of semantic knowledge-unifying WordNet and Wikipedia. In: Proceedings of the WWW conference, pp 697–706
32. Sun JT (2005) CubeSVD: a novel approach to personalized web search. In: Proceedings of WWW'05, pp 382–390
33. Surowiecki J (2004) The wisdom of crowds. Random House, New York. ISBN 978-0385721707
34. Voorhees EM (2003) Overview of TREC 2003. In: REC, pp 1–13

35. White MD, Livonen M (2001) Questions as a factor in web search strategy. Inf Process Manag 37(5):721–740 (p 723)
36. Wu F, Weld DS (2008) Automatically refining the Wikipedia infobox ontology. In: Proceedings of the 17th international world wide web conference, pp 635–644
37. Yokoi T (1995) The EDR electronic dictionary. Commun ACM 38(11):42–44

# Chapter 4
# DBpedia Mashups

**Mihály Héder and Illés Solt**

**Abstract** If you see Wikipedia as a main place where the knowledge of mankind is concentrated, then DBpedia—which is extracted from Wikipedia—is the best place to find the machine representation of that knowledge. DBpedia constitutes a major part of the semantic data on the web. Its sheer size and wide coverage enables you to use it in many kind of mashups: it contains biographical, geographical, bibliographical data; as well as discographies, movie metadata, technical specifications, and links to social media profiles and much more. Just like Wikipedia, DBpedia is a truly cross-language effort, e.g., it provides descriptions and other information in various languages. In this chapter we introduce its structure, contents, and its connections to outside resources. We describe how the structured information in DBpedia is gathered, what you can expect from it and what are its characteristics and limitations. We analyze how other mashups exploit DBpedia and present best practices of its usage. In particular, we describe how Sztakipedia—an intelligent writing aid based on DBpedia—can help Wikipedia contributors to improve the quality and integrity of articles. DBpedia offers a myriad of ways to accessing the information it contains, ranging from SPARQL to bulk download. We compare the pros and cons of these methods. We conclude that DBpedia is an unavoidable resource for applications dealing with commonly known entities like notable persons, places; and for others looking for a rich hub connecting other semantic resources.

## 4.1 Introduction

In this section, we take a closer look at Wikipedia itself, then we examine the process by which DBpedia extracts information from it.

M. Héder (✉) · I. Solt
Institute for Computer Science and Control (SZTAKI), Hungarian Academy of Sciences (MTA), Budapest, Hungary
e-mail: mihaly.heder@sztaki.mta.hu

I. Solt
e-mail: solt@tmit.bme.hu

B. Endres-Niggemeyer (ed.), *Semantic Mashups*, 
DOI 10.1007/978-3-642-36403-7_4, 

### *4.1.1 Wikipedia*

By now, Wikipedia is a big ubiquitous collaborative encyclopedia counting over 10 million articles in over 200 languages. Readers are very active: Wikipedia receives over 10 billion page views per month and over 200 thousand edits per day. However, growth in article count and number of contributions no longer seems to be exponential for the largest English language edition.[1]

For our purposes, contrasting Wikipedia to traditional printed works is not essential, but it allows us to draw attention to some of its key characteristics. Wikipedia is not governed by a formal editorial board, but instead by the community and its self-imposed guidelines, decision making and escalation processes. Unavoidably, the coverage of articles in a given language edition is biased towards public interest of the Wikipedians speaking the language. The English language Wikipedia has been found to be on a par in accuracy with Encyclopædia Britannica [12], and with peer reviewed medical journals [25]. Furthermore, Wikipedia has the unmatched ability to cover current events and incorporate changes in near real time.

Also, Wikipedia is free to download and hack for everyone. As all digital documents, it has structural elements, like lists and tables. Like encyclopedias, it also has a category system. Furthermore, it contains many infoboxes—structured schemas that communicate facts about the subject of the article, for instance of a city. Users can find the infoboxes at the top right part of certain articles. It is not easy to access the infobox data programmatically. Many parsing related issues originate from the rather complicated and less standardized nature of wikitext. In spite of these problems the DBpedia project was started in order to extract and structure Wikipedia information.

### *4.1.2 DBpedia*

As its name suggests, DBpedia,[2] aims to provide a structured view of user-contributed Wikipedia content [3, 6]. The structuring of the vast amount of data in Wikipedia allows new and innovative uses including querying, navigation, association and aggregation. While the consistency of DBpedia may not keep up with some domain-specific knowledge bases painstakingly crafted by domain experts, however, its broad coverage and almost real-time updates are key advantages to many applications.

#### **4.1.2.1 The DBpedia Ontology**

DBpedia normalizes information extracted from Wikipedia infoboxes at various levels. A nice example is shown in Fig. 4.2. First, the infobox type is mapped to an

[1] Since 2007, see http://stats.wikimedia.org/EN/#see_also.

[2] http://dbpedia.org.

Ontology type, e.g., the Wikipedia article having an “Infobox Austrian district” is classified as dbpedia-owl:AdministrativeRegion. Ontology types have a fixed set of properties that are populated based on infobox property mappings, e.g., the “twin1” property of the Austrian district infobox is mapped to the ontology property dbp-prop:twinCity which has the range dbpedia-owl:Settlement.

The mapping of infobox templates and properties to ontology types and properties ensures that users of DBpedia data need not be concerned about the peculiarities of the organically evolving Wikipedia infoboxes and can focus on remixing the data instead of creating it.

As you might expect, crafting mapping rules is straightforward in some cases, but overly complex in others. One source of complication is the fact that Wikipedia is not like a database: Infobox properties may be (and often are) filled out in unexpected ways due to the non-triviality of the information entered, but also due to the insufficient guidance on how infoboxes should (not) be filled out. To cope with this complex and also dynamically changing landscape across the dozens of Wikipedia editions DBpedia has opted for crowd-sourcing by the enaction of the DBpedia Mappings Wiki. This meta-wiki allows DBpedia contributors to define and adjust mapping rules for the Wikipedia edition they are most familiar with.

Property values are normalized based on their specified or expected range, e.g., a number entered may be interpreted as a year or a distance in miles/kilometers, depending on the property type and the presence of explicit range specifiers (such as Wikipedia date and conversion templates). Example property ranges contained in DBpedia:

- numeric: integer, float, double
- metric: length, area, volume
- geographical: latitude, longitude, elevation, region
- temporal: date, time, interval

Normalizing plain text property values to the appropriate ranges requires localized parsers to be added to DBpedia's codebase that cope well with incomplete, non-standard, mistyped or even inappropriate user input. For example, expect even correctly spelled dates to be represented as a native string instead of a normalized xsd:date for non-English Wikipedia editions.

Now that we have an overview of the structure of the database we should not forget to ask what is the connection of the data to the real world—what is the semantics of the data? DBpedia is a result of an empirical experiment, and as often happens with these kind of enterprises, the theoretical framework is running late in explaining its nature and widespread success. As long as we do not want to dig deep, we can say that Wikipedia describes all kinds of things, like persons, groups, locations, events, activities, concepts, etc. These things might be fictional or real, or thought to be real in the past or maybe expected to become real in the future. What we can say about them is that they are mostly backed by a consensus. At this point of course all kinds of exceptions come in mind, like articles about politicians, etc.—but that is not the majority, and also the debates are rather about evaluations and normative statements and much rarely about things like the birth date of a certain politician. Second, we can safely assume that the things on Wikipedia are notable enough—there are guidelines to ensure that.[3] Therefore DBpedia data represents machine-readable facts from all the kinds of things mentioned above.

This resonates nicely with the original Semantic Web project, but that project also included a heavy mathematical toolbox to define ontologies to allow machine inference: the Web Ontology Language—abbreviated as OWL—was created.

OWL by design has Description Logic semantics. The older OIL language was designed to implement a description logic called SHIQ and a software called FaCT[4] is used to carry out so-called T-box and A-box reasoning on it. OIL was submitted together with another language called DAML to W3C and there it became OWL [20].[5]

Terminology-box (T-box) and Assertion-box (A-box) are terms from description logic [4]. The terminology of a system is defined in a T-Box, and its statements are usually about concepts (sets of objects) and roles (binary relations). A-boxes are about individuals and contain two kinds of different statements: C(a) and R(a, b). C means "concept assertion", where R is a "role assertion". Examples look like Man(tom) and Parent(tom, jenny).

After this short introduction we can see why some DBpedia developers themselves often characterize their data set aptly as a large A-box. Although they map infobox properties to OWL properties, the development of the original infoboxes and therefore their terminology (the T-box) is outside the scope of DBpedia. One can think of editors on Wikipedia as the ones who develop both T-boxes (categories, infobox templates and textual descriptions of them) and also A-boxes (actual infobox data about individuals). DBpedia can only retrieve the data because it is machine readable, but not the meaning of text (unlike human readers) on which most of the knowledge relies.

Naturally, this is only one interpretation of DBpedia semantics. There are many others and in general there is a lack of consensus about this question (for a good criticism on Linked Open Data and its usefulness in general see [17]). We should not think that this question is only the businesses of theorists, however. To use DBpedia data in any kind of intelligent application an interpretation must be found, one way or the other. What is usually happening is that the developer of a mashup examines what is available at DBpedia; she either already knows or investigates what kind of articles was the data of her interest extracted from; finally she builds an application according to her own somewhat custom interpretation. This means that all the various applications of DBpedia we will discuss later in this chapter should have implied their own idea of the semantics of the data. There is no problem with that—this is just how it works.

[3] http://en.wikipedia.org/wiki/Wikipedia:Notability.

[4] http://owl.man.ac.uk/factplusplus/.

[5] http://www.w3.org/TR/owl-features/.

#### 4.1.2.2 DBpedia in Numbers

The DBpedia dataset is a collection of information on about 3.77 million things,[6] half of which are classified into the unique *DBpedia Ontology*. The distribution of things roughly matches public interest: 500+ k places, 400+ k persons, 180+ k species, 160+ k organizations, 100+ k music albums, 60+ k movies. Altogether 1 billion pieces of information are extracted from the various Wikipedia language editions, though 40 % come from the largest English edition. Names and abstracts are thus available in multiple languages, together with links to images and websites.

#### 4.1.2.3 DBpedia's Connections to Other Resources

One of the ways DBpedia goes beyond being just a large, isolated database is its rich connections to other projects. Such links explicitly state the equality (owl:sameAs) of a DBpedia entity and a third party concept, allowing creative mashups. DBpedia currently explicitly interlinks with 18 other knowledge bases available as RDF, including:

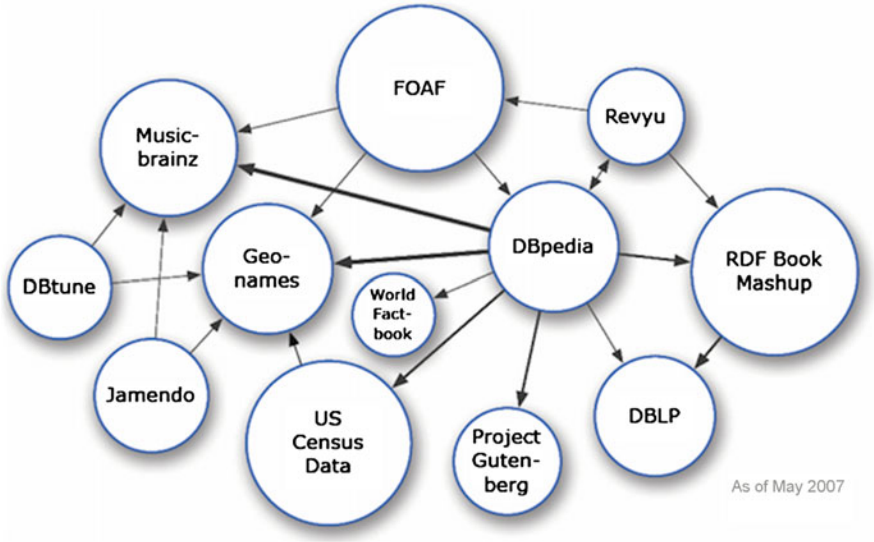
- specialized KBs, e.g. GeoNames, MusicBrainz, Project Gutenberg, Drugbank
- statistical KBs, e.g. US Census, EuroStat, World Factbook
- ontologies, e.g. WordNet, OpenCyc, New York Times

As DBpedia exposes external links found on Wikipedia articles, it may be used to associate things to other repositories, e.g.:

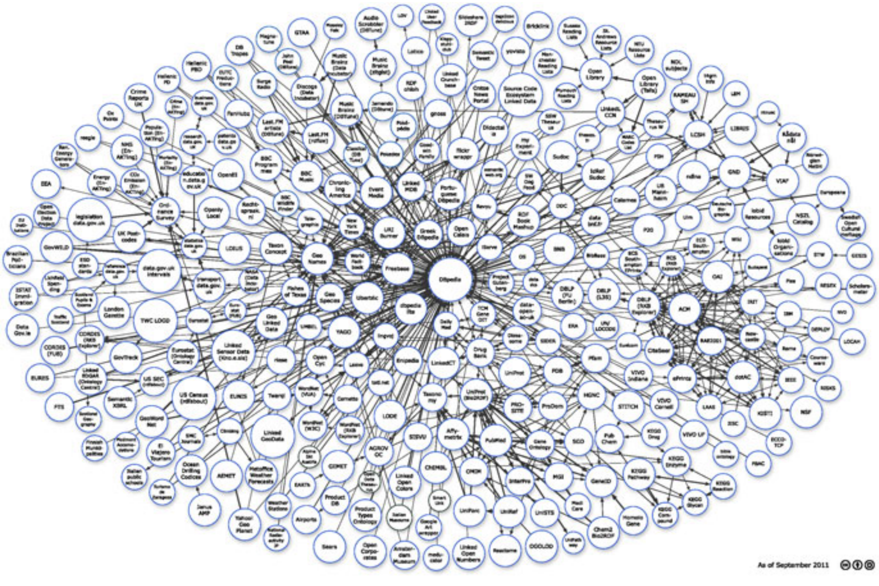
- images via Wikipedia uploads, Wikimedia Commons and Flickr wrappr
- videos via YouTube links
- movies via IMDb links
- social media profiles via Facebook and Twitter links

DBpedia has become an integral part and flagship of the Linking Open Data initiative under the umbrella of the Semantic Web. Key to the advancement and adoption of these agendas is to have meaningful and diverse data available to be linked to. In this respect, DBpedia is portrayed as a *nucleus* [3] due to its aforementioned coverage and outlinks to over a dozen other knowledge bases. Furthermore, many other datasets link to DBpedia, effectively making it a hub in the cloud of Open Data (Fig. 4.1).

[6] As of August 2012. For more details, see: http://wiki.dbpedia.org/Dataset.




(a) As of 2007



(b) As of 2011

**Fig. 4.1** The Linking Open Data cloud diagram [9] depicting the increase in the number of knowledge bases also available in RDF, and the central role DBpedia plays in their cross-linking


| Property | Value |
|---|---|
| dbpprop:founded | 1842 (xsd:integer) |
| dbpprop:founder | dbpedia:Julius Springer |
| dbpprop:country | dbpedia:Germany |
| dbpprop:headquarters | dbpedia:Berlin<br>dbpedia:Heidelberg |
| dbpedia-owl:abstract | Springer Science+Business Media S.A., mit Sitz ...<br>Springer Science+Business Media or Springer is ... |
| dbpedia-owl:wikiPageExternalLink | `http://www.springer.com` |
| dcterms:subject | category:Pan-European media companies<br>category:Academic publishing<br>category:Commercial digital libraries<br>category:Publishing companies of Germany |
| foaf:depiction | http://upload.wikimedia.org/.../Springer.jpg |
| is dbpprop:publisher of | dbpedia:Society (journal)<br>dbpedia:European Physical Journal |

**Fig. 4.2** Wikipedia infobox and DBpedia data for Springer, http://dbpedia.org/page/Springer_Science%2BBusiness_Media. The data on the *right-hand side* is derived from the infobox, the category links on the page, the abstract of the article and other features

### *4.1.3 Freebase*

Freebase,[7] a collaborative knowledge base backed by a for-profit organization,[8] has taken a different approach to extract and expose structured information from Wikipedia articles [7]. Instead of being a "read-only" repository, it allows and socially encourages its users to edit and extend its contents on a database-editor-like user interface (compare to Wikipedia's single textbox edit interface with cumbersome wikitext syntax). Thanks to this approach, Freebase can and indeed does grow independently from Wikipedia, housing data on things that would not meet Wikipedia guidelines[9] such as amateur artists, local businesses and offices. In Freebase, also the type system (the analog of the DBpedia Ontology) is dynamic and can be edited by users, however, only in limited ways to maintain consistency and avoid vandalism.

Freebase regularly crawls Wikipedia for new information, updating and creating entries as necessary, while also paying attention to preserve any Freebase user edits that may have taken place between two such cycles. Freebase's extraction framework[10] (WEX) transforms Wikipedia articles available as wikitext into a well-formed and structured XML dump, opening it up for other projects without the need to deal with peculiarities of wikitext. Both DBpedia and Freebase data are free as in 'free speech', they are made available under a Creative Commons Attribution license that allows both derivative works and commercial use as long as the source

[7] http://www.freebase.com/.

[8] Metaweb Technologies, Inc. It has been acquired by Google in 2010.

[9] Especially the notability test: https://en.wikipedia.org/wiki/Wikipedia:Notability.

[10] http://wiki.freebase.com/wiki/WEX.

**Table 4.1** Comparison of DBpedia and Freebase

| | **DBpedia** | **Freebase** |
|---|---|---|
| Entities | 3.77 million | 22 million |
| Data access | read only | read–write |
| Ontology modification | maintainer only | limited |
| Download | RDF (N3) | TSV |
| Query language | SPARQL | Metaweb Query Language |
| HTTP API | Structured query, keyword and prefix search results in JSON | |
| Content license | CC-BY-SA (Attribution-ShareAlike) | CC-BY (Attribution) |
| Source code license | GPL | proprietary |
| Hosted by | University of Leipzig, Freie Universität Berlin, OpenLink Software | Metaweb Inc. (acquired by Google Inc.) |

is acknowledged. DBpedia and Freebase are easily mashed up as both interlink with each other. A comparison of some aspects of the two databases is presented in Table 4.1.

### *4.1.4 Wikidata*

Finally, it is appropriate to briefly mention here a third project called *Wikidata*,[11] which aims to create a semantic database that is curated with similar principles as Wikipedia itself. At the time of writing this chapter (mid-2012), the system is under development, so it is too early to evaluate. But the situation can change quickly, so you should check on the project at the time of reading this.

## 4.2 Mashups of the Domain

DBpedia interfaces well with other applications due to its broad coverage across topics, languages and geographical regions. DBpedia makes it easy and free to internationalize and localize some applications, features that could be economically unfeasible to license or implement for oneself. Due to the steady growth of Wikipedias across the globe and continued development of DBpedia mapping and extraction frameworks, DBpedia further improves in coverage, consistency, and interoperability. In this section, we present a few mashups that use DBpedia, and give an overview of the various ways to access DBpedia data.

[11] http://www.wikidata.org.

## *4.2.1 Mashups That Are Already Using DBpedia*

One straightforward way of using DBpedia data is to create custom visualizations. The most spectacular things can be made with the geographical data. One can put a specific subset of the data on a Google map or an alternative map technology. Another nice looking and useful thing is to put events or life spans on a time scale. Of course, data can be cumulated by country or filtered by certain conditions before visualization—the options are endless. The following three mashups are nice examples of this kind of application.

### 4.2.1.1 Maps and Visualization

#### 4.2.1.1.1 DBpedia Mobile

DBpedia Mobile[12] [5] is a location enabled augmented map viewer mashup targeting mobile devices. As you might expect, the user can navigate on a Map enriched with hundreds of thousands of geo-referenced DBpedia entities, including of course points of interest and geographical features. Of course, all these geographical entities are not shown all at once—the whole point of this application is that the user can specify what exactly she wants to see using the semantic features of the data. This way the maps will not become over-populated. The fact that the DBpedia dataset is interlinked with GeoNames, US Census, the CIA factbook, and Eurostat datasets provides a rich user experience. What is more, for certain entities photos can be viewed with the help of Flickr wrappr[13] or even reviews can be read from Revyu (see later in this section).

Other distinguishing features of DBpedia Mobile are the ability to switch the language in which labels are displayed (independently from the region viewed), and SPARQL integration for selecting entities to be shown. DBpedia data offers some elegant ways to construct the interactive summaries for entities, including text summaries, native name, official website, or hierarchical navigation.

#### 4.2.1.1.2 *Vispedia*

Vispedia[14] is a visualization interface that is tuned for the visualization of the results using DBpedia. The main idea behind Vispedia is that the best way to consume the semi-structured data of Wikipedia is by interactive data exploration facilitated by a nice interface. In other words, the goal of the system is to bring down the cost of finding and accessing relevant data [8].

---

[12] http://wiki.dbpedia.org/DBpediaMobile.

[13] http://www4.wiwiss.fu-berlin.de/flickrwrappr/.

[14] http://graphics.stanford.edu/projects/vispedia/.

"Data" here means every kind of tables on Wikipedia that can be put on a map, timeline or scatterplot. As the tables often do not contain all the relevant features, semantics from DBpedia is involved. The starting point of a visualization is a Wikipedia article that contains a table. The rows of the table are converted into a graph, in which one node corresponds to each row. At this point, data integration with DBpedia data is carried out interactively by the user, who enters keywords to indicate what kind of information she needs. The keywords are compared to graph edge labels, a similarity measure is calculated and the similar graph edges are included in the result set. Using the similarity measure as path cost, a time-limited A* (A-star) graph search is performed to find relevant entities. By search for keywords being refined, an interactive sense-making loop can be established.

#### 4.2.1.2 Search

DBpedia data, along with the many options for visualization also facilitates search. Using semantics in search enables to find entities not only by literal occurrences in text but also by inference—that has been a long-standing goal of the Semantic Web project. The navigation and presentation can also be enhanced by structured data, like in the following two mashups.

##### 4.2.1.2.1 *Contentus*

Contentus[15] is a Semantic Search Engine with nice user interface that carries out multi-modal search over web documents and linked data like DBpedia. It is capable of marking up web documents with semantic data and present the result to the user[16] [24].

The main motivation behind the project is that ever-growing open data sets and available content could create novel ways for libraries or other cultural institutes to present their collections on the internet. In this scenario, the existing collections are most of the time already annotated by librarians or information scientists manually. These metadata have to be integrated with other sources. At the same time, the manual annotation of new digital content is becoming an ever-growing problem because of the increasing pace of collection growth and the lack of human resources. Besides of integrating internet resources (e.g. Wikipedia or GeoNames items) with the metadata about a whole document, in the case of digital documents the indexing and linking of entities *within* the content is also possible and could be facilitated. This is what is attempted by Contentus developers. Besides giving tools to the curators, a nice end user interface is provided. On this interface the users can search in the contents of various multimedia libraries. For instance, a newspaper article might be

[15] http://www.iais.fraunhofer.de/contentus.html.

[16] For a screencast see: http://www.yovisto.com/labs/vissw2011/.

presented in its original scanned format, complemented by the text extracted with OCR. In the document presentation, the semantic entities are highlighted. These entities and their relations are provided by the German National Library's[17] person database,[18] but these are mapped to DBpedia and thus linked to the Open Data cloud.

#### 4.2.1.2.2 SWSE

SWSE[19] is another Semantic Search Engine that crawls the web for semantic data. Among its results DBpedia entities are among the top ones [16]. SWSE is designed for the users of the general web. It provides search by keywords just like Google, Yahoo, Bing, and others, but, instead of giving back links to the documents that contain the keywords provided, it returns a ranked list of semantic descriptions about real-world entities. Along the relations of the found entities, users can navigate and discover other entities.

Just like any search engine, SWSE has its own indices that are built by crawlers and indexers. DBpedia and Freebase are only two of the many RDF sources the system relies on. Among the usual crawler and indexer components there are some that are specifically designed for RDF: there is a consolidation component that tries to merge the duplicate entities; also there is a reasoner that generates new RDF based on the existing data.

### 4.2.1.3 Recommendations and Reviews

The graph that is constituted by the data of the DBpedia makes it possible to reason about how "close" some things are to others by some measure. This allows for recommending things for the users based on what we already know about their preferences as in the following mashup.

#### 4.2.1.3.1 dbrec

dbrec[20] [22] is a music artist and band recommendation mashup based on DBpedia data. By correlating genres, joint performances and album releases of artists, recommendations are made and also explained to the user. The underlying distance algorithm is implemented via SPARQL queries, and is precomputed over the dataset. User experience is made more attractive by including DBpedia supplied images, descriptions and YouTube videos.

[17] Deutsche Nationalbibliothek—DNB.

[18] Personennamendatei—PND.

[19] http://swse.deri.org/.

[20] http://dbrec.net/.

dbrec can provide recommendations for almost 40 thousand artists and bands. These recommendations are based on the so-called Linked Data Semantic Distance (LDSD) algorithm. This algorithm is tailored for the characteristics of LOD data: it only relies on links, not on label distance, it does not use a general ontology, only instance data, and it exploits the fact that most of the LOD URIs are dereferenceable—meaning that the URI can be fetched by an HTTP GET. The result of the computation with LDSD is a measure normalized to the [0, 1] interval between two LOD resources.

This algorithm was applied to the more than 39 thousand artists and bands that could be found (after some data cleaning) on DBpedia at the time of creating the system. This means that basically all the distances between particular artists/bands and all the rest were computed (again, with some optimizations)—that took several days. Using the distance database users can simply find similar artists/bands to their favorite ones. Moreover, the database also provides information on what properties shaped the distance measure. This is called "explanation" and turned out to be a popular feature for the users.

#### 4.2.1.3.2 Revyu

Revyu[21] is a portal where the users can review anything they want. To render a better presentation of the reviews to the users it uses data from DBpedia [13]. At the same time, the site not only consumes but generates RDF as well. The reviews written in the system are processed and e.g. in case of movies, queries against the DBpedia endpoint are executed. In simpler cases this results in the DBpedia resource for the given film. Similar heuristics are applied in case of books, using the RDF book mashup. This is called "retroactive linking" by the developers. They also use "proactive linking", meaning that they generate "skeleton" (empty) reviews for things users might want to review based on LOD data. The limitation of this approach is that there are too many potential entities to cope with. Besides of linking to DBpedia and the LOD in general, Revyu is nicely linkable: the reviews have their own URIs that are dereferenceable.

### 4.2.1.4 Plain Text Enrichment

Recommendations cannot only be based on a distance measure within a graph, they can be based on co-occurrence and even on natural language processing (NLP) techniques.

*Zemanta* is a blogging assistant that helps bloggers to enhance their content by links to Wiki articles, other blogs, amazon, IMDB entries and such. With the enhancements the users' content tends to reach a better place in the search results and to get more links back when linking to other blogs.

[21] http://revyu.com/.

*BBC Content Link Tool* uses DBpedia to help editors in properly tagging any BBC URLs with appropriate semantic metadata [18].

*Apache Stanbol*[22] is an OSGi-based Semantic Enhancement Engine. This means that one can send content to the system through an API, and Stanbol responds with enhancements in RDF. The system is integrated with DBpedia Spotlight (see later in this section) and many other annotation sources, e.g. Zemanta or OpenCalais.[23]

#### 4.2.1.5 Identifying

DBpedia is good for identifying things that people put on a website or portal.

For example, *Flickr wrappr* itself is a mashup, it maps photos to things by correlating Flickr tags and geotags with DBpedia labels and geo-references.

Not only places, but also persons can be identified. The *White House Visitor Log*[24] is a demo mashup made at Rensselaer Polytechnic Institute which shows how different sources of data can be mashed up in a single application. On this website users can search for the visitors to the White House—the data are taken from data.gov—and the search results are enriched with DBpedia data, as many of the visitors are prominent politicians with their own Wiki pages [10]. Similarly, *Academia Europea*[25] has many members whose profile page was created with the help of DBpedia and DBLP data [19].

Finally, you can rely on the entities in DBpedia when populating a new portal with labels and categories. After all, why start with an empty category set or label set when you can have a sensible one right away? *Faviki*[26] is a social bookmarking platform that has chosen to use DBpedia entities (that is, Wikipedia articles) as tags. When it comes to tagging, the traditional choices have been using a fixed tag set (taxonomy; e.g., DMOZ Open Directory Project) or allowing any tags to be entered by the user (folksonomy; e.g., Flickr). Faviki's approach benefits from both sides: Users select tags from a large but also mostly unambiguous tag space, while the tag space itself is kept current by Wikipedia contributors. Additional benefits that come without extra user or maintainer effort are the support for multiple languages and the structured information associated to tags, notably hierarchical generalization–specialization relations.

This was only a sample of the DBpedia mashups out there that we have found mostly by browsing the many hundreds of citations the initial DBpedia white papers.[27] This means that lots of applications are probably excluded. Also a large

[22] http://stanbol.apache.org/.

[23] http://www.opencalais.com/.

[24] http://logd.tw.rpi.edu/demo/white-house-visit/search.

[25] http://www.ae-info.org/.

[26] http://faviki.com/.

[27] If you want to do your own research, use Google Scholar and search for "DBpedia: a nucleus for a web of open data" and "DBpedia—a crystallization point for the Web of Data". The two articles together received a remarkable 1,300 citations to date.

number of mashups still existed on paper and in the form of screen shots but the URLs were unaccessible by the time we checked. This indicates how fast the scenario changes—we can only hope that you could really find most of the mashups above.

## *4.2.2 Accessing DBpedia*

DBpedia dataset is accessible in many ways making it easy for both humans and computers to tap into its wealth of information. This section aims to give an overview of the most commonly used methods to access DBpedia dataset.

### 4.2.2.1 Download

If you like it raw, DBpedia offers regularly updated bulk downloads of its dataset and ontology at http://wiki.dbpedia.org/Downloads.

The DBpedia ontology itself is made in OWL and serialized in RDF/XML for download, while the dataset files are hosted in the less verbose N-Triple and N-Quad format. The latter RDF serialization format is more storage-friendly and also less resource-hungry to process.

### 4.2.2.2 SPARQL Endpoint

DBpedia also provides an interactive query interface and a RESTful web service at http://dbpedia.org/sparql providing a variety of output formats including de facto standard JSON. The endpoint interprets the popular RDF query language SPARQL, for a hands-on introduction to SPARQL[28] refer to [1]. Here is an example DBpedia query using only DBpedia and FOAF ontology properties:[29]

```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?name ?birth ?person WHERE {
     ?person dbo:birthPlace :Berlin .
     ?person dbo:birthDate ?birth .
     ?person foaf:name ?name .
     FILTER (?birth < "1900-01-01"^^xsd:date) .
}
ORDER BY ?name
```

This will result in an output containing:

[28] http://www.w3.org/TR/rdf-sparql-query/.

[29] http://wiki.dbpedia.org/OnlineAccess.

```
{"name":  { "type":"literal",  "xml:lang":"en",
             "value":"\"Helene\" Ellen Franz"        },
 "birth": { "type":"typed-literal",
             "datatype":"http://www.w3.org/2001/XMLSchema#date",
             "value":"1839-05-30" },
 "person":{ "type":"uri",
             "value":"http://dbpedia.org/resource/Ellen_Franz" }}
```


The above query will return the name and date of birth of individuals born in Berlin before 1900. Note that `Berlin` here unambiguously refers to the English Wikipedia article with title 'Berlin', thus the German capital; and that the property `name` obtained from the FOAF ontology is in English.

Be aware that processing many and/or complex queries puts a significant burden on the DBpedia's backend servers, which are operated in a non-profit fashion. In such situations, or when dealing with sensitive information, you might consider setting up DBpedia appliances for private use.

#### 4.2.2.3 Virtual Appliance

On-line DBpedia interfaces are driven by the Virtuoso[30] server platform. Virtuoso is also available for free[31] and comes with instructions on how to import the latest DBpedia datasets, thus offering users a way to set up DBpedia endpoints for private use. To make life easier, the developers of Virtuoso offer downloadable appliances preinstalled, preconfigured and populated with DBpedia data for Amazon's EC2 virtualization platform.[32]

Amazon's Public Data Set also includes DBpedia, which facilitates integration to other Amazon Web Services applications.

#### 4.2.2.4 DBpedia Spotlight

We have seen that the coverage of DBpedia offers some unique ways to semantically enrich structured data. However, finding the pieces of data to be enriched in content like free text can be a major challenge.[33] DBpedia Spotlight[34] [21] aims to overcome this semantic gap by analyzing plain text and automatically suggesting linkages between DBpedia entities and text spans, very much like Wikipedia internal links. Thus DBpedia Spotlight extends the scope of applications that can benefit from DBpedia, adding e.g., blogs, libraries, feed aggregators, or other applications

[30] http://virtuoso.openlinksw.com/.

[31] More precisely it has a free version, besides the enterprise plan.

[32] http://www.openlinksw.com/dataspace/dav/wiki/Main/VirtAWSPublicDataSets.

[33] See literature on *information extraction*.

[34] http://dbpedia.org/spotlight.

dealing with user generated text content. Sztakipedia uses DBpedia Spotlight as a source for link recommendation and there is another small application that helps in discovering Google Summer of Code projects.[35] This application leverages relationships between concepts in DBpedia in order to suggest "related topics" for students searching for a project. For example, if a student searches for "Cloud Computing", the mashup is able to suggest other concepts such as "Platform as a service" and "Scalability".

## 4.3 Sztakipedia Project

Sztakipedia builds upon structured data from DBpedia and from many other different sources and intends to be an intelligent assistant for Wikipedia editors. We are completely sure that every reader of this sentence has read at least one Wiki article in her/his life. However, the huge majority of the readers of Wikipedia have never written a single article. They probably do not suspect what exactly writing an article involves. Let us suppose someone wants to write an article on a more-or-less known historical person. Probably the author is deeply interested or is an expert in the topic, so she has some kind of draft in her head or even text portions ready to copy–paste and revise. She gets a standard HTML text area in which she can compile the plain text. The next step is to format the text. Right now it is mainly done by wikitext markup, although a new visual editor is being developed by the Wiki developers that will most probably become really popular.[36] But our focus right now is on what happens after the formatting: inserting links, infoboxes, categories, and the necessary citations to sources. In our case the author should at least use the `person` infobox, or a more specific one, like `philosopher`. Proper category labels should also be added, as well as source citations, which are required by the Wikipedia editing policies. And, naturally one should link the more important concepts in the text to their corresponding articles. If this is not done, either because the author is new to the system and is not familiar with infoboxes, the category system, etc., or because she does not have enough time to learn the details of the syntax, the newly written article will be reverted, or labeled as "stub" by someone with more administrative power, and for a good reason: source citations, pagelinks, infoboxes, and categories are crucial for the quality of the content. So they cannot be omitted, but the creation of them could be assisted by a recommender system—this is the idea behind Sztakipedia.

The system design is based on a requirement survey that was conducted among more than 1450 Hungarian Wiki editors [14]. Sztakipedia uses DBpedia data, among other sources, for making suggestions of different kinds, offering pertinent content for editors to add to the documents, e.g. Wikipedia infoboxes, categories and page

[35] http://spotlight.dbpedia.org/gsoc/.

[36] In the early stage of Sztakipedia project our team also developed a TinyMCE-based editor, but that is discontinued now.

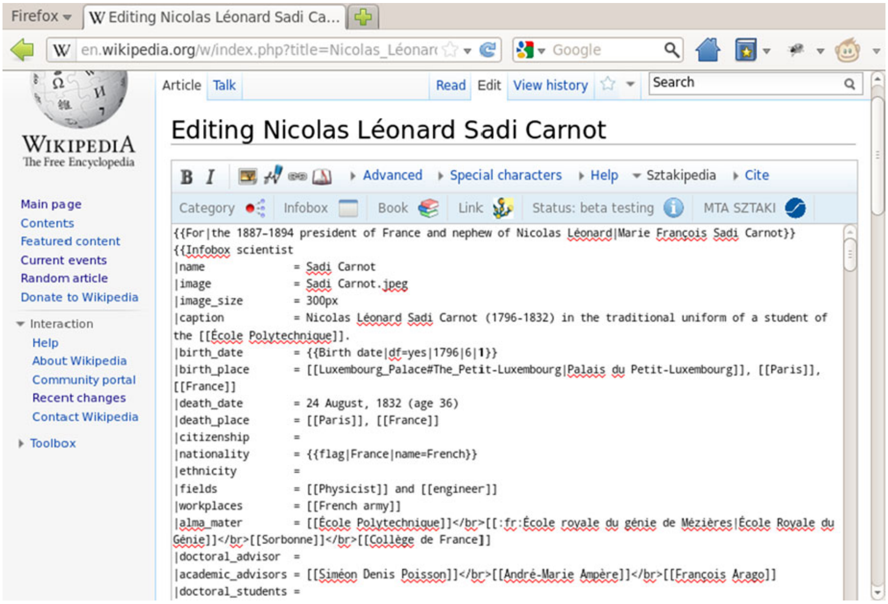



**Fig. 4.3** A screenshot from Sztakipedia toolbar

links. Among the information sources used, we highlight Web search, library catalogs, as well as tf-idf[37] database and co-occurrence data extracted from Wikipedia. Through the use of Sztakipedia, Wikipedia users can unknowingly reuse DBpedia data when editing articles, through a toolbar from the standard Wiki editor interface (Fig. 4.3). The assisted editing of articles can increase the level of interconnection of existing knowledge and potentially enhance the quality of articles on Wikipedia.

Our broader vision is that a virtuous cycle of semantic enhancement can be created by assisting knowledge creation. The more authors are using a recommender system to create machine-readable annotations in the content, the more training data is created for enhancing the recommenders, this creates a positive feedback loop. An important element of this vision is that the user should make most of the decisions about the suggestions. As a consequence the IS must be present on-line in the editor interface of the user, e.g. as a plugin.

Leaving the Wiki context, one can find more practical reasons for enriching an article with links and other enhancements. The point of writing a Wiki article, a blog post or a forum entry is to convey a message to others. On the web, it all works asynchronously: we find these messages by searching for keywords in a search engine or by clicking on interesting links on one of the very few a web pages we regularly visit, which are usually news portals or social platforms. This is why the writer has to put the new document in context by labeling or categorizing, and enriching it with links and metadata—to make it accessible and *related*. The more related the

[37] tf-idf is a widely used statistical relevance measure. For details, see [23].

content is, the more visitors will find it and the more revenue will come from advertisements. If you plan to become a professional blogger who is creating content frequently and effectively, you can already use a tool like Zemanta (which also uses DBpedia data) to streamline the document creation process.

## 4.3.1 Features of Sztakipedia

Here we present Sztakipedia-toolbar[38] [15]—which consists of a MediaWiki user script and a modular server in Java—that can be easily enabled by any Wikipedia user—currently fully functional only for the English and Hungarian Wikipedia. The toolbar provides access to four main functions.

### 4.3.1.1 Link Recommendation

Good links are essential in a document, and so is link recommendation in an intelligent assistant application. In Sztakipedia, this function is partly based on DBpedia's "Links to Wikipedia Article" dataset. This file contains every link that points from one Wiki page to another one. Also, one can count the frequency of linking to the articles. In the English case, given that most of the words or phrases the system encounters are likely to have a Wiki page, one can supplement or even replace tf-idf calculation. This is very useful if there is no initial corpus on which a statistical relevance system might be trained. In Sztakipedia we use a phrase weight measure which is based on the product of tf-idf and DBpedia frequency. This measure forms the basis of pagelink recommendations and their ordering. However, this mechanism is complemented with DBpedia Spotlight, DBpedia's own link recommendation feature (see earlier in this chapter). DBpedia Spotlight relies on a number of name–URI associations extracted from titles, redirects and disambiguates, as well as TF*ICF (Inverse Candidate Frequency) scoring [21] of the target text to choose between possible disambiguation options. When the author requests pagelink recommendations, the plain text document derived from wikitext is processed by an UIMA[39] engine, which finds all the words and phrases, which are also page titles, and calculates a weight for them for ranking. Parallel to this, DBpedia Spotlight also processes the text on the DBpedia server. The results are merged and presented as link recommendations to the user. See the top-right corner of Fig. 4.4 for a screenshot.

### 4.3.1.2 Infobox Recommendation

The implementation of this function is based on document similarity, calculated by the Lucene[40] framework. The articles in a Wikipedia dump are transformed

[38] http://pedia.sztaki.hu/.

[39] UIMA stands for Unstructured Information Management Architecture. It is a modular framework for annotating content. For more details, see http://uima.apache.org/.

[40] http://lucene.apache.org.

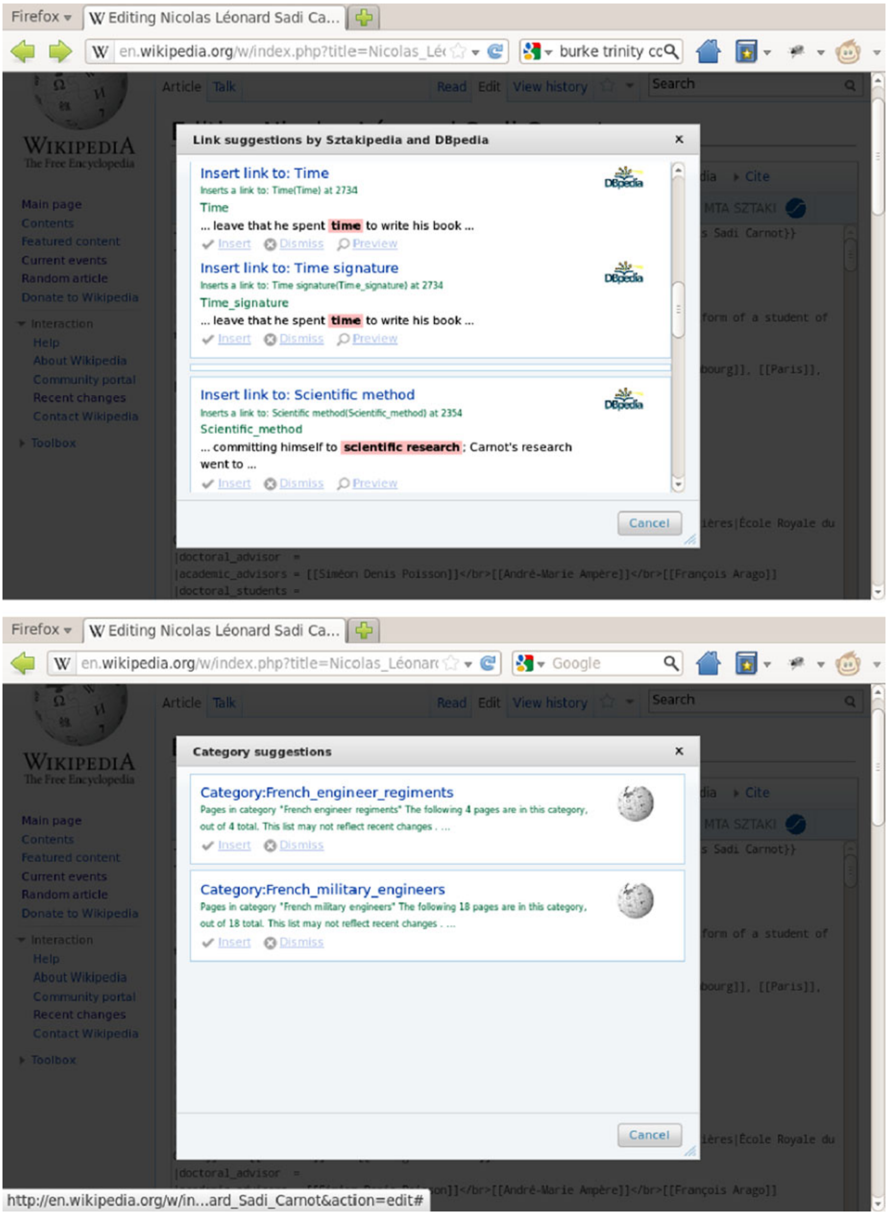



**Fig. 4.4** Link and category recommendation in Sztakipedia

into plain text and indexed by a Lucene instance. The currently edited document is also converted to plain text and used to search similar articles. If the retrieved articles have infoboxes on them—a fact provided by DBpedia—the hypothesis is that they will be applicable to this document as well. We have tried machine learning

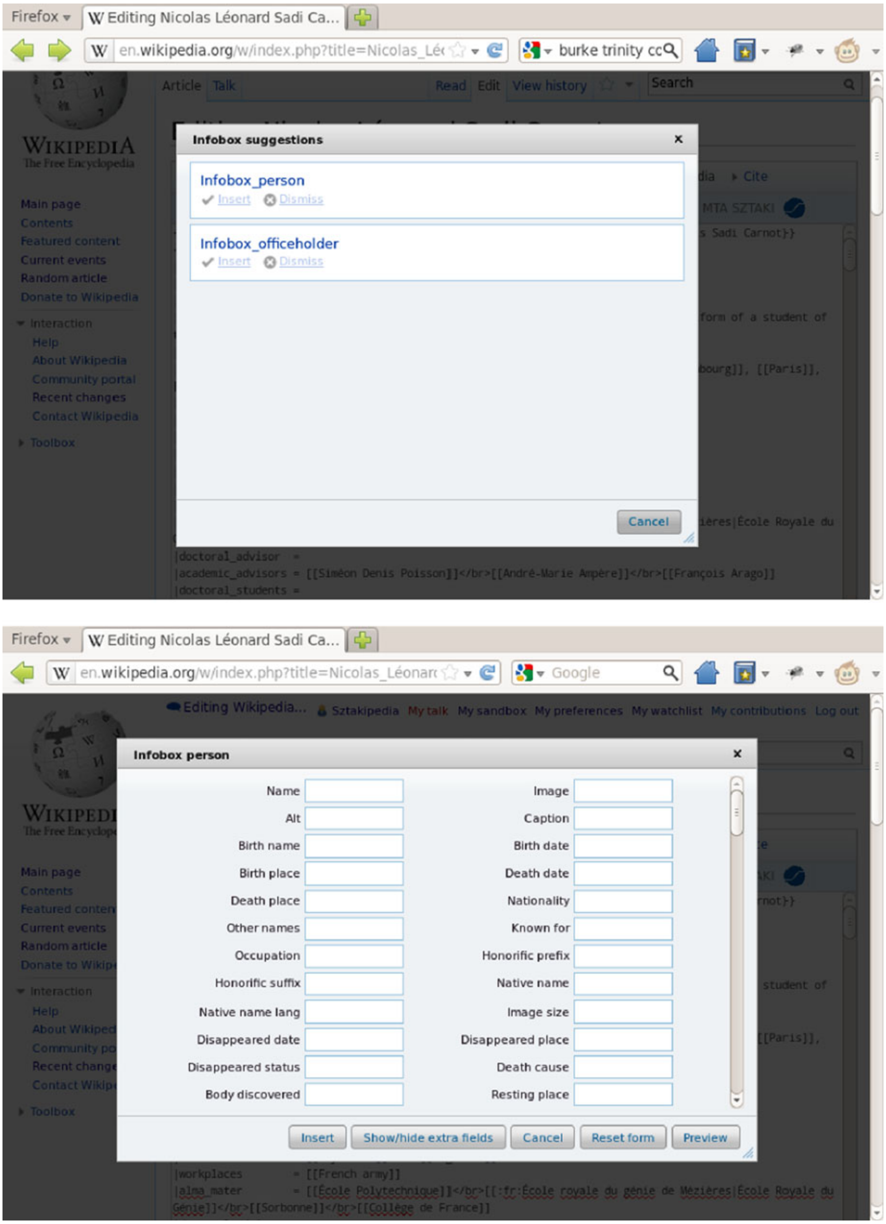



**Fig. 4.5** Infobox recommendation and filling assistant in Sztakipedia

techniques to recommend infoboxes and categories but the results were unsatisfactory and we also had to face serious technical problems—concerning mainly memory usage and speed—with a corpus this large. We could not conduct strict numeric measurements on the applicability of the infoboxes recommended by Lucene.

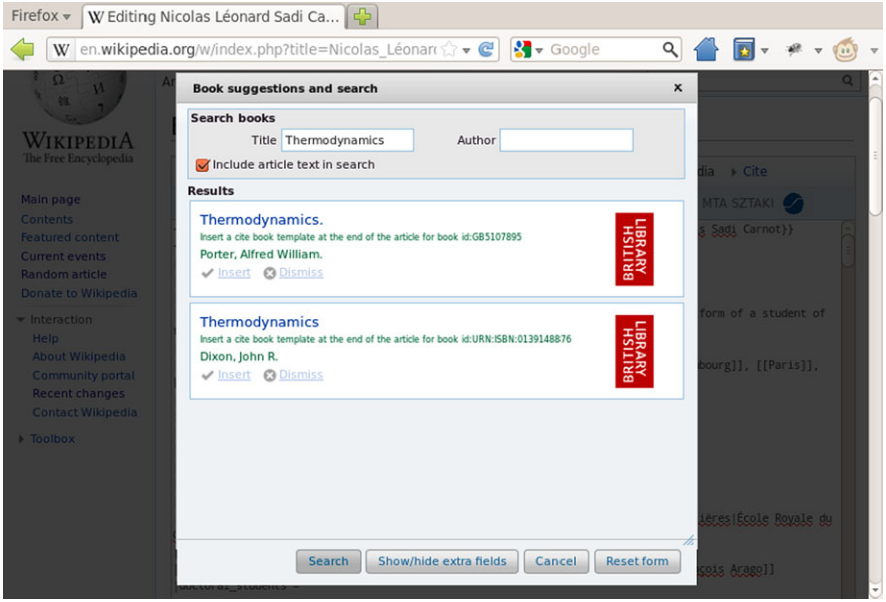



**Fig. 4.6** Book recommendation and search in Sztakipedia

However, user feedback indicates that in certain topics, like settlements and biographies, the Lucene recommendation works quite well—that is, the proper infobox is mostly in the top 3–5 recommendations. The recommendation of infrequent infoboxes is less robust in general, but many times it provides infoboxes previously unknown to the users which they usually consider as an added value. Infobox recommendation also has fill support that relies on DBpedia infobox data. For screenshots of the recommendations and the fill helper, see the middle row in Fig. 4.5.

#### 4.3.1.3 Category Recommendations

In general, categorization could be done in a very similar way to infobox recommendation. In the case of a Wikipedia of almost any given language however, this method did not prove to be precise enough. So to provide category recommendations we use another search engine, Yahoo's Build your Own Search Service (a.k.a. Yahoo BOSS). By searching for the most important phrases we gathered from our weight measure with the following query 'Category <important phrase1>, <important phrase2> . . .' we usually get good enough category recommendations.

#### 4.3.1.4 Source Citation Recommendation

Finally, there is a fourth kind of recommendation that is based on Linked Data: related literature. This feature is enabled by the fact that both British National Library

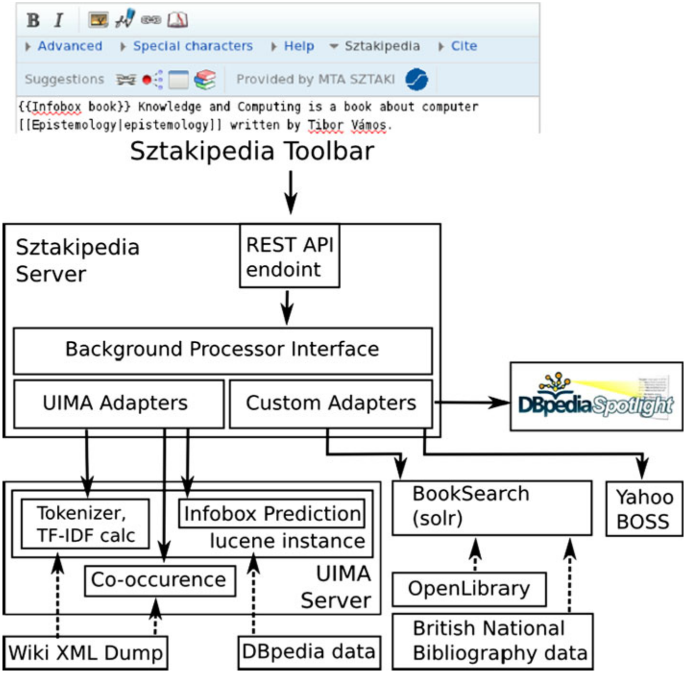



**Fig. 4.7** The architecture of Sztakipedia

(BNB) and OpenLibrary (OL) offer their data for download. Figure 4.6 shows two suggestions from BNB for the keyword "thermodynamics". BNB data is in RDF that one can use directly in an RDF store, while OL data is in JSON, which is also easy to process for a machine. We loaded both in a Lucene instance to make it searchable. Also it is good to know that many libraries offer a Z39.50 or Open Archives Initiative (OAI) interface.[41] The good thing with libraries is that the records they have are usually categorized and have keywords that are created by the enduring labor of generations of librarians. More specific topics are covered by only a few dozens of books; this makes it possible to offer a set of books that contains the one which is just in the article writer's mind.

#### 4.3.1.5 System Architecture

The architecture of Sztakipedia is depicted in Fig. 4.7. The user interface of the tool communicates with only one server-side endpoint. The mashup of the different sources happens at the server side. The main reason of this is that the system has to collaborate with many different interfaces, some of them not quite accessible from

[41] These are the old and new machine interface standards supported by most library systems.

browsers, like the ages-old Z39.50 library interface. But an equally important issue is that we need optimized performance for our application.

Behind Sztakipedia there is an Apache UIMA [11] server that can annotate any given text very quickly. Annotation means word importance detection for all words, finding the corresponding DBpedia entities, and also finding similar documents. The data needed for this processing are stored in a Lucene index that can be updated with new documents any time. This system is fed by the plain text version of all Wiki articles, plus DBpedia pagelink data.

This also explains why we decided to download some of DBpedia data and not use the API directly. Our application requires the lookup of many thousand words and their linking frequencies preferably under a second. Putting this load to an un-optimized endpoint would overload the system the first time of trying. Here lurks one of the greatest dilemmas of creating mashups.

### *4.3.2 Lessons Learned*

It is very convenient when a semantic data source is maintained by others; one does not have to care about the data updates, technical issues with the data sources, etc. On the other hand, the problem of quality of service is not quite solved in the Linked Data paradigm. And, of course, one does not have service quality that is enforceable by a contract when using a free service. On can perceive a tension in the minds of software developers and integrators when it comes to relying on an external service that is available for free: what if it goes down? In these situations it does not really matter that e.g. DBpedia has a great uptime. What matters is that when our team has contracted a customer to provide a service (e.g. Sztakipedia for a company's internal Wiki) under certain quality conditions, we should be able to rely on equally strict or stricter conditions from the services we use to provide ours. Otherwise, we will feel that our back is not covered, and there is a fear in everyone who maintains a system that if the third party service is down, we will not be able to do anything about it. This is why the downloadable DBpedia virtual machine is so important: one can use it locally, but still partly benefit from the service provider.

Furthermore, there are cases like the word importance measure combined from tf-idf and DBpedia link frequency, where the standard interfaces of linked data access just do not fit. In general, it is much more likely that a custom solution will be much faster than a SPARQL query for instance. However, the situation can very easily change with technology and QoS advancements.

Downloading and loading in the data in a customized system is problematic in a different way. In this case a regular nurturing of data is needed that can be very difficult. Consider for example how often Wikipedia changes. DBpedia Live[42] is able to follow these changes in almost real time, while one can only update one's own database when a new dump is created—once in every couple of weeks.

[42] http://live.dbpedia.org/.

This sequence introduced all the problems of the "download and use our linked data" approach. Some of the problems were solved when DBpedia Spotlight was introduced, but in general, we still have to do regular data maintenance.

## 4.4 Conclusion and Future Prospects

In this chapter we tried to introduce DBpedia and its richness to the reader. The famous Linked Open Data map rightly puts DBpedia at the center of the picture. We presented many applications and mashups that are using DBpedia, but these are only a small fraction of a large set of projects.

Many different projects explore similar areas like Named Entity Recognition, creating recommendations, semantic search, facilitated editing. However, we are sure that these are only the forerunners of more creative ways of using DBpedia, yet unknown. Consider, for example, the idea of finding the right and so-often missing column names for tables gathered from the web pages by a search engine [2].

We also presented Sztakipedia, a mashup application, in detail, that is—together with DBpedia Spotlight—trying to give something back to Wikipedia editors in return for great value they created by writing articles and thus enabling the DBpedia project.

**Acknowledgements** We are deeply indebted to Domonkos Tikk and Pablo Mendes for their valuable comments on the text. The authors were partially supported by the grant TÁMOP-4.2.2.B-10/1-2010-0009 of the Hungarian National Development Agency (NFÜ).

## References

1. Allemang D, Hendler JA (2008) Semantic web for the working ontologist: effective modeling in RDF and OWL. Morgan Kaufmann, San Mateo
2. Amin MS, Jamil H (2010) An efficient web-based wrapper and annotator for tabular data. Int J Softw Eng Knowl Eng 20(2):215
3. Auer S, Bizer C, Kobilarov G, Lehmann J, Ives Z (2007) DBpedia: a nucleus for a web of open data. In: 6th international semantic web conference, Busan, Korea. Springer, Berlin, pp 11–15
4. Baader F (2003) The description logic handbook: theory, implementation, and applications. Cambridge University Press, Cambridge
5. Becker C, Bizer C (2008) DBpedia mobile: a location-enabled linked data browser. In: Linked data on the web (LDOW)
6. Bizer C, Lehmann J, Kobilarov G, Auer S, Becker C, Cyganiak R, Hellmann S (2009) DBpedia—a crystallization point for the web of data. Web Semant Sci Serv Agents World Wide Web 7(3):154–165. doi:10.1016/j.websem.2009.07.002
7. Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J (2008) Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on management of data, SIGMOD'08. ACM, New York, pp 1247–1250
8. Chan B, Talbot J, Wu L, Sakunkoo N, Cammarano M, Hanrahan P (2009) Vispedia: on-demand data integration for interactive visualization and exploration. In: Proceedings of the 35th SIGMOD international conference on management of data. ACM, New York, pp 1139–1142

9. Cyganiak R, Jentzsch A. Linking open data cloud diagram. http://lod-cloud.net/
10. DiFranzo D, Graves A, Erickson JS, Ding L, Michaelis J, Lebo T, Patton E, Williams GT, Li X, Zheng JG et al (2011) The web is my back-end: creating mashups with linked open government data. In: Linking government data, pp 205–219
11. Ferrucci D, Lally A (2004) UIMA: an architectural approach to unstructured information processing in the corporate research environment. Nat Lang Eng 10(3–4):327–348
12. Giles J (2005) Internet encyclopaedias go head to head. Nature 438(7070):900–901
13. Heath T, Motta E (2008) Revyu: linking reviews and ratings into the web of data. Web Semant Sci Serv Agents World Wide Web 6(4):266–273
14. Héder M (2011) Integrating artificial intelligence solutions into interfaces of online knowledge production. ICIC Express Lett 5(12):4395–4401
15. Héder M, Farkas M, Oláh T, Solt I (2011) Sztakipedia: mashing up natural language processing, recommender systems and search engines to support Wiki article editing. In: AI mashup challenge 2011 at extended semantic web conference (ESWC 2011), Iraklion, Crete, Greece
16. Hogan A, Harth A, Umbrich J, Kinsella S, Polleres A, Decker S (2011) Searching and browsing linked data with SWSE: the semantic web search engine. Web Semant Sci Serv Agents World Wide Web 9(4):365–401. doi:10.1016/j.websem.2011.06.004
17. Jain P, Hitzler P, Yeh PZ, Verma K, Sheth AP (2010) Linked data is merely more data. In: Linked data meets artificial intelligence, pp 82–86
18. Kobilarov G, Scott T, Raimond Y, Oliver S, Sizemore C, Smethurst M, Bizer C, Lee R (2009) Media meets semantic web—how the BBC uses DBpedia and linked data to make connections. In: The semantic web: research and applications, pp 723–737
19. Korica-Pehserl P, Latif A (2011) Meshing semantic web and Web 2.0 technologies to construct profiles: case study of Academia Europea members. In: Networked digital technologies, pp 334–344
20. Mcguinness DL, Fikes R, Hendler J, Stein LA (2002) Daml+oil: an ontology language for the semantic web. IEEE Intell Syst 17(5):72–80
21. Mendes PN, Jakob M, García-Silva A, Bizer C (2011) DBpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th international conference on semantic systems (I-semantics), pp 1–8. doi:10.1145/2063518.2063519
22. Passant A (2010) dbrec: music recommendations using DBpedia. In: Proceedings of the 9th international semantic web conference on the semantic web, ISWC'10, vol II. Springer, Berlin, pp 209–224
23. Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval* 1. Inf Process Manag 24(5):513–523
24. Waitelonis J, Osterhoff J, Sack H (2011) More than the sum of its parts: Contentus—a semantic multimodal search user interface. In: Proceedings of workshop on visual interfaces to the social and semantic web (VISSW), co-located with ACM IUI, vol 13
25. Wood A, Struthers K (2010) Pathology education, Wikipedia and the Net generation. Med Teacher 32(7):618

# Chapter 5
# Mashups for the Web of Things

**Klemen Kenda, Carolina Fortuna, Alexandra Moraru, Dunja Mladenić, Blaž Fortuna, and Marko Grobelnik**

**Abstract** The Web of Things (WoT) together with mashup-like applications is gaining popularity with the development of the Internet towards a network of interconnected objects, ranging from cars and transportation cargos to electrical appliances. In this chapter we provide a brief architectural overview of technologies which can be used in WoT mashups with emphasis on artificial intelligence technologies such as conceptualization and stream processing. We also look at data sources and existing WoT mashups. In the last part of the chapter we discuss the architecture and implementation of Videk, a prototype mashup for environmental intelligence.

## 5.1 Introduction

The Web of Things [12, 15] is an emerging concept which extends already existing concepts such as the Sensor Web [21], where all sensor data and metadata would be published and available to anyone. The *things* themselves are everyday objects (i.e. coffee mug, chair, truck, robotic arm, etc.) containing a small computing and

K. Kenda (✉) · A. Moraru · D. Mladenić · B. Fortuna · M. Grobelnik
Artificial Intelligence Laboratory, Jožef Stefan Institute (JSI), Jamova 39, 1000 Ljubljana, Slovenia
e-mail: klemen.kenda@ijs.si

A. Moraru
e-mail: alexandra.moraru@ijs.si

D. Mladenić
e-mail: dunja.mladenic@ijs.si

B. Fortuna
e-mail: blaz.fortuna@ijs.si

M. Grobelnik
e-mail: marko.grobelnik@ijs.si

C. Fortuna
Communication Systems Department, Jožef Stefan Institute (JSI), Jamova 39, 1000 Ljubljana, Slovenia
e-mail: carolina.fortuna@ijs.si

B. Endres-Niggemeyer (ed.), *Semantic Mashups*,
DOI 10.1007/978-3-642-36403-7_5, 

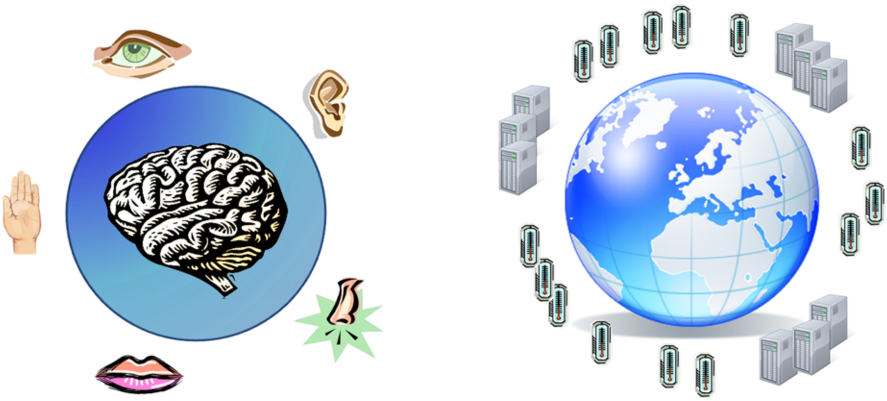


**Fig. 5.1** Illustration of the five senses analogy to WoT

communicating device. This device is most often a sensor node, however, it can also be an active or passive RFID tag in which case computing is done at the server. The *things* currently form isolated networks, controlled by different entities, and most often the data remain closed and are rarely used to full potential. Connecting (or federating) the islands of things using web standards is referred to as the Web of Things (WoT).

The mashups for the WoT, also referred to as physical mashups in [15], use raw or processed data coming from things, as well as already existing web data and services to build new applications. The development of such technology is expected to have a high impact on humanity, among others on efficiently servicing increasingly urbanized cities with food, transport, electricity and water in an environmentally sustainable way [24].

One way of looking at the Web of Things—the way we look at it in this chapter—is to see *things* as organs which detect stimuli. These are then sent via wireless or wired technology, typically on an IP/HTTP network, to processing and storage engines. These engines then crunch the received information and generate knowledge. Sometimes they can also trigger an action, such as sending a tweet. This is somewhat similar to how we, humans, function: we have five senses which are perceived by corresponding organs, then the stimuli are sent to the brain via the nerves, finally the brain processes these stimuli. The result is most often knowledge, and sometimes also actions can be triggered: the brain transmits commands via the nerves to the muscles which then contract and cause moving of hands, legs, talking, etc. One distinction is that while in the case of the humans the sensors and processors are spatially close to each other (e.g. nose and brain or ears and brain), in the case of WoT we may be looking at a global distributed system—see Fig. 5.1.

This chapter is structured as follows. Architectural considerations related to mashups for the Web of Things are presented first, addressing the fact that we are dealing with the network of things, providing conceptualization of the domain and discussing stream data processing. WoT mashups are presented next covering physical and non-physical data sources and presenting some of the existing related

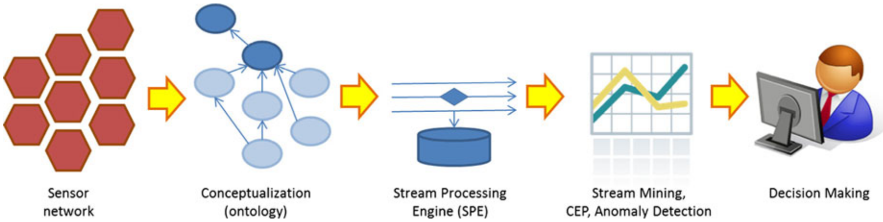



**Fig. 5.2** Technological pipeline for WoT. Data obtained from sensor network are annotated and enriched using ontology, sent through stream processing engine, processed using stream mining or complex event detection or similar and delivered to a decision maker

mashups. The discussion highlights the main components of mashups and their owners. Then, Videk, a mashup for environmental intelligence is presented, wrapping up the chapter with a concrete example of a mashup for the WoT. Finally, we summarize the chapter.

## 5.2 Architectural Considerations

The technological pipeline for the WoT, as we see it in this chapter, is presented in Fig. 5.2. The raw data and metadata coming from the network of things can be annotated and enriched—we refer to this as conceptualization—it can be stored using specific approaches for streaming and it can be processed using techniques such as stream mining, event and anomaly detection. WoT mashups can take and use the data at any of these stages.

### *5.2.1 The Network of Things*

The *things* are objects that can be digitally identified by some code such as Electronic Product Code (EPC), Radio Frequency IDentification (RFID), Near Field Communication (NFC), Internet Protocol (IP) v4 or v6, etc. Using these digital identities, things can then be observed by tracking in production plants, warehouses, etc.; by observing usage patterns, by observing their context, etc. In this chapter, we focus on *things* that feature sensors and an embedded device, mostly because the mashup we develop addresses environmental intelligence based on sensor data streams.

The embedded device typically contains four modules: the central processing unit and memory, the communication module, the sensor/actuator and the power source (see Fig. 5.3). The CPU controls the embedded device: it tells the sensors to capture data, it sends the data to the storage and/or to the communication module which then transmits them to the destination. A sensor is a device that measures physical phenomena and converts them to a signal that can be read by an observer,

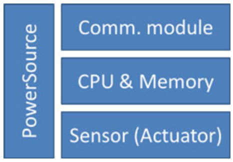



**Fig. 5.3** Functional components of a typical sensor device consisting of the central processing unit and memory, the communication module, the sensor/actuator and the power source

or, in our context, by a computer. The communication module typically uses wireless transmission (i.e. IEEE 802.15.4). The operation of the embedded device is constrained by the available power.

Different configurations of networks of things are used in extremely diverse applications [1], however, winning applications that scale are yet to emerge [10, 24]. Currently only islands of things exist and only in the next years large systems (20,000 nodes and more) are to be expected [22].

### *5.2.2 The Conceptualization of the Domain*

For small and medium size isolated projects it can be relatively straightforward to know which stream of data measures a given property. Traditional database tables can work well in such situations. However, if we are talking about web scale and are aiming for interoperability, some conceptualization of the WoT domain is needed.

Knowledge about sensors needs to be encoded and structured so that it can be used to its full potential. Raw numbers are as meaningless for the user as no numbers at all. Additional information such as the phenomena they are measuring, the units of measurement, the location of the sensor node, etc. are needed to accompany the numbers. For instance, if we wanted to know the amount of rain, we should be able to recognize that raindrop, rainfall, and precipitation belong to the same physical phenomena and that all such sensors are a good source for our query. If we were interested in the outside temperatures in the morning, we should be able to infer that a sensor node that is positioned in a stable, is not a good source for us, because it is measuring the temperature inside. If we wanted to find out what is the air pressure in our city, we would need the system to be able to tell which geographical coordinates of a sensor node belong to the area (inverse geocoding). The conceptualization of the domain refers to modeling all this knowledge in a standard way. By using standards also interoperability between different systems can be achieved.

To address the lack of interoperability between sensors, the application of Semantic Web technologies has been proposed [23]. Efforts within the Sensor Web Enablement initiative[1] from the Open Geospatial Consortium (OGC) go in the same direction by abstracting from XML-based serializations:

[1] http://www.opengeospatial.org/ogc/markets_tehcnologies/swe.

- SensorML[2]—models and schema for sensor systems and processes surrounding measurements
- Observations & Measurements (O&M)[3]—models and schema for packaging observation values

Such standards have also been mapped into an ontology by the W3C Semantic Sensor Network Incubator Group.

The conceptual model in SensorML contains four main elements: Process Model, Process Chain, System and Component. All of them inherit properties of a more general concept, Abstract Process, which is also derived from Abstract Feature. The general model that SensorML provides is a skeleton for describing sensor systems and processes, but for achieving interoperability in the Sensor Web, SensorML should be complemented with some semantic specification (e.g. ontologies). On the other hand, the SensorML encodings follow the Object–Association–Object pattern, facilitating the association with RDF and Semantic Web.

In order to overcome one of the OGCs Sensor Web Enablement shortcomings, that is, the lack of semantics and reasoning, the authors of [5] are proposing a Semantic Sensor Network ontology. The SSN ontology is the result of a collaborative ontology engineering effort and is based on several pre-existing sensor ontologies [4].

The ontologies describing sensor networks may contain concepts for defining technical aspects (calibration, temporal resolution-sampling frequency, accuracy, observed property), information about access to the sensor for control and configuration, location, meaning of data. Four main clusters (groups) of concepts have been identified [4]:

- domain of sensing (contextual information)
- sensor description (sensor node level)
- physical characteristics and location description
- and observation model (functions and processes of obtaining the measurements)

In the later developed SSN ontology, the concepts are organized into 10 modules, providing a more differentiated conceptual representation of concepts. However, the usage of the ontology is presented from similar perspectives. While the SSN ontology provides the main concepts for the sensor network domain, the application domain concepts are left unspecified, requiring the usage of additional ontologies or the extension of SSN. Such ontologies are needed for geospatial information (e.g. Basic GeoWGS84 Vocabulary,[4] GeoNames[5]), time representation (W3C time ontol-

[2]http://www.opengeospatial.org/standards/sensorml.

[3]http://www.opengeospatial.org/standards/om.

[4]http://www.w3.org/2003/01/geo/.

[5]http://www.geonames.org/.

ogy[6]), properties observed (SWEET ontologies[7] or application-specific extensions of SSN).

### 5.2.3 Stream Data Processing

The avalanche of data which information systems had to deal with in the last years influenced their evolution and characteristics. Continuous on-time processing of incoming data streams imposed particular requirements, which traditional Database Management Systems (DBMS) were not able to fulfill. In consequence, new tools have been developed that are able to process multiple data sources, often streams, in a timely fashion in order to extract relevant information [6].

In [27] the authors identify requirements for real-time stream processing based on the assumption that efficient stream processing requires the ability to handle large amounts of messages in a stream with low latency.

*Keep the data moving*. A real-time stream processing system should process the data “in-stream” (i.e. keep its “current state” in working memory), without any requirement to store them. Storing (i.e. writing to disk) significantly increases latency. The system should also use an active processing model, which means that it should issue alerts and actions rather than wait to be polled for the results (see Fig. 5.4).

*Handle stream imperfections*. Data in a stream might be delayed, missing or literally “out of order”. Computations should not be blocked by such situations; the system should be able to handle messages by their time stamp and not only by the time of their arrival.

*Integrate stored and streaming data*. Many applications need initialization using historical data from the stream or maybe users might want to test their new algorithms on such historical data. It should be seamless to switch from historical data to live feed.

Stream-processing applications should be able to process high volumes of data with very low latency. This means processing hundreds of thousands of messages per second with micro/millisecond-range latency. They should be able to take advantage of modern multi-core/multi-processor computer architecture and be able to recover from crash in real-time. The authors also suggest that real-time stream processing engines should implement an extended SQL language to handle streams.

Typical implementations of stream processing systems include (1) traditional DBMS system, (2) rule engines and (3) specialized stream processing architectures as depicted in Fig. 5.5.

DBMS systems are widely available and well known to developers. They have proved to be a reliable storage for large datasets. Main-memory DBMS can even provide much better performance than traditional ones, when avoiding writing to the disk for most operations, given sufficient main memory.

[6] http://www.w3.org/TR/owl-time/.

[7] http://sweet.jpl.nasa.gov/.

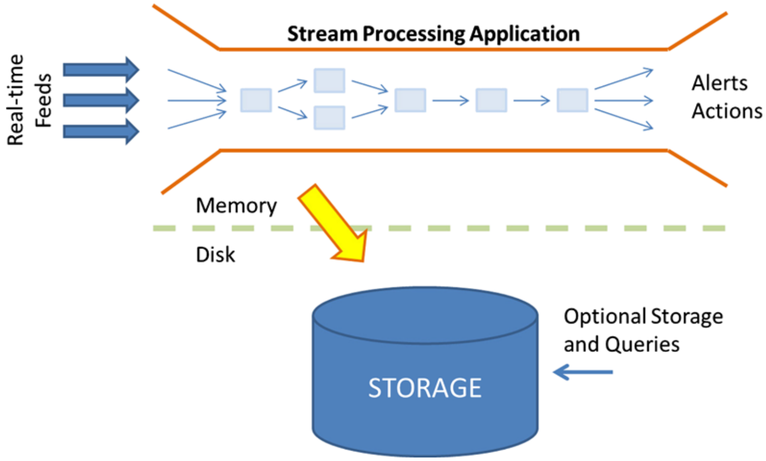



**Fig. 5.4** "In-stream" processing with optional storage [27]

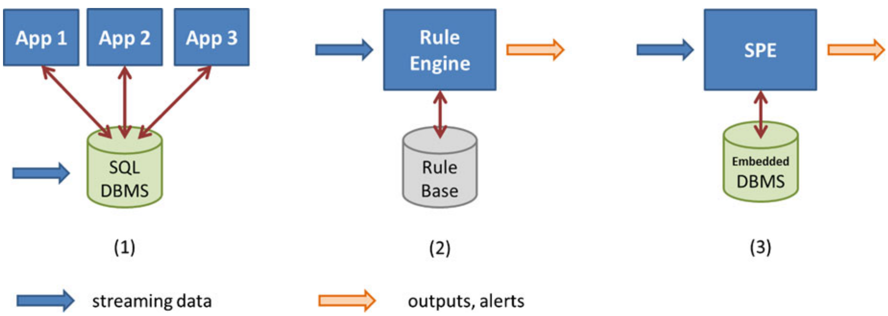



**Fig. 5.5** Three typical implementations of stream processing systems [27]: database management system, rule engine and specialized stream processing

Rule engines have been known from the early 1970s. Given the set of rules, which are typically formed in condition/action pairs, they only watch incoming stream for any conditions of interest and when these conditions are met, they trigger an action.

Stream processing engines (SPE) are specifically designed to deal with streaming data. They perform SQL-like queries on the streaming data without necessarily storing them. They use specialized primitives and constructs to express stream-oriented processing logic. States of the stream can be expressed with different operators and can be stored in a traditional way.

#### 5.2.3.1 Complex Event Processing (CEP)

In [8] the term *event processing* is coined to "any form of computing that performs operations on events", where an event can be anything that has happened or is perceived as having happened.

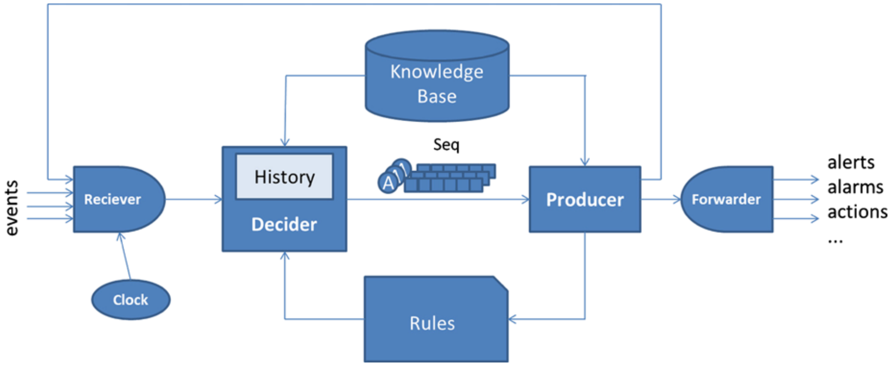



**Fig. 5.6** The functional architecture of a complex event processing system [6]

A common characteristic of event processing applications is to continuously receive events from different event sources (e.g. sensors, software modules, blogs, etc.). The central module processing the events, called the CEP engine, detects event patterns from the incoming data streams and outputs the detected or predicted complex events which can be further used by other event consumers, or returned as an input to the CEP engine.

The event patterns role is to specify how the incoming events should be processed in order to extract relevant information. The language used to define these patterns should have the ability of specifying complex relationships among events flowing into the CEP engine. Different types of languages exist and they can be classified by their style in two main categories: stream-oriented and rule-oriented [8]. The stream-oriented languages specify operations for processing the input streams (such as filtering, joining or aggregating) in order to obtain some other output streams. The rule-oriented languages specify rules for processing streams, clearly separating the triggering conditions and the actions to be taken when the conditions are met (the conditions represent event patterns). Stream-oriented languages are typically used within DBMS, while rule-oriented languages are used in CEP systems.

The main components of a CEP engine are illustrated by Fig. 5.6 and can be summarized as follows. When a new event is detected by an event observer, it is sent to the CEP engine by the *Receiver* component, which acts as a wrapper for the incoming stream of events. Depending on the application, the Receiver may (1) act as a demultiplexer by receiving events from multiple sources and sending them further one by one, (2) be connected to an internal clock for periodic processing (e.g. events may be processed every hour and not immediately when they are generated), (3) perform some pre-processing of the events.

The CEP engine has been divided in two main sub-components: the *Decider* and the *Producer*, for distinguishing between two main phases of event processing: the detection phase and the production (result) phase. The role of these two components is to process the incoming events according to a set of *Rules*. From a logical point of view, a processing rule is defined by two components ($C \rightarrow A$): a condition $C$ and an action $A$. The condition specifies the event pattern (e.g. sensor reading exceeding some threshold) that is continuously checked by the *Decider* on the arrival of new events. When an event pattern is detected, the action corresponding to it is sent to the *Producer*, which generates the result (e.g. an alarm, creation of a new complex event) for the event sequence that triggered the specific rule. The result produced is sent to event consumers (through the *Forwarder*), or it can also be sent internally, to be processed again (recursive processing). The *Knowledge Base* component is an optional component that can store the static information needed for event processing.

#### 5.2.3.2 Stream Mining

Data stream mining is the process of extracting knowledge structures from continuous rapidly changing streams of data. Stream mining found its solutions in well-established statistical and computational approaches that can be categorized into *data-based* and *task-based* ones [13]. Data-based solutions focus on examining a subset of the whole dataset or to reduce it vertically (features) or horizontally (number of records) in size. Task-based solutions engage techniques from computational theory that have been adapted to achieve time and space efficient solutions.

*Data-based techniques* include sampling, load shedding, sketching, synopsis of data structures and aggregation. Both sampling and load shedding have a problem that they are dropping out chunks of data streams that might represent a pattern of interest in time series analysis. The major drawback of sketching (randomly projecting to a subset of features from a stream) is its accuracy. Creating synopsis from data may include wavelet analysis (Fourier transform), histograms, quartiles and frequency moments. When using these techniques one needs to be aware of the inaccuracies that might occur due to incomplete representation of the data. With aggregation statistical measures, such as means, variance, minimum, maximum, count, etc. are computed. These measures can be used by mining algorithms. Again, such a method does not perform well with highly fluctuating data distributions.

*Task-based techniques* modify existing techniques and introduce new techniques for stream processing that are able to cope with the computational challenges of data stream processing. These include approximation algorithms, sliding windows and algorithm output granularity. The intuition behind sliding windows technique is in the fact that user is more concerned with the analysis of most recent data. Detailed analysis is done over the most recent data items and a summarized versions of historic data. The sliding windows approach (see Fig. 5.7) is widely implemented and can be combined with data-based techniques for even higher efficiency.

For example, summaries in the form of aggregates or synopsis analysis can be performed on a sliding window over the most recently arrived data. In [9], the authors propose using the Discrete Fourier Transformation (DFT) on a sliding window to achieve fast subsequence matching. Instead of keeping the whole set of measurements, they suggest performing transformation into frequency space and keeping the first few most relevant harmonics. In this space it is also easier to define the similarity measure between time series. In the frequency space similarity can be

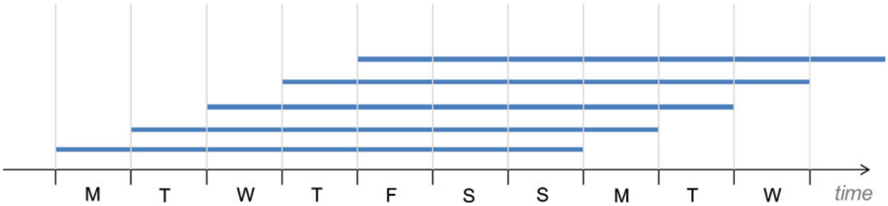



**Fig. 5.7** Illustrating usage of weekly sliding window on data, sliding on a daily basis (M—Monday, T—Tuesday, Thursday, W—Wednesday, F—Friday, S—Saturday, Sunday)

computed with one of the standard distance metrics. Instead of DFT other similar methods can be used for feature generation.

Possible mining techniques on preprocessed data include clustering, classification, frequency counting and time series analysis [13, 14].

#### 5.2.3.3 Anomaly Detection

Anomalies detected in data from sensor networks can have two meanings: either sensors are faulty, or they are detecting events (such as intrusions) that are interesting for analysts [3]. Sensor data can also be a subject of noise from the environment or nature of the sensor or can have missing values in the data collection process. In the case of streaming data anomaly detection techniques are required to perform in an on-line approach, where there are many resource constraints. The techniques need to be lightweight to permit fast processing and due to the nature of the sensor data they also need to distinguish between interesting anomalies and noise/missing values. The key to anomaly detection and successful stream mining is proper feature engineering. Anomalies are detectable if data instances are represented in an informative feature space. Often it is hard to precisely define what an anomaly is, as there are many possible types and there is no universal technique that would detect them all.

Methods that have been used for anomaly detection in sensor data [3] include Bayesian networks, rule-based systems, parametric statistical modeling, nearest neighbor-based techniques and spectral techniques.

## 5.3 WoT Mashups

The data sources for WoT mashups can be classified into two categories: physical data sources (typically data from the network of things) and other existing data sources on the web (social networks, encyclopedia, etc.). A WoT mashup integrates at least one of the physical sources and can enrich the data with other sources. For instance, in [17] the authors show how physical data about deforestation can be linked to other data, such as population density and income level, to explore possible correlations. The Web of Data or Linked Open Data are key resources for such integration and enrichment.

### 5.3.1 Physical Data Sources

Two types of physical data sources have been identified based on the way they are encoded and accessible. The first type represents standard-based physical data sources which are published according to the OGC SWE (Sensor Web Enablement) standards and the second type represents non-standard data sources which are published using a custom API and a popular non-structured format such as text or CSV.

#### 5.3.1.1 Standard-Based Data Sources

The OGC SWE standards for accessing physical data are:

- Sensor Observation Service (SOS)[8]—standard web interface for accessing observations
- Sensor Planning Service (SPS)[9]—standard web interface for tasking sensor systems and model and requesting acquisitions
- Sensor Alert Service (SAS)[10]—web interface for publishing and subscribing to sensor alerts

The input/output format of those services is encoded in a structured form according to the OGC specifications: SensorML and Observations & Measurements (O&M). The basic scenario for obtaining sensor measurements using SWE is depicted in Fig. 5.8. The figure shows sensors in the upper right corner. Sensors push their observation results either to a database (which is accessed by the SOS service) or directly to the SOS service (Transactional-SOS in this case). Sensors are registered, using SensorML, at the catalog service. When the user queries the catalog service, it responds with a list of SOS service instances that fulfill the requirements. Eventually the user binds the SOS and retrieves the observation data, encoded in O&M format. More complex situations are handled by SPS and SAS scenarios [25].

A comprehensive list of SOS services around the world may be found in [28]. Some of them are:

- Envision Project Use Cases[11]
- National Data Buoy Center[12]
- Northeast Coastal and Ocean Data Partnership[13]

---

[8] http://www.opengeospatial.org/standards/sos.

[9] http://www.opengeospatial.org/standards/sps.

[10] http://www.ogcnetwork.net/SAS.

[11] http://www.envision-project.eu/ (Scenarios).

[12] http://sdf.ndbc.noaa.gov/sos/.

[13] http://www.necodp.org/.

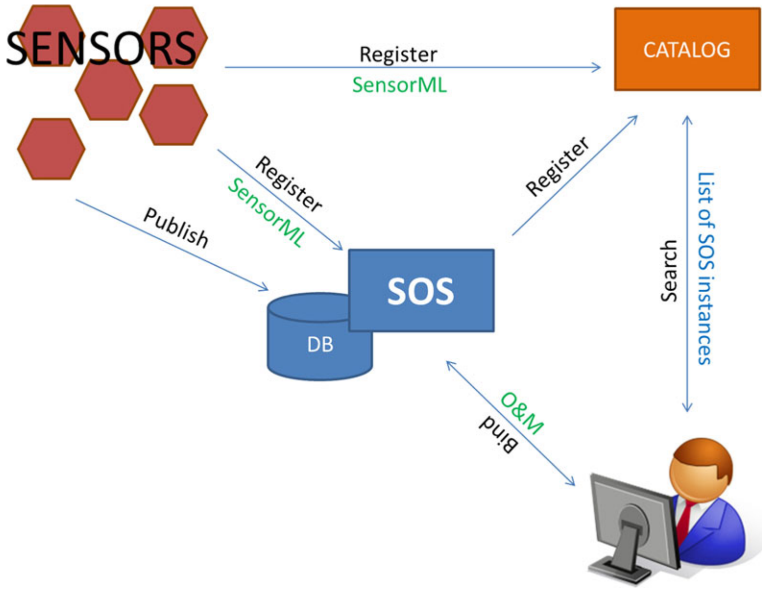



**Fig. 5.8** Simple scenario for obtaining sensor measurements using SWE, where sensors are registered in a catalog and push their observation results either to a database or to a sensor observation service

The SWE initiative is a step into the *universally interconnected WoT* direction, but only a minority of useful data sources tends to use the standards provided by the OGC. OGC standards, however, cover complex scenarios and sometimes represent too much administrative overhead for the data providers.

#### 5.3.1.2 Non-standard-Based Data Sources

Many services use simpler proprietary protocols that require the mashup developer to prepare specific adaptors. Some of the global sensor catalogs include Cosm,[14] SensorPedia,[15] SwissExperiment. Other emerging platforms in this area are nimbits,[16] open.sen.se[17] and Evrythng.[18]

Single testbeds might also have their own custom interface. For instance, the JSI sensor testbed[19] features a custom API and provides open data. The API includes

[14]https://cosm.com/ (former Pachube).

[15]http://www.sensorpedia.com/.

[16]http://www.nimbits.com/.

[17]http://open.sen.se/.

[18]http://evrythng.com/.

[19]http://sensors.ijs.si/.

a complete set of requests: sensors-on-node and node-from-sensor (which link sensor node with sensors integrated in it), sensor-type (which links a specific sensor type with the sensor), current-state (which reports on the configuration of the sensor network and last measurements), get-measurements (which retrieves measurements of the sensor), get-aggregates (which retrieves sensor aggregates of the sensor measurements). Within the JSI stream processing service (SenseStream) measurements are kept in the main memory only—in an expandable FIFO buffer with run-time defined time window. On the stream aggregates and other synopsis parameters are calculated, which are stored in a CurrentState object, which is part of every sensor record in the store. Current state stores IDs of measurements from the FIFO buffer for the sensor, enabling synopsis parameters to be calculated, while aggregates are being calculated on-line (updated with each incoming measurement). When the time window closes, the aggregates and synopsis parameters are stored in a smart cache, which keeps the last records in main memory and older ones on the disk.

The current state of the sensors can be retrieved with http://sensors.ijs.si/xml/current-state. The XML structured answer includes:

- **nodes**
  - records (int)—*number of nodes*
  - node
    - id (int)—*internal id of the node*
    - name (string)—*identifier of the node*
    - latitude (double)—*latitude of the node*
    - longitude (double)—*longitude of the node*
    - **sensor**
      - id (int)—*internal id of the sensor*
      - sensortypeid (int)—*internal id of the type of the sensor*
      - featureofmeasurement (string)—*human readable feature of measurement*
      - unitofmeasurement (string)—*human readable unit of measurement*
      - lastmeasurement (double)—*value of the last measurement from the sensor*
      - measurementtime (time)—*MySQL formatted timestamp of the measurement*

### *5.3.2 Non-physical Data Sources*

Non-physical data sources in a mashup scenario help developers to enrich the physical data and add additional value to it. Sensor mashups typically use an on-line maps service as a basis of their GUI. Widely available platforms are Open Layers, Google Maps, Yahoo Maps and MSN Virtual Earth. There are many others that might include better map sources on a local level. Each service has its own API and most of them can be used to show positions of sensors and their environmental context. They also provide events and actions so the user can interact with the map.

In the "maps" context inverse geocoding services are typically very useful. Sensor nodes usually include data about their position, therefore an inverse geocoding service such as Geonames can provide the name of the nearest place for that particular position. Furthermore, the names or near towns, municipalities, countries, etc. can be provided in a straightforward manner. Places, cities, countries can be further linked with Wikipedia, Panoramio, DBpedia and other sources to automatically retrieve additional information. Linking Open Data (LOD)[20] can be particularly useful for collecting additional context. If we choose Geonames as the entry point, records there can be linked to Freebase and DBpedia within the LOD cloud.

Many news items, photographs, tweets and other pieces of information are geotagged on the web nowadays. One can retrieve photos from neighboring location from Panoramio, Flickr, Picasa or other similar services or enrich the data with news items or tweets. There is a whole bunch of interesting applications with combining sensor and textual data from social media on event detection, enriching sensor data, etc.

### *5.3.3 Existing Mashups*

There are many mashup applications that deal with data from real objects published on the web. This section focuses on bigger projects which mostly emerged from academic efforts. Examples of such mashups are EPCmashup, SensorMasher, Sentient Graffiti, Cosm (former Pachube), SensorMap and also Google Maps.

The Google Maps from the mashup perspective combines several sources of data including real-time traffic monitoring, weather, webcams, video, Wikipedia and public data. It is able to provide a view of the city (i.e., New York) including information on work on public roads, traffic congestion, images of locations, etc.

SensorMasher[21] [18] publishes sensor data as Web data sources which can then easily be integrated with other (linked) data sources and sensor data. Raw sensor readings and sensors can be semantically described and annotated by the user. These descriptions can then be exploited in mashups and in linked open data scenarios and enable the discovery and integration of sensors and sensor data at large scale.

Cosm provides a platform for crowd-sourced sensor streams and offers simple services on top of these. It can track device's state and location, add alerts and notifications, it enables the discovery of data nearby and different visualizations of current and historic data.

Sensorpedia[22] is a similar platform for crowd-sourced sensor streams. It includes a Google Maps interface enabling sensor discovery, some clustering algorithms and is able to visualize the sensor data.

---

[20] http://linkeddata.org.

[21] http://sensormasher.deri.org/.

[22] http://www.sensorpedia.com.

EPCMashup [15] is a research prototype based on simulated data in the context of supply chain management using RFID (Radio Frequency IDentification). It is able to receive EPCs (electronic product codes) from readers, show the history of its transport on the map and in the calendar, it is able to request pictures of the stock using webcams. In case of a theft it is able to issue alerts with snap-shots and product info.

In [7] Deusto Sentient Graffiti (DSG) a context aware mobile mashup for the ubiquitous web is introduced. In DSG the mobile clients send metadata and context data to the server. When the mobile client sends a query, these data are used to return a context aware mashup. The system also uses an inference engine which operates on a knowledge base populated with metadata. According to a set of rules it uses the knowledge to annotate associated surrounding resources available under their current contextual conditions. Another context-based mobile mashup is introduced in [20]. The mobile phone acts as a gateway that collects raw sensor data, pre-processes them and extracts user-centric contexts. The context management platform stores all contexts according to the redefined context ontology, detects the inconsistent information, deduces the high-level knowledge using heuristic rules and provide web service application programming interfaces (APIs) to a mashup server.

The SensorMap[23] [20] portal and tools allow users to make queries over live data sources and provide mechanisms to archive and index data, process queries, and aggregate and present results in geocentric web interfaces. The SensorMap architecture has three components: the GeoDB storing sensor metadata, the DataHub that handles real-time data publishing and the Aggregator which creates icons representing sensors which the users can then mashup with maps.

### *5.3.4 Discussion*

According to [16], mashups do not require complex programming and middleware technologies as they should enable users to create new user interfaces by reusing existing user interface artifacts with high-level scripting languages such as HTML and JavaScript.

In the case of WoT, the mashups can use raw data coming from the network of sensors, provided the owners of these networks permit this by exposing an API to these data. However, raw sensor data are often not useful for more than being fed into a chart, which is why most mashups are likely to pull in processed data. For instance, there can be conceptualization services that take the sensor data, annotate them, possibly enrich them, and expose the resulting data. The same is valid for stream processing, mining and other services. Therefore, if we look at the system architecture behind the WoT mashups depicted in Fig. 5.9, it is very likely that, as the field develops and matures, different components will be implemented and

[23] http://atom.research.microsoft.com/sensewebv3/sensormap/.

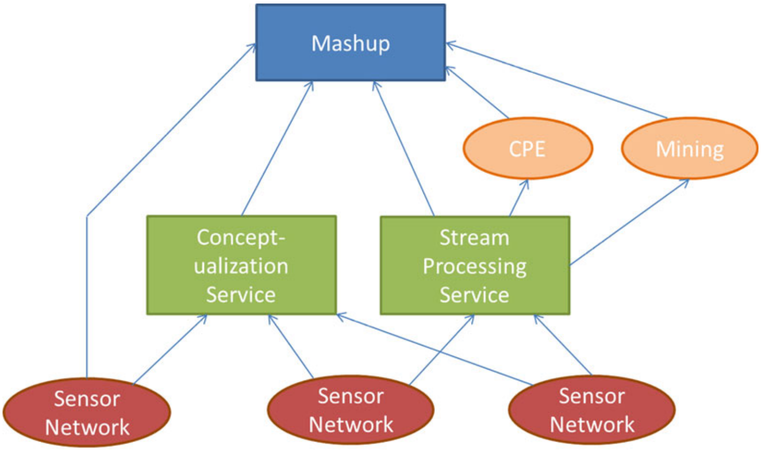



**Fig. 5.9** Architecture of distributed services behind WoT mashups

maintained by different owners similarly to how the systems behind existing non-WoT mashups are today. However, there can be one owner of all components, this is mainly because the lack of appropriate data forces developers of services to deploy their own sensor solutions—typically of smaller scale. Videk, the environmental intelligence mashup described in the next section, is an example of the former approach.

## 5.4 Videk—Mashup for Environmental Intelligence

Videk is a mashup that enables users to visualize sensor data from the environment on the map. A user can follow the current state of the system with latest measurements or can view history of the measurements with different aggregates for different time periods in real time. The mashup enriches the data and metadata from several external sources and enables the user to better understand the measurements.

### *5.4.1 System Architecture*

The architecture of Videk is depicted in Fig. 5.10 and consists of four main components: the sensor data, the external sources, the mashup server and the user interface.

The Mashup Server acts as an enabling platform by interfacing with sensor and external data sources as well as with the users. It mashes up raw and processed sensor data with external sources of data and provides a GUI for human end-users and

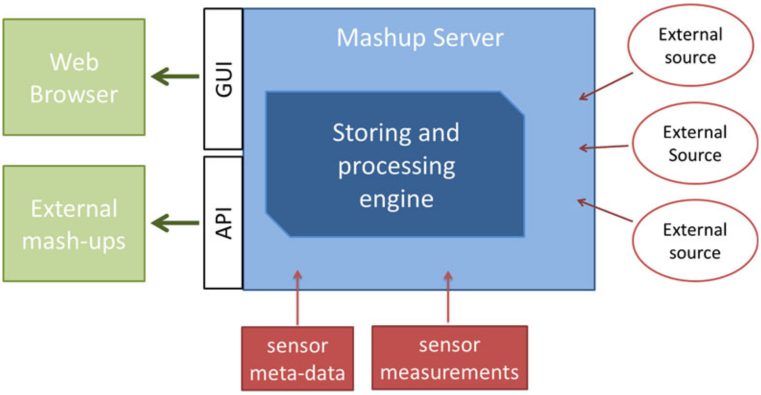



**Fig. 5.10** System architecture of WoT mashup consisting of the sensor data, the external sources, the mashup server and the user interface

an API for external mashups. At the core of the system is the Storing and Processing Engine which transforms the sensor data into useful information and knowledge. The engine stores, indexes and aggregates new sensor measurement data. For instance, it computes aggregates such as average, maximum, minimum and it can extract rules and export them as RuleML [2].

By external users we refer to humans and machines. For the human users, a GUI using the mashed-up data and knowledge is available. For the machines, the exposed API allows the system to be plugged into other mashups therefore making it another piece in the ecosystem.

The architecture of Videk is designed in a way that allows organic growth by adding new external and sensor data and updating the user interface to support new sources. The Storage and Processing Engine supports addition of plug-ins to add functionality to the data processing pipeline.

The system is built so that the need for human involvement with respect to data or metadata provisioning from sensors is reduced to a minimum. It achieves this by accommodating automatic collection of sensor metadata using a custom Device Identification Protocol (DIP). Based on the collected data it performs processing and pulls relevant information from external data sources. The metadata contains information such as a unique ID of the sensor node, GPS coordinates, configuration (i.e. what type of sensors the node features), the unit of measurement for each measurement stream, sampling frequency and accuracy.

All the external services and data sources are interrogated and mashed up starting from the information acquired from the sensors in the form of metadata. For instance, based on GPS coordinates, the system queries the Geonames service to retrieve the name of the geographical place where the sensor is located, or based on the sensor type (e.g. air pressure), details on how such sensors work or what are expected values that can be pulled from Wikipedia.

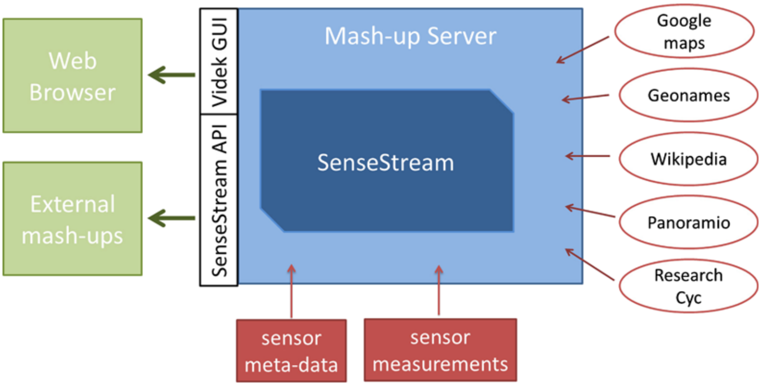



**Fig. 5.11** Implementation of the Videk system with SenseStream as the storage and processing engine and using Google Maps, Geonames, Wikipedia, Panoramio and ResearchCyc as external sources

## *5.4.2 System Implementation*

Videk is modular. Therefore it can constantly evolve by adding features and external sources. The current implementation (see Fig. 5.11) uses sensor data collected from VESNA [26] sensor deployments across Slovenia and Google Maps, Geonames, Wikipedia, Panoramio and ResearchCyc as external sources. The Storage and Processing Engine is called SenseStream while the Mashup Server is Apache with some custom Java and PHP code. The GUI uses jQuery for data manipulation, event handling and Ajax interactions.

### 5.4.2.1 SenseStream

SenseStream is a stream mining and event detection engine which powers Videk. SenseStream is built on the top of QMiner that is using an in-house C++ data mining library. The QMiner architecture is depicted in Fig. 5.12. The central part of the system consists of two components: Data Layer and Mining Algorithms [11]. At the bottom of the architecture diagram is a set of data sources, for example sensor measurements, sensor metadata, text documents and images. Each data source requires an adaptor, which maps the data source to a common interface. Adaptors for several standard data sources are provided: a directory of text documents or images, a website, unique users accessing the website, etc.

**SenseStream Architecture** The Data Layer provides unified access to all the data sources from the higher architectural layers, and includes an integrated inverted index and multi-modal feature extractors. The integrated inverted index is used to provide faceted search functionality over the records from the data sources. Integrated multi-modal feature extractors provide functionality for extracting feature vectors

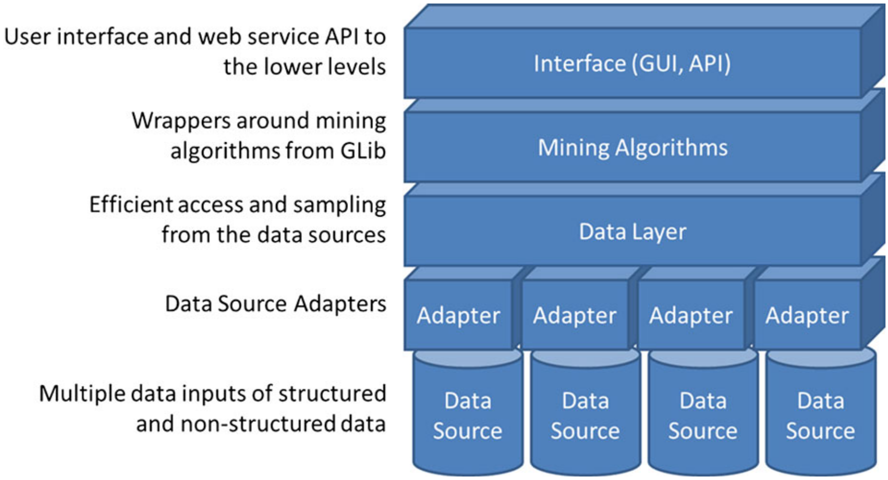



**Fig. 5.12** QMiner architecture, where the central part of the system consists of two components: Data Layer and Mining Algorithms

from raw data provided by the various data sources. Examples of feature extractors would be the vector-space model for text data, visual words for images, or time series statistics for sensor measurements. Feature extractors provide an abstraction layer required by machine learning algorithms.

The measurement feed is processed in real-time. On arrival, each measurement is put into a FIFO queue, and used to update the running aggregate for various time windows. The sizes of time windows are provided as input parameters. On specified intervals (e.g. minute, hour, day, week), aggregates are stored and indexed using their value, time, period, aggregation type, measured phenomena and sensor. The SenseStream API offers browsing, querying and mining of all stored measurements and their aggregates. The results can be exported using Linked Data[24] standards (RDF, RuleML).

The current implementation of SenseStream uses sensor metadata and measurements as data sources. On the Data Layer depicted in Fig. 5.13, the implementation has six stores, two for metadata (sensor node and sensor type), two for measurements (sensor measurements) and aggregates (derived from measurements) and one which essentially connects them (sensor). An additional store in the schema is able to store basic event information that can be used for additional analysis of the data and generating rules for events [19].

For mining we currently use k-means, agglomerative clustering and basic event detection. SenseStream can export the rules discovered using these data mining algorithms in RuleML format. The RuleML Datalog format provides a simple and clean syntax for expressing "if–then" rules. Each condition is represented by one or

[24] http://linkeddata.org/.

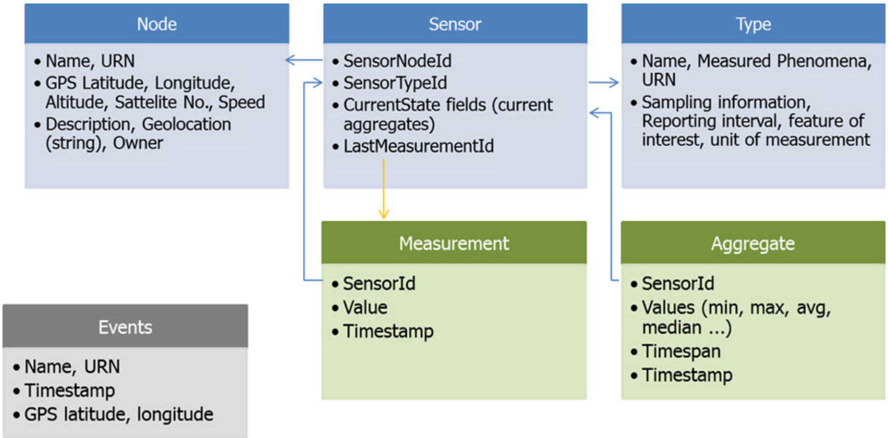



**Fig. 5.13** SenseStream Data Layer schema, the implementation has six stores, two for metadata, two for measurements and aggregates and one which essentially connects them

```
<And> <Atom>
  <op> <Rel iri="cyc:sensorObservation"/> </op>
  <Var> sensor </Var>
  <Ind iri="cyc:Raindrop"/> </Atom>
<Atom>
  <op> <Rel iri="cyc:doneBy"/> </op>
  <Var> sensor </Var>
  <Var> measurement </Var> </Atom>
<Atom>
```

**Fig. 5.14** RuleML sample from a rule related to raindrop measurement

more atomic formulas (“Atom”). For example the condition that raindrop exceeds 250 mm per day is represented in Fig. 5.14. The export in the RuleML format is dependent on the vocabulary used for the relation constants (“Rel”). Specialized domain ontologies can simplify the RuleML representation as they can have more specific relations and concepts.

#### 5.4.2.2 The Videk Mashup Server

The Videk mashup server acts as a glue between different components and services used. It interacts with sensors by receiving and parsing the data from them and then multiplexing it to the backup database, SenseStream, ResearchCyc and to the triple store. It exposes and extends the API of the SenseStream to the external applications and users and provides all the infrastructure to be used by the Videk GUI with the widgets that mash up the data and resulting knowledge described above.

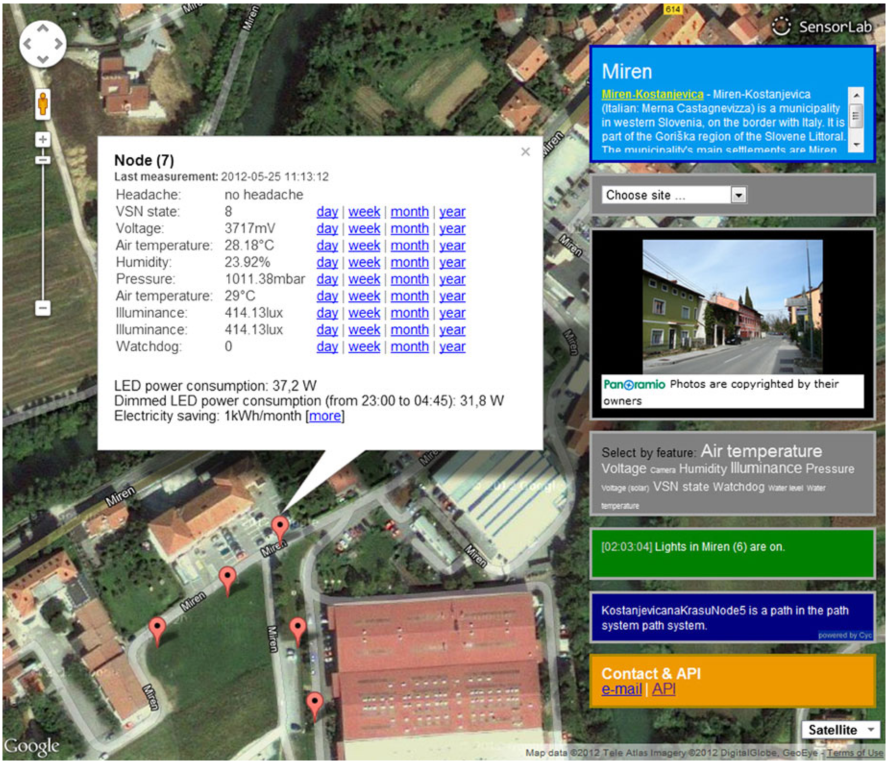



Fig. 5.15 The Videk mashup GUI

### 5.4.2.3 The Videk API and the GUI

The currently exposed API[25] allows retrieving sensor metadata such as lists of sensors on a node as well as last measurements from all devices. An example of corresponding calls is:

- http://sensors.ijs.si/xml/current-state
- http://sensors.ijs.si/xml/sensors-on-node/nodeid

For data manipulation, event handling and Ajax interactions the jQuery[26] library is used. The GUI[27] (see Fig. 5.15) receives data through the API layer, which is based on a PHP/MySQL custom made content management system (CMS). The GUI is based on Google Maps API and enables browsing and exploring the sensor nodes, looking at the last measurements data and visualizing measurement aggregates. It includes six widgets in the right sidebar: a location widget with some data

[25] http://sensors.ijs.si/sl/api/index.html.

[26] http://jquery.com.

[27] http://sensors.ijs.si.


**Fig. 5.16** Example knowledge about a sensor node from Cyc

about the place of the cluster of sensor nodes, a clustering widget which enables panning to a certain sensor cluster, a photo widget which shows some publicly available photographs taken within the map-view area, a selection widget which toggles visibility of the sensor nodes according to its measuring capabilities, a simple rule-based events widget and a Cyc widget, which shows natural language generated on sensor network metadata.

## *5.4.3 External Mashed-up Sources*

### 5.4.3.1 ResearchCyc

ResearchCyc[28] is the research release of Cyc, an artificial intelligence system which comprises a knowledge base and a reasoning engine. The idea behind it is to encode human knowledge in a structured way and reason about it similar to the way the human mind does. For instance, each of us learns the concept of a tree, branch, leaf and fruit. Then, we learn relationships between these concepts: that a tree has branches, on a branch grow leaves and fruits. Finally, we are able to recognize instances of these: this apple tree, this apple, this apple tree leaf. This knowledge builds up in our brains over the years and makes it possible to understand, communicate and reason.

In the process of building up such a knowledge we use our senses: sight, hearing, smell, taste and touch. Similarly, the purpose of using this technology in a mashup

[28] http://research.cyc.com/.

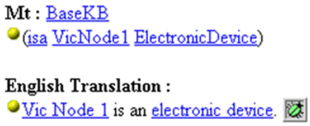



**Fig. 5.17** Example natural language transliteration of a logical statement

is to allow sensors—the equivalent of senses—to populate the knowledge base—the equivalent of part of our brain—with new knowledge. Finally, as humans communicate verbally, the Cyc system implements transliteration technology allowing the generation of natural language sentences based on the inserted and derived (via logical mechanisms) knowledge. In the current implementation only sensor metadata are sent to ResearchCyc. The concepts already exist in the knowledge base and the metadata represent instances of these concepts. An instance (i.e. an individual in Cyc terminology as can be seen in Fig. 5.16) corresponding to each sensor node and sensor is created. Then the relationships between them together with other data are inserted in the knowledge base. For instance, VicNode1 in Fig. 5.16 is an electronic device which has six instances of sensors connected to it or on-board, for example scp1000VicNode1 and virtualsensor1VicNode1.

The sensor knowledge is structured in Research Cyc's knowledge base in the form of logical sentences on which inference can be performed. For instance, we only state in the knowledge base that VicNode1 is an ElectronicDevice and the reasoning engine then infers that it is also a Computer. All this knowledge (i.e. asserted and inferred statements) can then be transliterated to natural language as shown for VicNode1 in Fig. 5.17. The ResearchCyc widget in the mashup GUI displays these statements.

#### 5.4.3.2 External Web Services

The mashup uses external data sources which are available as RESTful web services. Based on the sensor nodes' GPS coordinates, Google Maps are used as the GUI's background to present the geographical context of the deployments' locations. Then, also based on the GPS coordinates, the Geonames service is used to retrieve the names of the places where the sensors are deployed. For the deployment which is zoomed in Fig. 5.15, the name of the neighborhood Miren is displayed together with the relevant information from Wikipedia. Finally, the Panoramio service is invoked to retrieve and render pictures of the surrounding area.

## 5.5 Summary

In this chapter we provided a brief architectural overview of technologies which can be used in WoT mashups with emphasis on artificial intelligence technologies such as conceptualization and stream processing. Then, we looked at data sources and existing WoT mashups. In the last part of the chapter we discussed the architecture and implementation of Videk, a prototype mashup for environmental intelligence.

With respect to the architectural overview, we explained the components that form the network of things and we looked at existing standards and technologies which can be used for the conceptualization of this domain. We continued by looking at stream processing techniques that are relevant for processing the data coming from the network of things. With respect to WoT mashups we discussed physical and non-physical data sources that can be used, emphasizing standard and non-standard-based data sources. We have presented existing WoT mashups and discussed the possible systems powering WoT mashups. Finally, we presented in detail the implementation of Videk which covers the architectural components described in the chapter.

**Acknowledgements** This work was partially supported by the Slovenian Research Agency through the program P2-0016 and project J2-4197, the competence center KC OPCOMM, and the ICT Programme of the EC under PASCAL2 (ICT-NoE-216886), ENVISION (ICT-2009-249120) and PlanetData (ICT-NoE-257641). The authors are also grateful to Envigence d.o.o. for the collaboration on Slovenian sensor deployment.

## References

1. Atzori L, Iera A, Morabito G (2010) The internet of things: a survey. Comput Netw 54(15):2787–2805
2. Boley H, Tabet S, Wagner G (2001) Design rationale of RuleML: a markup language for semantic web rules. In: Proceedings of the first semantic web working symposium, SWWS'01, Stanford University, July 30–August 1, 2001
3. Chandola V et al (2009) Anomaly detection: a survey. ACM Comput Surv 41(3):15. doi:10.1145/1541880.1541882
4. Compton M et al (2009) A survey of the semantic specification of sensors. In: Proceedings of the 2nd international workshop on semantic sensor networks, October 2009
5. Compton M et al (2012) The SSN ontology of the W3C semantic sensor network incubator group. J Web Semant. doi:10.1016/j.websem.2012.05.003
6. Cugola G, Margara A (2012) Processing flows of information: from data stream to complex event processing. ACM Comput Surv 44(3):15. doi:10.1145/2187671.2187677
7. De Ipina DL et al (2007) A context-aware mobile mashup for ubiquitous web. In: Proceedings of IET international conference on intelligent environments, University of Ulm, Ulm, Sept 24–25, 2007
8. Etzion O, Nibblet P (2011) Event processing in action. Manning Publications, Shelter Island
9. Falutsos C, Ranganathan M, Manolopoulos Y (1994) Fast subsequence matching in time-series databases. In: Proceedings of ACM SIGMOID, Minneapolis, May 25–27, pp 419–429
10. Fenn J, LeHong H (2011) Hype cycle for emerging technologies. Gartner Inc, ID number: G00215650
11. Fortuna B (2011) Semi-automatic ontology construction: doctoral dissertation (Polavtomatska gradnja ontologij: doktorska disertacija), Ljubljana, October 2011
12. Fortuna C, Grobelnik M (2011) The web of things. In: World Wide Web conference. http://sensorlab.ijs.si/files/publications/2011-03-29-Fortuna_Grobelnik_Web_of_Things_Tutorial_WWW.pdf. Accessed 30 May 2012
13. Gaber MM, Zaslavsky A, Kirshnaswamy S (2005) Mining data streams: a review. SIGMOD Rec 34:18–26
14. Gama J (2010) Knowledge discovery from data streams. CRC Press, Boca Raton
15. Guinard D (2011) A web of things application architecture—integrating the real-world into the web. PhD thesis No 19891, ETH Zurich, Zurich

16. Hanson JJ (2009) Mashups—strategies for modern enterprises. Addison-Wesley, Reading
17. Kaupinen T, Benedikt G, de Espindola MG (2012) Sharing and analyzing remote sensing observation data for linked science. In: ESWC 2012
18. Le Phuoc D, Hauswirth M (2009) Linked open data in sensor data mashups. In: Kerry Taylor DD (ed) Proceedings of the 2nd international workshop on semantic sensor networks (SSN09), in conjunction with ISWC 2009, vol 522. CEUR, CSIRO, Canberra, pp 1–16
19. Moraru A, Kenda K, Fortuna B, Bradesko L, Skrajnc M, Mladenic D, Fortuna C (2012) Supporting rule generation and validation on environmental data in EnStreaM. In: ESWC 2012
20. Nath S et al (2007) SensorMap for wide-area sensor webs. IEEE Comput 40(7):90–93. doi:10.1109/MC.2007.250
21. Reichardt M (2003) The sensor web's point of beginning. http://spectroscopyonline.findanalytichem.com/spectroscopy/article/articleDetail.jsp?id=52681&sk=&date=&pageID=2. Accessed 11 Jan 2011
22. Sanchez L et al (2011) SmartSantander: the meeting point between future internet research and experimentation and the smart cities. In: IST future networks & mobile summit, Warsaw
23. Sheth A, Henson C, Sahoo SS (2008) Semantic sensor web. IEEE Internet Comput 12(4):78–83
24. Siegele L (2010) It's a smart world. The Economist http://www.economist.com/node/17388368. Accessed 30 Sep 2010
25. Simonis I (2008) OGC® sensor web enablement architecture. Open Geospatial Consortium. http://portal.opengeospatial.org/files/?artifact_id=29405. Accessed 2 Mar 2012
26. Smolnikar M, Platise U (2010) Versatile sensor node—a platform for the sensor as a service concept. In: Proceedings of prosense WSN and SME workshop: wireless sensor networks (WSN) and small medium enterprises (SME), Ljubljana, May 20, 2010
27. Stonebroker M, Çetintemel U, Zdonik S (2006) The eight rules of real-time stream processing. SIGMOID Rec 34:42–47
28. Tamayo A et al (2011) Empirical study of sensor observation services server instances. In: Advancing geoinformation science for a changing world. Lecture notes in geoinformation and cartography, pp 185–210

# Chapter 6
# Mashups Using Mathematical Knowledge

## Why Formulas Are Different

**Christoph Lange and Michael Kohlhase**

**Abstract** Mashups offer new functionality by combining, aggregating and transforming resources and services available on the Web. This chapter deals with *mathematical* mashups and focuses on those that process mathematical *knowledge* rather than, e.g., huge amounts of numeric *data*, as it is the structure of knowledge that distinguishes mathematics from other application domains.

The resources that mathematical mashups process primarily include *formulas* and the services they offer involve *computation*. In knowledge-rich mashups, the formulas are not hard-coded into the implementation but represented as explicit data structures, and often also presented to the user. These structures are different from the (re)presentations mashups usually process. To allow for automated processing, formulas need to be represented neither as plain text nor as images, but in a *symbolic* way. The representation of choice is, for compatibility and scalability reasons discussed in this chapter, usually not JSON or RDF, but semantic XML *markup*. Besides tables or graphs, mathematical mashups may also require formulas to be presented to the end user. The highest degree of interaction with formulas is offered by MathML—in those browsers that fully support it.

After introducing typical education and engineering use cases that benefit from mathematical mashups, this chapter reviews the conceptual and technical foundations for representing and presenting mathematical formulas, discussing MathML as well as alternatives. We continue with a review of mathematical web services and collections of mathematical knowledge that provide suitable building blocks for mathematical mashups. We then present the Planetary system, a math-enabled social semantic web portal that provides an environment for executable papers, and the SAlly framework that mashes up user interfaces of software applications with mathematical web services. Both environments mash up assistive services by hooking them into document structures, which have been annotated with terms from

C. Lange (✉)
School of Computer Science, University of Birmingham, Birmingham, UK
e-mail: c.lange@cs.bham.ac.uk

M. Kohlhase
Computer Science, Jacobs University Bremen, Bremen, Germany
e-mail: m.kohlhase@jacobs-university.de

B. Endres-Niggemeyer (ed.), *Semantic Mashups*, 171
DOI 10.1007/978-3-642-36403-7_6, © Springer-Verlag Berlin Heidelberg 2013

a mathematical background ontology. We conclude with an outlook towards contributing collections of mathematical knowledge to the Web of Data, and outline how such linked open datasets can drive further mathematical mashups in the near future.

## 6.1 Mathematical Knowledge and Its Management

Mathematics is a ubiquitous foundation of science, technology, and engineering. These fields share mathematics as a common foundation and consequently use the same rigorous style of argumentation and the same symbolic formula language. The process of understanding results is similar, too. For example, a software engineer can hardly understand a piece of software from its source code and the brief embedded documentation alone—i.e. the counterpart to rigorous mathematical notation—but will usually have to consult external manuals—compare mathematical textbooks—and records of developers' communication about the code, such as discussions in mailing lists and bug trackers. The latter resemble transcribed dialogs about mathematical proofs (think "software features") and their refutations (think "bug reports"), as Imre Lakatos studied them [75].

While the work presented in this chapter is mainly motivated from a mathematical perspective, we point out connections to other STEM fields (science, technology, engineering, mathematics) wherever appropriate.

This chapter focuses on mashups that deal with mathematical *knowledge*. These are not the only mashups in the mathematical domain, but we argue that they are the most interesting ones, because the structure of mathematical knowledge—most prominently the structure of *formulas*—distinguishes them from other mashups, which we call "data mashups" for the purpose of distinction. The Developer Apps showcase of Data.gov, the open data site of the US government, shows a selection of typical data mashups [38]: DataMasher [33], for example, accesses datasets that contain various kinds of statistical figures per US state, such as population, crime rate, government spending figures, or unemployment; it allows to combine two such datasets, to choose an arithmetic operator (e.g. sum or division) and computes the result of applying this operator to the respective data points for each state (cf. Fig. 6.1). The result is displayed on a map (with different color shades per state) or as a table. Now, we are interested in mashups that handle potentially arbitrary mathematical functions, and that know the types of their data well enough to tell that it does not make sense to subtract the total tax revenue per capita from the percentage of population covered by health insurance.

Where do knowledge-rich mathematical mashups help? Recently, an "industrialization" of mathematical research has been observed, exhibiting patterns such as big teams of authors, instant communication, more fluid collaboration, de-centralized modes of publication and knowledge authentication, and the usage of big computer systems [17, 23]; Andrea Asperti et al. similarly argued that "mathematics is destined to assimilate some practices of software development" [9]. Nowadays, computers assist with all steps of "doing mathematics": numeric as well as symbolic

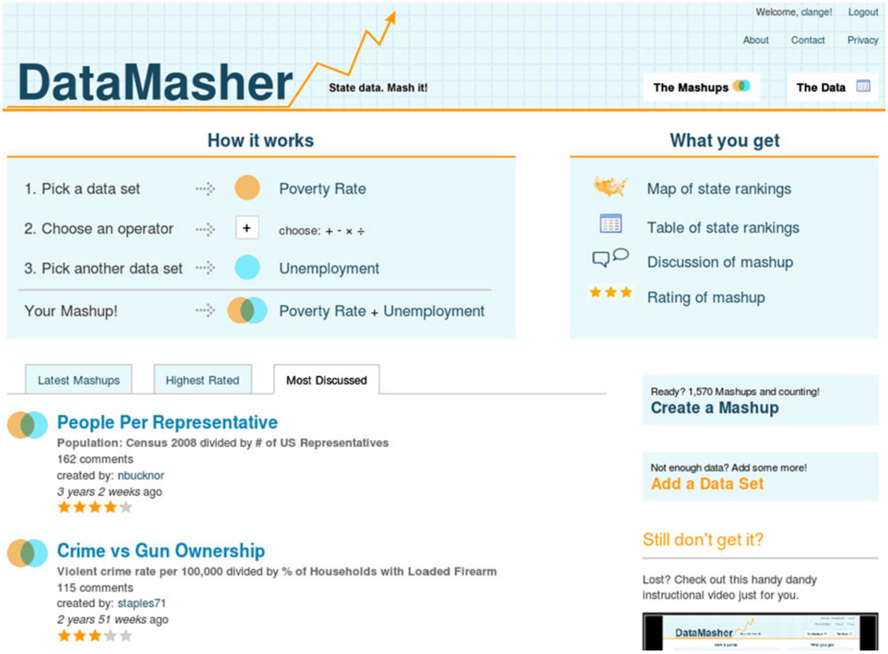



**Fig. 6.1** DataMasher [33], a data mashup

computation (the original purpose of *computers*!), verification and even generation of proofs [118] (these are subsumed under symbolic computation in the following), gaining intuition by experiment and generating counterexamples [51, p. 154], high-quality publishing (e.g. using the LaTeX typesetting system), and education (cf. e.g. [1, 26, 86, 99]). The proceedings of the previous conferences on intelligent computer mathematics (CICM) provide some further overview [11, 12, 22, 27, 34, 61].

Those aspects of computer mathematics that are not immediately concerned with numeric and symbolic computation are commonly referred to as *mathematical knowledge management* (MKM). The interdisciplinary MKM community consists of computer scientists, computer-savvy mathematicians, and digital library researchers, whose objective is "to develop new and better ways of managing mathematical knowledge using sophisticated software tools" [41],[1] or, more specifically, "to serve (i) mathematicians, scientists, and engineers who produce and use mathematical knowledge; (ii) educators and students who teach and learn mathematics; (iii) publishers who offer mathematical textbooks and disseminate new mathematical results; and (iv) librarians and mathematicians who catalog and

[1]This notion of the term "knowledge management" is wider than that of its traditional definition as "a range of practices used [. . . ] to identify, create, represent, distribute and enable adoption of insights and experiences. Such insights and experiences comprise knowledge, either embodied in individuals or embedded in organizational processes or practice" [66].

organize mathematical knowledge" [41].[2] Knowledge-rich mathematical mashups are needed in such application settings, but MKM research also provides the know-how to build them: As soon as a mathematical mashup performs computations beyond standard library functions, or accesses structured data beyond arrays of numbers, it needs an explicit representation of knowledge in order to know where to get its data (e.g. numbers) from, what these data mean, and how to publish the results to its end-users.

The rest of this chapter is structured as follows: Section 6.2 introduces the specific conditions of mathematical knowledge on the Web with a brief historical excursion. Section 6.3 provides an overview of existing mathematical mashups—or, rather, as very few of them exist to date, of *possible use cases* for mathematical mashups and of existing technology that helps building them. Section 6.4 introduces the Planetary system and the SAlly framework, two approaches to mashing up web or office documents with mathematical services. Section 6.5 concludes the chapter with an outlook towards a mathematical Web of Data.

## 6.2 Mathematical Knowledge on the Web

A lot of mathematical knowledge has been created and published on the Web,[3] both by practitioners doing mathematical research, education, or applications, and in research projects that investigated the applicability of web technology to mathematics. This section reviews the state of the art, focusing on systems and projects that could use mashup technology. We first review traditional web 1.0 applications, as they are still widely in use. Web 2.0 applications, enabling better communication and collaboration, are becoming more and more commonplace also among mathematicians, whereas semantic web technology is barely on the verge of achieving a breakthrough in the mathematical domain.

### *6.2.1 Web 1.0—Digital Libraries for Humans and Tools*

Virtually all mathematicians nowadays use digital libraries. For example, Zentralblatt MATH [144] and MathSciNet [4] are the largest services that provide reviews and abstracts of mathematical publications. The knowledge base is searchable online, by full text and by metadata, such as author, title, and the Mathematics Subject Classification (MSC [88]; see also Sect. 6.3.2.4).

Many of the libraries used by computer-based mathematics tools, such as computer algebra systems (CAS), proof assistants, and program verification systems,

[2] Numbering added by the authors.

[3] To reduce eye strain, we only capitalize this term, as well as the terms "Web 2.0" and "Semantic Web", when they denote the Web as a whole, but not when they are in an adjective position, as in "semantic web services".

have also been published on the Web—while still being edited and maintained off the Web. Consider, for example, the Journal of Formalized Mathematics, publishing machine-verified proofs from the Mizar Mathematical Library (MML [101]).

Finally, there are educational or general-purpose reference works, such as the Digital Library of Mathematical Functions (DLMF) and Wolfram MathWorld. The DLMF is a centrally edited reference of special functions [39]. MathWorld collects about 13,000 entries on mathematical topics, including downloadable files ("notebooks") for the Mathematica CAS [87].

**Easy Access, but Poor Collaboration and Retrieval** Besides facilitating *access* to mathematical knowledge, these sites (i) offer limited internal interaction and do not facilitate collaboration other than sharing links with collaborators, and (ii) the means of automatically retrieving, using, and adaptively presenting knowledge, e.g. in mashups, are restricted. Web 2.0 technology addresses problem (i), and semantic web technology addresses problem (ii). The following subsections review to what extent these developments have been adopted for mathematical applications.

### *6.2.2 How Working Mathematicians Have Embraced the Web 2.0*

Working mathematicians, both researchers and instructors, are increasingly using the Web 2.0 for collaboratively developing new ideas, but also as a new publication channel for established knowledge. Research blogs, wiki encyclopedias, and open educational repositories are typical representatives.

Blogging about *established* mathematical knowledge follows the traditional practice of publishing short reviews and abstracts of previously published material (cf. Sect. 6.2.1). Researchers have also found blogs useful to gather early feedback about preliminary findings. Successful collaborations among mathematicians have started in blogs and converged into conventional articles [13]. In the Polymath initiative, blogs are the exclusive communication medium for proving theorems in a massive collaborative effort [14, 114]. Compared to research blogs, the MathOverflow forum [94], where users can post their problems and solutions to others' problems, offers more instant help with smaller problems. Its reputation mechanism simulates the traditional scientific publication and peer review process in an agile way.

For ideas emerging from a blog discussion, or for creating permanent, short, interlinked descriptions of topics, wikis have been found more appropriate. The nLab wiki [106], a companion to the n-Category Café blog [105], is a prominent example. Where MathOverflow focuses on concrete problems and solutions, the Tricki [125] is a wiki repository of general mathematical techniques—reminiscent of a web 2.0 remake of George Pólya's classic "How to Solve It" [113].

Wikis that collect *existing* mathematical knowledge, for educational and general purposes, are more widely known. The PlanetMath encyclopedia [111] counts more than 8,000 entries at the time of this writing. The general-purpose Wikipedia with over 21 million articles in over 280 languages also covers mathematics [115]. Targeting a general audience, it omits most formal proofs but embeds the pure mathematical knowledge into a wider context, including, e.g., the history of mathematics, biographies of mathematicians, and information about application areas. The lack of proofs is partly compensated by linking to the technically similar ProofWiki [117] with over 2,500 proofs, or to PlanetMath.

Similarly, open educational resources (OER) about mathematics can be found on general-purpose as well as mathematics-specific sites, some of them driven by wikis. For example, the Connexions [28] open courseware repository, having the more traditional ownership model of a content management system. It promotes the contribution of small, reusable course modules—more than 20,000, about 4,000 from mathematics and statistics, and about 7,000 from science and technology—to its *content commons*, so that the original author, but also others can flexibly combine them into collections, such as the notes for a particular course. i2geo [58, 82] is an example of a wiki dedicated to educational interactive geometry constructions.

Finally, established mathematical knowledge bases are starting to employ web 2.0 front-ends to simplify and crowd-source maintenance—for example the recently developed prototypical wiki front-end for the Mizar Mathematical Library (MML) [2, 130] mentioned in Sect. 6.2.1.

**Little Reuse, Lack of Services** Web 2.0 sites facilitate collaboration but still require a massive investment of manpower for compiling a knowledge collection. Machine-supported intelligent knowledge reuse, e.g. from other knowledge collections on the Web, does not take place. Different knowledge bases are technically separated from each other by using document formats that are merely suitable for knowledge presentation but not for *re*presentation, such as XHTML with LaTeX formulas. The only way of referring to other knowledge bases is by untyped hyperlinks. The proof techniques collected in the Tricki cannot be automatically applied to a problem developed in a research blog, as neither of them is sufficiently formalized. Conversely, the Polymath community does not have any automated verification tools at hand.[4]

Intelligent information retrieval, a prerequisite for finding knowledge to reuse and to apply, is restricted to regular text search even though crucial parts of mathematical knowledge are only conveyed in formulas.

Finally, the integration of mathematical web 2.0 sites with automated reasoning and computation services is scarce. Interactive computation is available in mathematical eLearning systems, such as ActiveMath [1] or MathDox [86]—where document authors have sufficiently formalized the underlying mathematics in separate editing tools before publishing—but less so in general-purpose digital libraries and collaboration environments. Mashups, which have otherwise been a driving force of Web 2.0 development, scarcely exist for mathematical tasks, as detailed in Sect. 6.3.

[4] Other than the above-mentioned wiki front-end to the Mizar Mathematical Library and a similar one for the library of the Coq theorem prover, none of which are the *primary* entry points to these libraries yet, we are not aware of mathematical web 2.0 sites that integrate formal verification.

### 6.2.3 Adoption of the Semantic Web in MKM

In the early 2000s, when XML was increasingly used for mathematics, particularly for formulas (cf. Sect. 6.3.2), the first building blocks of the Semantic Web vision approached standardization. This sparked interest in the emerging MKM community (cf. Sect. 6.1), whose members hoped that semantic web technology would help to address their challenges. This seemed technically feasible, particularly as both communities made use of XML as a serialization format and URIs for identifying things [84]. The two main lines of applying semantic web technology to MKM focused on *digital libraries*—improving information retrieval and giving readers access to automated reasoning and computation services—and *web services*—providing self-describing interfaces to automated reasoning and computation on the Web, so that they could solve problems sent to them by humans or other agents. We refer the reader to [78] for a survey of these (largely frustrated) early attempts of using Semantic Web techniques for MKM. Important systems of this time are the "Hypertextual Electronic Library of Mathematics" HELM [7, 52], the MathServe distribution architecture for automated reasoning [147], and the MONET project, which pioneered an architecture for mathematical semantic web services [20, 102]. The problems encountered by these systems can be attributed to (i) semantic web technologies were adopted before they were mature (which benefited the Semantic Web, but not MKM), and (ii) a fundamental mismatch between RDF-based technologies and the requirements for dealing with mathematical formulas.

The combination of the Web 2.0 and the Semantic Web, sometimes called "Web 3.0", started to emerge in 2006 (cf. [5]) and is now conquering mainstream applications via incremental enhancement of successful Web 2.0 applications with semantic technology, as can be seen from general-purpose social semantic web engines such as Semantic MediaWiki [122] or Drupal 7 [31]. In 2007, Stefano Zacchiroli suggested jumping on this train as a way of retrying the application of semantic web technology to mathematics after the initial failure: Web 2.0 technology would allow for interactively editing mathematical content in the web browser, and Social Web initiatives such as PlanetMath had already proven that there is "a community of people interested in collaboratively authoring rigorous mathematics on the web" [142].[5] Moreover, both HELM and MONET would now benefit from the wide support for SPARQL, the standard query language for RDF.

We are now witnessing a resurgence of mathematical applications on the Web 3.0. The emerging HTML5 [54] includes MathML without requiring the strict XML conformance that authors and UI widget toolkits often fail to achieve; soon, more browsers can be expected to support MathML. PlanetMath is being ported to the Drupal-based Planetary engine (cf. Sect. 6.4.2) and is currently being deployed for beta testing [73, 112]. The Mathematics Subject Classification scheme (MSC),

[5] Similarly, Baez suggests that the release of a TEX formula editor plugin for the popular WordPress blog engine was a major incentive for mathematicians to start blogging [13].

widely used in paper-based and digital libraries, has recently been officially published as a linked open dataset (cf. Sect. 6.3.3.5), which enables easier access for web services, as well as an easier construction of links between mathematics and related fields. We are even seeing first commercial systems like "truenumbers" [126] a system for supporting the representation, management and copy/pasting of engineering values as semantically enhanced data.

## 6.3 Mathematical Knowledge Mashups

Few mathematical knowledge mashups or mashup-based mathematical knowledge management systems exist to date. The ProgrammableWeb [116] mashup directory lists three mashups and three APIs tagged with "math", out of more than 6,000 mashups and more than 5,000 APIs overall—an observation that suggests that the mashups we classified as "data mashups" in Sect. 6.1 are not commonly considered proper *mathematical* mashups. This can, however, be expected to change soon: In 2010, Wolfram, who had already released the Wolfram Alpha "computational knowledge engine" (cf. Sect. 6.3.3.1), released an on-line development environment for building custom "widgets" as front-ends to Wolfram Alpha [137]. These widgets are mini-applications with a custom user interface. They perform simple computations backed by Wolfram Alpha and can be embedded into web pages. So far, users have developed several thousand widgets. However, these widgets only give a foretaste of the potential of mathematical mashups: (i) Most of them are rather trivial front-ends to Wolfram Alpha, merely offering input forms for information that one would otherwise include in a query to Wolfram Alpha, such as an amount of money, a source and a target currency, (ii) they are limited to acting as front-ends to Wolfram Alpha, and (iii) they can only be created within Wolfram's development environment.

Thus, as very few actual mathematical mashups exist to date, this section focuses on *possible* use cases for them, and on existing technology that helps building them.

### *6.3.1 Characterization and Purpose*

The general *purpose* of mathematical mashups (including data mashups) can be subclassified into providing a user-friendly front-end to *computation* tasks (such as computing the derivative of a given function and plotting its graph), or retrieving and aggregating mathematics-related *information* (such as finding existing theorems applicable to a given formula). Both can occur in combination; Wolfram Alpha, for example, first retrieves functions related to the user's query and then evaluates them for some common values. Mathematical mashups can be further classified by their input and output interfaces. Possibilities for providing *input* include *entering* a *mathematical expression* (visually such as $\frac{4\pi r^3}{3}$ or in a linear text form such as

`(4*PI*r^3)/3`) or a *sequence of data points* (e.g. to apply a statistical function to). Instead of manually entering such information, one may also *select* existing information—e.g. an existing expression in a document, or existing data points in a table or chart. Mashups may provide their *output* as a *mathematical expression* (e.g. an expression representing the derivative of the input expression, or a plain number representing the result of a numeric computation) or as a *chart or graph* (e.g. for visualizing large amounts of numbers, or functions, or geometric shapes having a given property). If the input was provided by selecting existing information, a mashup may also provide its output by *rewriting or redrawing*, e.g., the selected expression or graph.

### *6.3.2 Technical Background*

Data mashups that compute a straightforward result from some given numbers do not require any mathematics-*specific* mashup technology. This section focuses on handling *mathematical formulas*, a feature unique to mathematical web applications.

The symbolic language of mathematical formulas is non-trivial for its extensibility; particularly in mathematical research it is common practice to define new concepts and introduce new notation for them. Mathematics is not the only domain that employs its own symbolic notation: Formulas are also used in chemistry to express the structure of atoms forming a compound. An interesting aspect of musical notation is its multimodal combination with text in vocals. The notation of art music has been largely standardized until the 20th century[6] and then diversified. However, we argue that mathematics is distinct in that new notation can be introduced within the language of mathematics itself: Besides symbolic language, mathematics employs a conventionalized natural language (also called "mathematical vernacular"). In this language, the notation of a symbol is usually introduced with its first declaration, typical phrases being "We will denote by $Z$ the set ...", "The notation $aRb$ means that ...", etc. [127].

On the user interface, formulas are often entered and rendered in textbook style; the data formats for this are called *presentation markup* (Sect. 6.3.2.1). For exchanging formulas with tools that perform computation or reasoning, they must not be encoded by their layout but by their meaning; this is called *content markup* (Sect. 6.3.2.2). MathML, the most common language for both presentation and content markup of mathematical formulas, also allows for interacting with mathematical formulas in the browser—an important feature of the user interface of a mathematical knowledge mashup (Sect. 6.3.2.3). Finally, mathematical mashups may draw on mathematical *data*, such as number series or mathematical background knowledge such as definitions or theorems. Section 6.3.2.4 explains how to publish such data on the Web.

[6] But see Donald Byrd's documentation of "extremes of conventional music notation" [19].

```
<math>                                        <!-- Presentation MathML -->
  <msub>                                                <!-- subscript -->
    <mi>a</mi>                                         <!-- identifier -->
    <mn>1</mn>                                             <!-- number -->
  </msub>
  <mo>+</mo>                                             <!-- operator -->
  <mfrac>                                                <!-- fraction -->
    <mn>1</mn>
    <mn>2</mn>
  </mfrac>
</math>
```

**Listing 6.1** The formula $a_1 + \frac{1}{2}$ in Presentation MathML (namespace declarations omitted)

#### 6.3.2.1 Mathematical Formulas on the User Interface: Presentation Markup

Possibilities to present mathematical formulas in a human-readable textbook style include plain text, images, and MathML.

Plain text rendering is sufficient for simple mathematical formulas. The following Unicode blocks are of particular relevance [129]: Mathematical Operators, Miscellaneous Mathematical Symbols-A, Miscellaneous Mathematical Symbols-B, Supplemental Mathematical Operators, Letterlike Symbols, Miscellaneous Technical, Arrows, Miscellaneous Symbols and Arrows, and Mathematical Alphanumeric Symbols.

More complex mathematical formulas cannot be rendered within a line of text as they have a two-dimensional layout; consider operators embellished with sub/superscripts, expressions in sub/superscripts, fractions, or matrices. On the Web, they have traditionally been published as images (generated, e.g., with the LaTeX typesetting system). Images are, however, limited w.r.t. rendering quality, accessibility, reuse (e.g. copy/paste), and interaction possibilities.

Therefore, the preferred method for rendering complex mathematical formulas is the XML-based MathML (Mathematical Markup Language [10]) format. It was originally conceived for embedding mathematical objects into (X)HTML and is now part of HTML5. Besides a presentation-oriented sublanguage (unofficially called "Presentation MathML"), it features a content-oriented one (Content MathML). Listing 6.1 shows a formula in Presentation MathML.

As of 2012, Presentation MathML is natively supported by Mozilla's Gecko browser rendering engine used by Firefox. Safari supports it from version 5.1; Chrome, another browser using the WebKit engine, supports it from version 24. Opera has limited MathML support in that it treats MathML like generic XML but applies a custom built-in CSS stylesheet to it. Internet Explorer supports MathML via the MathPlayer plugin, which offers additional accessibility support by speech output and magnification [95].[7] Independently, any browser with CSS and

[7]The information on browser support has been taken from Wikipedia [91], "When can I use…" [134], and a review by Timothy Vismor [131].

```
<math>                               <!-- non-strict Content MathML -->
  <apply>      <!-- application of an operator to arguments -->
    <plus/>           <!-- short name for a common operator -->
    <!-- mixed presentation and content markup -->
    <ci>                                      <!-- identifier -->
      <msub><mi>a</mi><mn>1</mn></msub></ci>
    <!-- built-in constructor for a common type -->
    <cn type="rational">1<sep/>2</cn>  <!-- rational number -->
  </apply>
</math>
```

**Listing 6.2** The formula $a_1 + \frac{1}{2}$ in non-strict Content MathML (namespace declarations omitted)

JavaScript support can display MathML via MathJax [90], which replaces MathML subtrees in the DOM with CSS layouts and special fonts at display time.

#### 6.3.2.2 Mathematical Formulas in Computation: Content Markup

In applications that go beyond high school mathematics, presentation markup leaves too many ambiguities to be useful as an input format for numeric or symbolic computation. This problem is addressed by content markup, which represents formulas (then usually called "mathematical objects") by their functional/operator structure, similar to an abstract syntax tree.

As an example for the ambiguity of presentation markup, consider three possible meanings of the formula[8] $O(n^2 + 1)$:

Landau symbol (a.k.a. big-$O$ notation): the set of all functions that asymptotically grow at most as fast as $n^2$, where the $+1$ is actually superfluous.

Function application: the application of *some* function named $O$—which needs not be the Landau set constructor function—to $n^2 + 1$.

Invisible times: $O$ (e.g. some variable) multiplied with $n^2 + 1$, where the multiplication operator is invisible.[9]

Important building blocks of mathematical objects as supported by Content MathML [10, Chap. 4] are numbers, variables, symbols (operators, functions, sets, constants), and applications of mathematical objects to other mathematical objects. Content MathML comes with a default supply of symbols that cover high school and introductory university education. The non-strict sublanguage of Content MathML offers pragmatic shorthands for referring to these symbols (Listing 6.2), whereas

[8] If we do not have the information that this is a *single* formula, then additional readings appear.

[9] Even in Presentation MathML, it is, however, best practice to mark up the distinction between multiplication and function application explicit using the special Unicode characters *Function Application* (U + 2061) and *Invisible Times* (U + 2062). Both characters occupy no space on the screen and are thus invisible.

```
<math>                              <!-- strict Content MathML -->
  <apply>
     <!-- a symbol referenced by CD and name; the actual URI of
      this symbol is http://www.openmath.org/cd/arith1#plus -->
    <csymbol cd="arith1">plus</csymbol>
    <semantics>
      <ci>a1</ci>
      <!-- annotation ("parallel markup") -->
      <annotation-xml
       encoding="application/mathml-presentation+xml">
        <msub><mi>a</mi><mn>1</mn></msub>
      </annotation-xml>
    </semantics>
    <apply>            <!-- constructor for rational numbers -->
      <csymbol cd="nums1">rational</csymbol>
      <cn type="integer">1</cn>
      <cn type="integer">2</cn>
    </apply>
  </apply>
</math>
```

**Listing 6.3** The formula $a_1 + \frac{1}{2}$ in strict Content MathML (namespace declarations omitted)

the strict sublanguage references all symbols by URI (Listing 6.3). Their semantics is described in external vocabularies called *Content Dictionaries* (CDs); authors can create and use additional CDs as needed. MathML delegates the task of writing CDs to other languages, such as OpenMath [18].

OpenMath has originally been invented to facilitate data exchange between computer algebra systems (CAS). It comprises a sublanguage for mathematical objects and a language for CDs. This chapter does not cover OpenMath's object language, for three reasons: (i) HTML5 only supports MathML, but not arbitrary other XML namespaces (such as OpenMath's), (ii) OpenMath is isomorphic to strict Content MathML, and (iii) XSLT stylesheets for a bidirectional conversion exist [109]. OpenMath CDs usually do not fully specify the semantics of mathematical operators; instead, developers of CAS and other systems for numeric and symbolic computation are supposed to use the CDs as specification manuals when implementing *phrasebooks*, which translate OpenMath objects into the native languages of such systems.

Several CAS support a subset of the CDs built into Content MathML. Note, however, that (i) developers of mathematical software do not always use the OpenMath terminology of "CDs" and "phrasebooks" (instead, they may simply refer to the ability to import and export OpenMath or Content MathML), and that (ii) few systems support OpenMath or Content MathML as a part of their core functionality. In many cases, plugins particularly for OpenMath have been developed in academic research projects and are no longer maintained now. Systems offering varying degrees of OpenMath or Content MathML support include (in alphabetical order) GAP [32], Mathematica [139], MuPAD (now known as the Symbolic Toolbox of Matlab) [55], and Yacas [141].

Content MathML is encoded as XML; OpenMath additionally specifies a binary encoding. No JSON encoding has been developed so far, but with Popcorn [56, 120] there is a de facto standard text encoding for OpenMath objects (Listing 6.4). RDF encodings for Content MathML have been suggested, but none has been implemented so far [78, Sect. 4.3.2] as they are considered space-inefficient.

```
a1{                                                      /* Popcorn */
  mathmlkeys.alternate-representation
  -> 'application/mathml-presentation+xml<msub>
       <mi>a</mi><mn>1</mn></msub>'
} + 1 // 2
```

**Listing 6.4** The formula $a_1 + \frac{1}{2}$ in Popcorn (namespace declarations omitted)

#### 6.3.2.3 Prerequisites for Interacting with Formulas in the Browser

There are two important prerequisites for interacting with mathematical formulas in the browser: (i) their presentation markup should carry semantic annotations, which the client side of a mashup can use to interact with a server-side web service, and (ii) the browser must be able to redisplay them, completely or partly, according to the action the user performed.

*Parallel markup* is the most comprehensive solution satisfying requirement (i): MathML allows to combine the presentation and content markup of a (sub-)formula in one expression tree and identify corresponding sub-formulas by cross-references (see Fig. 6.2). This correspondence supports interactions with computational services where the user must identify sub-expressions of formulas. For instance, if we want to export a sub-expression for evaluation in a CAS, then we could make use of the parallel markup in Fig. 6.2, there the light gray range is the user's selection, with the start and end node in bold face. As the CAS only accepts well-formed content expressions, we first look up their closest common ancestor that points to content markup via the *xref* attribute—here: *E.2*. Now, we can pass the target content markup tree to the CAS.

HELM [7, 52] was an early system that satisfied the interaction requirements listed under (ii) above. In HELM, one could perform actions on MathML formulas, e.g. simplifying a selected (sub)-expression using an automated reasoning backend attached to the library. In modern MathML-aware browsers, the *maction* element [10, Chap. 3.7.1] serves as a generic container for one or more Presentation MathML expressions, with which the user can interact. The *@actiontype* attribute allows for defining the type of interaction. The MathML recommendation suggests some (cycling through multiple alternative children, displaying a tooltip, requesting user input that can replace the current expression, etc.), but generally the interpretation *@actiontype* is up to applications. MathPlayer [95] supports those suggested by the MathML recommendation. No other browser has built-in support for specific

```
<semantics>
 <!-- a+ b 2 c -->
 <mrow xref="#E">
  <mi xref="#E.1">a</mi>
  <mo xref="#E.0">+</mo>
  <mrow xref="#E.2">
   <msup xref="#E.2.1">
    <mi xref="#E.2.1.1">b</mi>
    <mn xref="#E.2.1.2">2</mn>
   </msup>
   <mo xref="#E.2.0">&#x2062;
    <!-- INVISIBLE TIMES -->
   </mo>
   <mi xref="#E.2.2">c</mi>
  </mrow>
  <mo xref="#E.0">+</mo>
  <mi xref="#E.3">d</mi>
 </mrow>
```

```
  <annotation-xml>
   <apply id="E">
    <csymbol cd="arith1" id="E.0">
     plus</csymbol>
    <ci id="E.1">a</ci>
    <apply id="E.2">
     <csymbol cd="arith1" id="E.2.0">
      times</csymbol>
     <apply id="E.2.1">
      <csymbol cd="arith1" id=
       "E.2.1.0">power</csymbol>
      <ci id="E.2.1.1">b</ci>
      <cn type="integer"
       id="E.2.1.2">2</cn>
     </apply>
     <ci id="E.2.2">c</ci>
    </apply>
    <ci id="E.3">d</ci>
   </apply>
  </annotation-xml>
</semantics>
```

**Fig. 6.2** Parallel markup with Presentation MathML elements (*left column*) pointing to Content MathML elements (*right column*)

action types, but the Gecko rendering engine allows for choosing the expression to be displayed from multiple alternative child expressions by changing the value of the @*selection* integer attribute (which is possible from JavaScript via the DOM).

#### 6.3.2.4 Publishing Mathematical Data for Mashups

Mathematical knowledge is more than just formulas. Formulas occur in mathematical statements such as definitions, axioms, theorems, proofs, and examples. Like entries of CDs (cf. Sect. 6.3.2.2), definitions and axioms may fix the semantics of new mathematical symbols; theorems assert additional properties of symbols, proofs validate theorems, and examples demonstrate the usage of symbols, definitions, axioms or theorems in practical settings, e.g. for symbolically simplifying an equation that describes the behavior of a technical system, or for numerically approximating a solution of such an equation. Publishing such knowledge in a suitable way for mashups requires implementing it in a data format that mashups can process, and making the data accessible on the Web. This section briefly reviews existing possibilities, whereas Sect. 6.3.3.5 points to concrete datasets published on the Web.

The traditional format of choice for exchanging mathematical knowledge beyond formulas has been XML, RDF encodings have appeared more recently, whereas JSON is practically unknown. XML is the primary way of encoding OpenMath CDs, which have been introduced in Sect. 6.3.2.2. OMDoc is an XML language that extends CDs by textbook-style statements and a notion of modular theories [67, 107]. Both the OpenMath CD language and OMDoc have unofficial RDF encodings [78, Sect. 4.3.2].

Publishing knowledge as linked data [50] allows mashups to access it with little effort: They can download information about any resource (e.g. a mathematical theorem) by dereferencing its identifier (i.e. treating its URI as a URL), and these information records usually provide links to further relevant resources (e.g. examples in which the theorem is applied). Most commonly, such information is provided as RDF, but actually the client (here: the mashup) indicates its preferred format in the HTTP request header. The dominance of XML for mathematical formulas and the existence of both XML and RDF encodings for knowledge beyond formulas suggest a dual XML and RDF publishing approach; one way to achieve this is to maintain the knowledge primarily in XML and translate it to RDF (cf. [78, Sect. 5.1]). A major complication to publishing mathematical knowledge that declares custom symbols is the OpenMath scheme of identifying symbols with URIs of the form *base-URI/cd#symbol* (consider, e.g., http://www.openmath.org/cd/arith1#plus used in Listing 6.3), which Content MathML also relies on. The need to use "hash" URIs limits the possibility of publishing and may impair performance in the case of CDs that declare many symbols; see [78, Sect. 5.2] for a detailed discussion of this and other related problems.

Finally note that the openness of the RDF data model allows for combining purely mathematical knowledge with related non-mathematical knowledge, such as knowledge about real-world application scenarios. Or, conversely, seen from the point of such application scenarios, linked data mechanisms allow for enriching existing datasets with mathematical semantics, as we have previously shown for governmental statistics datasets, which we enriched with pointers to OpenMath definitions of mathematical operators used to derive values from original data [132].

### *6.3.3 Tools*

Tools useful for mathematical mashups can roughly be classified into: services performing computation, including verification (Sect. 6.3.3.1), services for publishing mathematical content in web-compliant formats (Sect. 6.3.3.2), user interface components for input (Sect. 6.3.3.3) and output (Sect. 6.3.3.4), and useful data published for reuse (Sect. 6.3.3.5).

We mainly focus on tools supporting content markup, as this is the most versatile format for communicating with computational tools. Beyond the tools reviewed here, mathematical mashups may employ further components, such as user models for recording knowledge items read or exercises mastered by the user (e.g. in the ActiveMath eLearning system [1, 97]), course generators that arrange learning objects into a sequence (e.g. PAIGOS [128], used in ActiveMath), or facilities that enable communication between users or between users and instructors. However, besides the possibility to *parameterize* them with a mathematical domain ontology (cf. PAIGOS [128] or the discussion facility of the SWiM wiki [76, Chap. 6.6]), such tools work independently from a particular application domain.

#### 6.3.3.1 Services for Computation

Tools that perform numeric and symbolic computation, including formal verification, have existed as stand-alone command-line or desktop applications for a long time. Few of them have a web service interface, and hardly any such tool is freely accessible on the Web.

Wolfram Alpha [136], based on the webMathematica web front-end to the Mathematica CAS (which itself has a natural language input interface), offers publicly accessible computation facilities. Wolfram Alpha is primarily designed for interactive use via its web interface: users enter a mathematical expression (or a natural language phrase), and Wolfram Alpha tries to return everything that it can find out about this expression and that it deems relevant, such as a factorization, its roots, or a plot (cf. Fig. 6.3). Additionally, there is an API for non-interactive use, which allows for requesting output in different formats, but the Mathematica syntax is the only content markup supported; for details and possible workarounds see [35]. Using the API requires, depending on the feature, free registration or buying a license.

Hiroshi Nakano et al. have developed a non-public mashup that integrates formulas resulting from symbolic computation, as well as plots of functions, into web

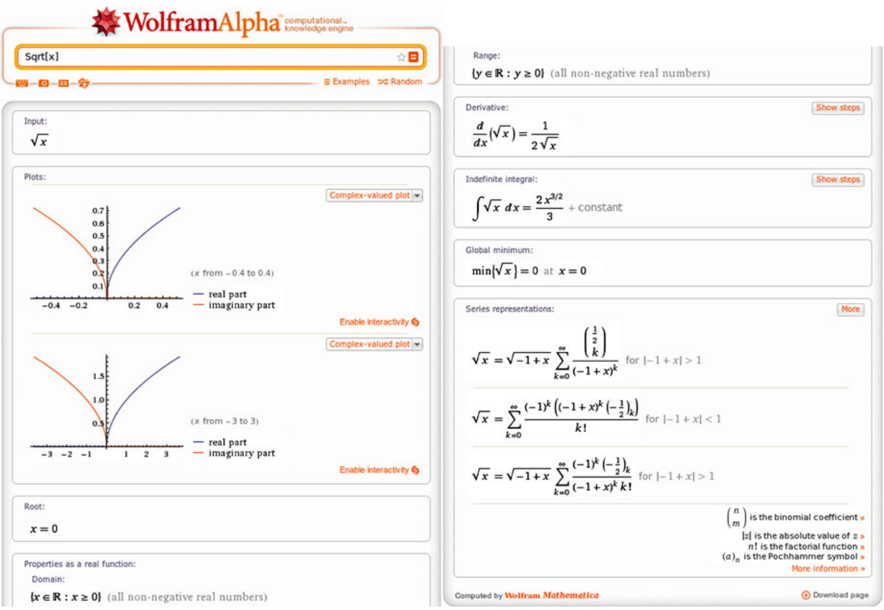



**Fig. 6.3** Wolfram Alpha's results for `Sqrt[x]`

pages [104]. On the server side, they obtain Presentation MathML formulas or gnuplot data from the Maxima CAS and wrap it into JSON-P.

Web service interfaces for exchanging content markup with computational software exist, e.g. as front-ends for CAS that support OpenMath (cf. Sect. 6.3.2.2), but have not generally been made available for public use, e.g. by mashups. For developing a mathematical mashup, one would have to install one's own instance of any such system.

The MONET project pioneered an OpenMath-aware semantic web service architecture [20, 102]; some MONET services are still used internally in the MathDox eLearning system [26, 86] in a mashup-like architecture. The SCIEnce project (Symbolic Computation Infrastructure for Europe [119]), a more recent driving force of research on symbolic computation web services, developed SCSCP (Symbolic Computation Software Composability Protocol [49]), a lightweight XML remote procedure call protocol, which transfers OpenMath objects and whose communication semantics heavily relies on a custom OpenMath vocabulary. SCSCP front-ends have been developed for a number of CAS [119]. We are not aware of JavaScript clients libraries for MONET or SCSCP, which would facilitate the integration of such services into mashups.

#### 6.3.3.2 Services for Publishing

Basic possibilities for publishing mathematical formulas on Web pages have been mentioned in Sect. 6.3.2.1. This section focuses on Presentation MathML, the most advanced solution for this, and on how to obtain it from content markup, as, e.g., returned by a computational service. Conceived as a function $\rho : \text{content} \to \text{presentation}$, this translation is usually achieved by recursing over the abstract syntax tree represented by the content markup and applying different rules for different mathematical symbols, so that, e.g. $\rho(\text{plus}(a, b))$ would result in $\rho(a) + \rho(b)$. An individual per-symbol or per-symbol-pattern rule is also called a *notation definition* for this symbol.

Traditionally, the content $\to$ presentation translation has been studied on the XML level and implemented as XSLT stylesheets. The probably largest collections of notations natively defined in XSLT translates Content MathML to Presentation MathML and accompanies the MathML 3 specification as non-normative "example XSLT code" [133], and a related collection that defines 143 notations for the symbols of the official OpenMath 2 CDs [108].

To see why these are cautiously declared "examples", realize that mathematical notation depends on multiple dimensions of context. Consider, for example, the French/Russian notation of the binomial coefficient $C_n^k$ vs. the German/English notation $\binom{n}{k}$, and see [99, 103] for systematic surveys.

XSLT as a low-level XML $\to$ XML translation language has been found inadequate for modeling notation definitions with contextual information. The rendering component of the ActiveMath eLearning system and the JOMDoc library developed in our research group offer high-level XML languages for defining context-sensitive notations. ActiveMath generates low-level XSLT stylesheets from the XML notation definitions [83], whereas JOMDoc implements a rendering algorithm internally [63, 71]. JOMDoc is particularly notable for producing cross-linked parallel markup of the presentation markup annotated with the original content markup.

Formula rendering engines are typically not ready for use in mashups; one would have to install them on the server and wrap them into a web service interface. Both ActiveMath [1] and TNTBase [146]—a document database that integrates JOMDoc, albeit without the presentation context features—provide such HTTP interfaces (cf. [145] for TNTBase). One can store content markup documents in the database and have them published as presentation markup, or POST an arbitrary content markup expression to the server and have it rendered to presentation markup on the fly (ActiveMath only).

As an alternative, Mozilla's Gecko browser rendering engine and Internet Explorer are capable of processing XSLT on the client, albeit with different APIs and limited to version 1.0. The above-mentioned MathML and OpenMath XSLT stylesheets are implemented in XSLT 1.0. The Sentido formula editor mentioned in the following section also comes bundled with client-side XSLT stylesheets for rendering a preview of the formula being edited.

#### 6.3.3.3 User Interface Input Components

Input components for mathematical formulas can be subdivided as follows: linear text input vs. visual composition (also referred to as direct manipulation) of a formula, presentation markup vs. content markup generation, and client-side vs. server-side production of the resulting markup. Separately, this section addresses interaction with existing formulas. Table 6.1 provides an overview of existing components.

A large number of visual formula editors, offering visual selection of mathematical symbols as well as visual cursor navigation, supports Presentation MathML [93]. Fewer visual editors exist for content markup. Visual editors for OpenMath objects—with a restricted set of supported CDs—have been developed by WIRIS [85, 135] and for the MathDox eLearning system [86]. The Connexions MathML editor supports the built-in symbol vocabulary of Content MathML 2, including embedded Presentation MathML [29].

As a means of facilitating the implementation of content markup input interfaces, particularly when the supply of symbols/CDs is unlimited, one-dimensional ("linear") text input syntaxes have been developed. The Popcorn text encoding for OpenMath objects [56, 120] has been introduced in Sect. 6.3.2.2. For a restricted set of symbols from the official OpenMath/MathML CDs, Popcorn supports intuitive infix notations such as $a + b$ instead of `arith1.plus(a,b)`. QMath [47] has originally been created as an extensible linear input syntax for OpenMath objects, with built-in support for the official CDs but the possibility to support arbitrary other CDs as well. In addition to QMath's own syntax, the QMath processor supports alternative syntaxes resembling the syntaxes of various CAS (Maxima, Yacas,

**Table 6.1** Classification of formula input tools

| Name | Input | Output | Site of execution | Ref. |
|---|---|---|---|---|
| WIRIS OpenMath input editor | fully visual[a] | OpenMath | client (Java applet) | [85, 135] |
| MathDox formula editor | fully visual | OpenMath | client (HTML canvas) | [86] |
| Connexions MathML editor | fully visual | Content MathML | client (Presentation MathML) | [29] |
| Popcorn | linear | OpenMath | server (no interface provided) | [56, 120] |
| QMath | linear | OpenMath | client | [47] |
| Sentido | linear + visual[b] | OpenMath | client (Presentation MathML) | [48] |
| LaTeXML daemon | linear (LaTeX) | Presentation MathML | server (HTTP POST) | [44, 46] |

[a] Visual symbol selection plus visual navigation

[b] Just visual symbol selection

Mathematica, Maple). The Sentido formula editor [48] combines linear input using QMath with a tool palette for visual selection of symbols.

The preferred linear input format for presentation-oriented formulas is LaTeX, as most users in academic mathematics and science are familiar with it. There is a large number of converters from LaTeX to Presentation MathML [92]. LaTeXML [100] in particular has a high output quality, as it reimplements the original TeX parser. The LaTeXML daemon [46] wraps LaTeXML into an HTTP POST interface publicly accessible at [44]. Content representations can be extracted from restricted subsets of LaTeX, e.g. by the above-mentioned QMath and LaTeXML tools. For unrestricted LaTeX this is a hard problem, due to the inherent ambiguity of mathematical notation. In a medium to long term perspective, linguistic techniques taking into account contextual information can be expected to solve this problem [45].

It is not always necessary that the author inputs a formula from scratch. Formulas may already exist in a mashup—embedded into a document that the mashup enriches, or resulting from an earlier step of computation. For sufficiently annotated formulas, as specified in Sect. 6.3.2.3, the JOBAD toolkit developed in our research group (see Sect. 6.4.1 for details) offers functions that determine the content markup associated to the range of presentation markup selected by the user. With cross-linked parallel markup this even works for subterms, as shown in Fig. 6.2.

#### 6.3.3.4 User Interface Output Components

The output of a mathematical mashup can consist of structured text, formulas, or graphics. Structured text output, e.g. tabulating numerical results of a computation, does not work differently from non-mathematical mashups and is therefore not detailed here. Formula output has been covered in Sect. 6.3.3.2.

Plots are an important and well-supported kind of mathematical and statistical graphics. For example, Flot [42] is a JavaScript library for producing graphical plots of arbitrary datasets, using the HTML canvas element. The mashup by Nakano et al., mentioned in Sect. 6.3.3.1, employs Flot for plotting graphs of functions. However, implementations that plot a sequence of data points do not generally live up to the complexity of mathematical functions, as they may have poles or other discontinuities. JSXGraph [64] is a library that does; in addition to function plotting, it supports interactive geometry, charting, and data visualization. Both Flot and JSXGraph allow the user to interact with their drawings; JSXGraph even supports multi-touch.

#### 6.3.3.5 Data Published for Reuse

Huge amounts of mathematical data have been published on the Web, but only a small fraction in a machine-friendly form. This section first explains why many collections of mathematical data are not suitable for reuse in mashups, and then points out the few ones that are.

Most of the large existing collections of mathematical knowledge either target human end-users, or, when they are machine-comprehensible, their representation is specialized to a single system. Of the collections mentioned in Sect. 6.2, Zentralblatt MATH [144], MathSciNet [4], DLMF [39], MathWorld [138], PlanetMath [111], Wikipedia [115] and the other wikis completely target human end-users. Some of them have machine-friendly APIs (cf. MathSciNet's "MR Lookup" [3], or the web service API of the MediaWiki engine that drives Wikipedia [96]), but these APIs are not uniform and thus not interoperable. Similarly, existing libraries of formalized mathematics, such as the MML [101], have been designed for automated processing but are only understood by one system. Some of the systems allow to produce representations of the libraries in a custom XML format. Since there are (to the best of our knowledge) no publicly hosted versions, these libraries cannot be considered readily available for mashup purposes, but establishing a hosting service would be rather simple. While we have suggested OMDoc (cf. Sect. 6.3.2.4) as a common interchange format among such systems [67, Chap. 25.2], it has not yet been adopted on a large scale. Another knowledge collection that aims at machine-comprehensibility via XML but largely uses its own XML vocabulary is Connexions [28]. It uses Content MathML for the formulas, but the course documents (with explicit markup of definitions, rules, examples and exercises) and modules (including links and metadata) are written in a custom XML language [30], even though—at least for some aspects of the knowledge represented—more common vocabularies would exist (cf. the review in [78, Sect. 4.1.5]). There are of course other educational repositories, but to the best of the authors' knowledge these do not expose the mathematical knowledge in a reusable way, but just the educational/pedagogical metadata; for example, i2geo (cf. Sect. 6.2.2) exposes a LOM record (Learning Object Metadata) for each resource [57], which follows the LOM standard [59] with i2geo-specific extensions [53].

Linked data (cf. Sect. 6.3.2.4) is an approach to publishing data in a uniform way for reusability. Major linked open datasets contain information that could be relevant in a mathematical context, but it is not made sufficiently explicit. For example, DBpedia [37], the linked open dataset derived from Wikipedia, is not capable of making any mathematically relevant information explicit beyond a rough categorization by topic, as the Wikipedia sources only contain presentation-oriented LaTeX formulas. Statistical datasets as, e.g., published by governments, usually contain a lot of numbers (statistical data points), but hardly any information of how they have been derived (e.g. by direct observation, or by mathematical computation from other data points; see [132] for a problem statement).

As of 2012, linked open datasets with relevant mathematical knowledge are in their infancy. We have published the OpenMath CDs (cf. Sect. 6.3.2.2), containing peer-reviewed semiformal descriptions of 260 mathematical symbols—those built into non-strict Content MathML and some more—as linked open data in 2011 [77]. In 2012, we published the Mathematics Subject Classification (MSC) as a linked open dataset [79, 89]. While these datasets provide two fundamental aspects of mathematical knowledge on the Web in a machine-comprehensible way—descriptions of the most commonly used mathematical operations, and a classification of all subjects of mathematical research and application—further work needs to be done towards interlinking them among each other and with other existing linked datasets (such as those mentioned above), and towards annotating legacy collections of mathematical knowledge (such as those mentioned initially in this section) with pointers to these datasets. We refer to [78, Sect. 6.1] for a detailed agenda.

### *6.3.4 Issues and Open Points*

The foundational representation issues for enabling mashups with mathematical formulas and mathematical knowledge have been solved, and the web standards have been largely been established. There are some minor issues in the integration of the technologies into hybrid document markup systems like HTML5, but they are mostly being addressed in the ongoing standardization process.

The main remaining issue is the prevalence of presentation markup in mathematical practice, and our difficulties in semantics extraction, i.e. transforming it into content markup that can be used as a basis for mashup technologies. The semantics extraction problem is essentially equivalent to the information extraction problems from natural language, but in the case of mathematical/technical documents it is aggravated by the fact that mathematics (i) uses custom notation definitions that are particular to the respective community of practice, (ii) cannot be covered fully by standard concept and operator dictionaries (such as those of OpenMath/MathML), but introduces (and discharges) concepts, symbols, and notations dynamically (using the paradigmatic definition/theorem/proof forms), (iii) and tries to avoid duplication and redundancy of content, which allows to use information extraction algorithms with lower precision and recall.

Given that automated semantic annotation of legacy document collections using information extraction techniques is still in its infancy, we are missing large linked open datasets like DBpedia that could kick-start the explosion of mashup services we are seeing in other areas. First technologies that allow authors to input content-marked up formulas—and more generally semantically annotated mathematical documents—have been developed, but engender additional effort on the part of the authors, which would only be justifiable, if more services and mashup system were available. Thus the lack of semantics extraction techniques raises a prisoner's dilemma for the individual author and a chicken-and-egg problem for MKM; see [69] for additional discussion.

### *6.3.5 Evaluations*

Evaluation criteria in the MKM domain often differ from those commonly applied to mashups. This section first explains the difference and then mentions a few known examples of evaluations that are relevant to the development of mathematical knowledge mashups.

MKM research has so far been biased towards formalized representations and applications in symbolic computation—less so towards web applications with a human user interface [21]. For MKM tools with a formal background, such as proof assistants or computer algebra systems, it is common to formally verify the correctness of their operation, as far as possible, on paper or using machine support. There is hardly any track record of systematic performance or usability evaluations, which would be useful for mashups. Performance evaluation has been restricted to the evaluation of systems that do symbolic or numeric computation; see, e.g., the world championship for automated theorem proving [123, 124] or the ISSAC conference series (International Symposium on Symbolic and Algebraic Computation [60]). Performance evaluation of actual *MKM* technologies may not have been necessary so far, as large-scale deployment of advanced MKM technology is still in its infancy (cf. [72]).

Among the tools from which mathematical knowledge mashups can be built, formula editors have most frequently been evaluated for usability [6, 65, 74, 85]. Evaluation methods included performance testing—for example how many key presses or clicks users need for creating a given formula—and questionnaires assessing subjective usability. William H. Billingsley has evaluated the usability of a graphical notation for mathematical objects and proofs, consisting of composable tiles, for students submitting solutions to exercises [15]. Test subjects were asked to prove a given set of theorems; Billingsley analyzed their solution attempts and classified the mostly unstructured feedback they had given by features of the software and aspects of the mathematical domain studied. Furthermore, he evaluated the notation against a Cognitive Dimensions of notations[10] questionnaire.

The pedagogical support of the ActiveMath integrated eLearning environment has been evaluated for usability [98]. However, that publication focuses on the qualitative results, not on the method of evaluation, and the mashup-like integration of

[10]"An approach to analysing the usability of information artifacts" [16], which has also been applied to mashups [40, 80].

different services in ActiveMath was not in the focus of that evaluation. In previous work, we have evaluated the usability of a wiki that integrates browsing, editing, publication, and discussion services for collaboratively maintaining OpenMath Content Dictionaries [76, Chap. 10] in three steps: analyzing user-generated content, surveying the target community about the perceived utility of the wiki and their satisfaction with it, and conducting supervised experiments to assess learnability and effectiveness.

Where usability evaluations are commonly conducted via experiments with test users, mathematical applications have also been subject to *conceptual* analyses; cf., e.g., a structured comparison of the behaviors of visual formula editors [110], or an added-value analysis of a formula search engine [70].

## 6.4 Applications with Math Mashup Techniques

This section presents mathematical knowledge mashups that have been developed in our research group. As we have argued above, any mathematical mashup system has to combine semantic annotations with formula support—where we can interpret parallel markup as semantic annotations of formula presentations. As this is a rather big investment, there are few systems that are capable of managing mathematical content in a way that enables mashups. These include computer algebra systems with document-oriented interfaces (such as the notebooks of Mathematica, a commercial system with an open API but a proprietary content representation format [87]) and the technology stack based on our own open, web standards compliant OMDoc format, including our own tools, presented in the following subsections, but also the ActiveMath eLearning system and its authoring tools [81].

Rather than solving individual specific problems using mashup technology, we aim at designing general, open, and extensible *architectures*. While our reference implementations do provide services for specific problems, such as looking up the definition of a symbol that occurs in a formula, we also envision that third parties extend and customize them with their own services.

JOBAD (Sect. 6.4.1) is an architecture for integrating assistive services into semantically annotated HTML5 documents. Planetary (Sect. 6.4.2) integrates JOBAD plus further front-end and back-end components into a web-based content management system. The SAlly framework generalizes this approach to a mashup enabler for arbitrary document-oriented interfaces, including desktop applications.

### *6.4.1 JOBAD, a Toolkit for Integrating Assistive Services into Interactive Documents*

JOBAD [43, 62] is an architecture for integrating interactive services into documents—services that assist readers in adapting the document's appearance to their

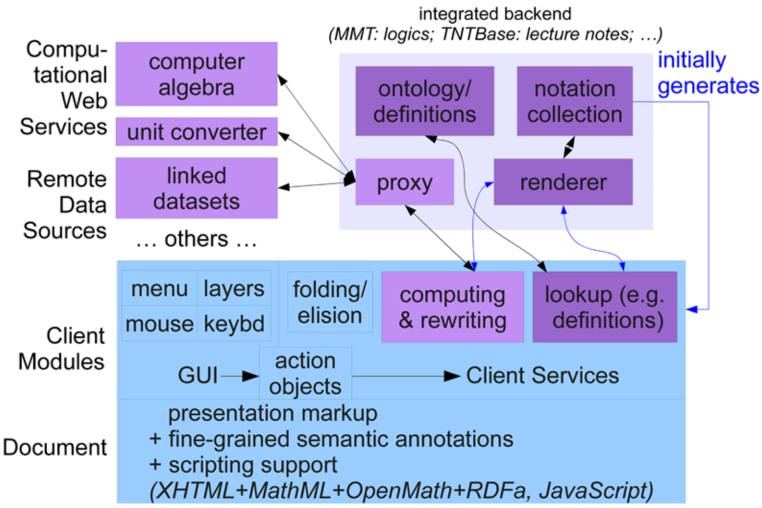



**Fig. 6.4** The JOBAD architecture (concrete implementation systems/languages in *italics*)

preferences or in looking up additional information and displaying it right in place, i.e. without forcing them to switch their attention away from the document.

The JOBAD architecture (cf. Fig. 6.4) assumes documents that have been sufficiently prepared for interaction; Sect. 6.3.2.3 states the particular requirements for the markup of mathematical formulas.[11] These annotations serve as hooks for assistive services, whose client side runs within the document. The level of interaction afforded by JOBAD depends on the service plugins loaded and on the depth of annotations in the documents; the set of functions applicable to a mathematical expression selected by the user—either by clicking on it or selecting a range as shown in Fig. 6.2—depends on the structure of the selection and its annotations. For example, if a unit conversion service is available, it is only applicable to expressions that consist of a unit symbol applied to a quantity (cf. [24] for further background). If a concept definition lookup service is available, it is only applicable to symbols in a mathematical object, or to technical terms in a text, either of which is a concrete occurrence of that concept. Our JOBAD implementation makes the functions registered by its service plugins accessible in a context menu, at each time only showing those that claim to be applicable to the current selection. We have designed and implemented

1. client services that rely exclusively on annotations given in the rendered document—mostly for customizing its appearance, such as folding away subexpressions of long formulas, or document sections (as shown in Fig. 6.6)—and thus also work off-line,

[11]The original abbreviation means "JavaScript API for OMDoc-Based Interactive Documents", but JOBAD-enabled documents do not have to be generated from an OMDoc source.

2. client services that retrieve additional information from background ontologies on the primary server back-end that has also generated the document—such as definition lookup from a CD collection on the server, or in-place unit conversion using conversion rules given for unit symbols in such CDs, and
3. client services that retrieve information from arbitrary external sources, with the primary server back-end serving as a proxy due to the "same origin policy" [143, Part 2]. This is the case for information lookup from Wolfram Alpha [35]; similarly, definition lookup could be extended to arbitrary external content dictionaries.

The client's communication with a web source may be as simple as downloading some data, e.g. the rendering of the definition of a symbol, from a URL, but it may also involve POSTing an expression to a web service and receiving a rewritten expression, as in the case of unit conversion. Information retrieved from the Web may be displayed in a tooltip-like popup—as for definition lookup and Wolfram Alpha lookup—or result in a part of the document, usually the selected mathematical expression, being rewritten—as for unit conversion.

### *6.4.2 The Planetary eMath 3.0 System*

Planetary is a comprehensive framework for semantic publishing and knowledge management, which has been instantiated prototypically in a variety of settings to validate the framework and to support communities. The portals realized with Planetary range from eLearning systems over scientific archives to theorem prover interfaces. All share a common basic architecture (see Fig. 6.5), which integrates previously developed components and interfaces into a central content management system (CMS; here: Drupal) that mediates all user interaction:

1. the TNTBase document database (cf. Sect. 6.3.3.2) for storage and rendering,
2. the LaTeXML daemon (cf. Sect. 6.3.3.3) for transforming TeX/LaTeX document fragments—not just formulas—to HTML5, including MathML formulas, SVG diagrams, and RDFa metadata,
3. sTeX [68, 121], a semantic variant of LaTeX that can be transformed to the OMDoc XML language (cf. Sect. 6.3.3.5) and further to semantically annotated HTML (also covered by the JOMDoc library introduced in Sect. 6.3.3.2),
4. and the JOBAD toolkit (see Sect. 6.4.1) for embedding semantic services into web documents.

The Planetary system is mashup-based at the heart: The CMS supplies management and interaction at the "container level", i.e., without ever looking into the documents it manages, which is why we actually interpret the abbreviation "CMS" as *container* management system here. All other services are based on *structured document content* and background ontologies (providing, e.g., symbol definitions), which are provisioned by the TNTBase document database and the RDF triple store in Planetary. These subsystems, which are accessed via RESTful HTTP interfaces and may therefore be installed on different hosts, also perform semantic services,

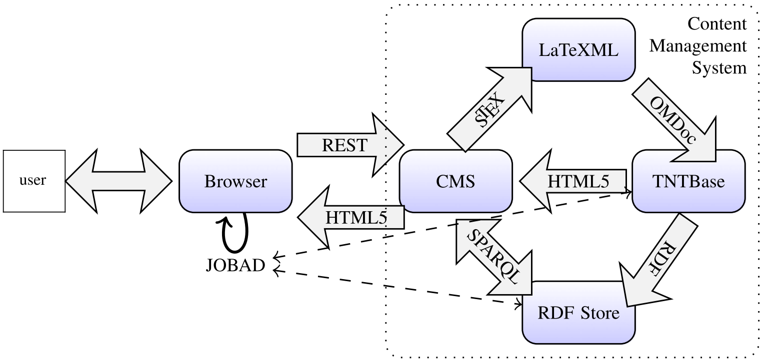



**Fig. 6.5** Architecture of the Planetary system

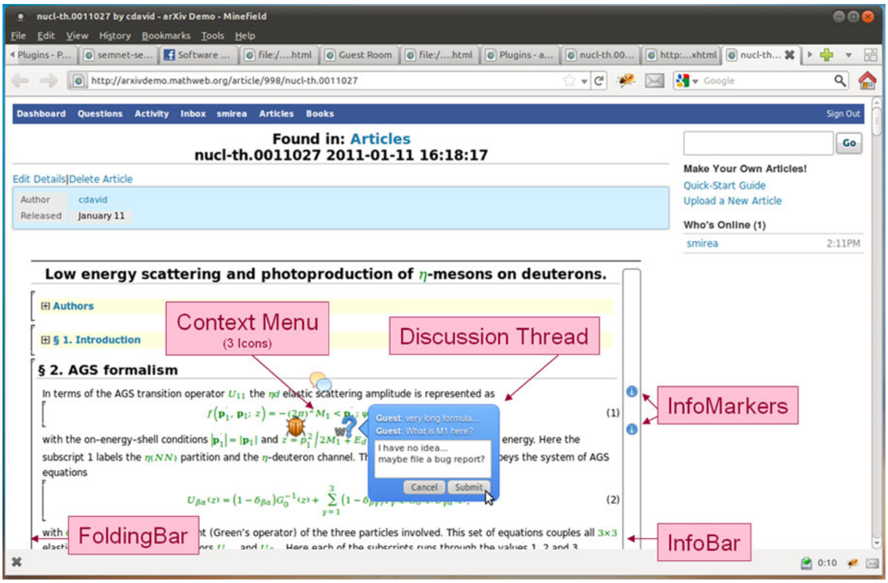



**Fig. 6.6** Mashing up semantic services into arXiv.org documents in Planetary

which are integrated (mashed up) into the mathematical documents via the JOBAD toolkit.

Our different instances of the Planetary system have documents with different levels of annotation, and thus afford different levels of interaction, as explained in Sect. 6.4.1: from simple folding and localized commenting services in a front-end system for the arXiv.org preprint library (see Fig. 6.6) to an in-place type reconstruction and elision of arguments and brackets for the fully formal LATIN atlas of logical theories [25].

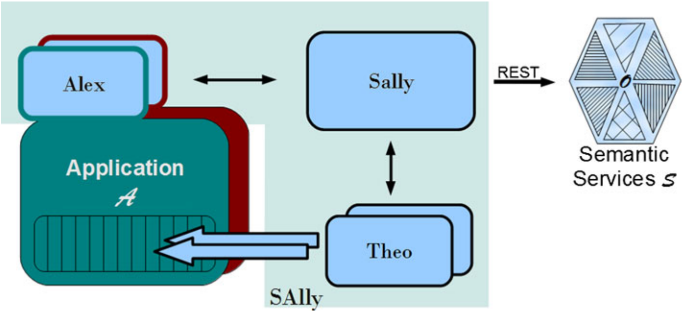



**Fig. 6.7** SAlly as a mashup enabler for semantic allies

### *6.4.3 Semantic Allies: Mashing up User Interfaces*

The basic mechanism behind the Planetary system introduced in the previous section is the mashup of semantic mathematical services in a web-based system. Planetary acts as a *document player* that presents documents to the user for interaction, where annotations in the documents provide hooks for JOBAD services. From this general perspective, we observe that document players for mathematical documents are ubiquitous not only on the Web, but also on the *desktop* and *mobile devices*. Furthermore, almost all documents with mathematical content are supported by some of these document players. Examples include spreadsheets, word processors, symbolic and numerical software, and even slide presenters. Users (readers as well as authors of documents) have usually invested heavily into becoming efficient in interacting with their preferred document players, and are therefore unlikely to leave them. Therefore, such document players are interesting candidates for integrating mathematical services beyond the built-in ones via mashup technology.

The SAlly framework [36] provides a mashup enabler for such situations. It builds on the observation that a service feels embedded into an application if it occupies a screen area that is part of the area originally claimed by the application itself. This perception is amplified, if a service and its request refer to the local semantic objects. In particular, the service does not need to be implemented as application-specific invasive technology, and can be provided as a mashup service by a *semantic ally*, which in the SAlly framework has three components (see Fig. 6.7):

- a platform-independent semantic interaction manager “SAlly” (as a semantic ally), which has access to semantic services, and that
- partners with a set of invasive, thin, application-specific API Alex, which essentially only manages user interface events in the application, and that
- has access to a set of application-independent screen area managers Theo that can render the available services.

We have implemented the SAlly framework with Alexes for OO Calc, MS Excel, and a Theo based on XULRunner [140]—the layout and communication engine behind Mozilla Firefox and Thunderbird. Figure 6.8 shows the result of mashing up the

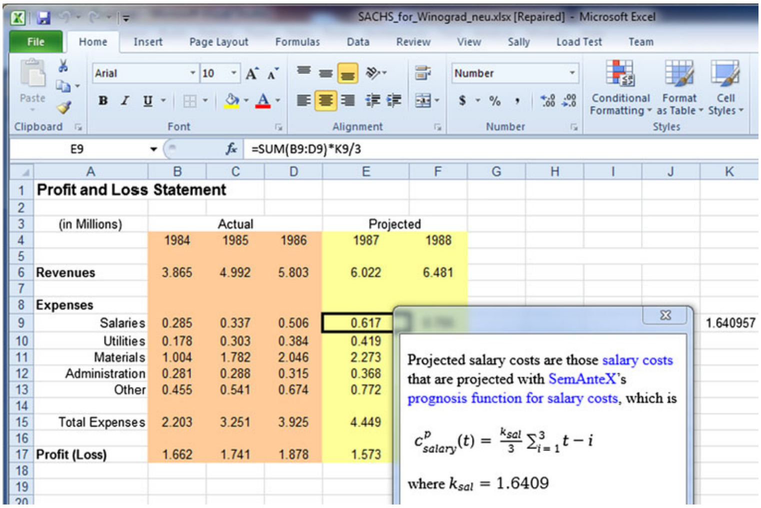



**Fig. 6.8** Definition lookup via SAlly in a spreadsheet

JOBAD service for looking up definitions from a Planetary back-end into an MS Excel spreadsheet using the SAlly framework.

## 6.5 Conclusion, Lessons Learned, and Future Prospects

As of summer 2012, proper mathematical *knowledge* mashups hardly exist (in contrast to data mashups that perform simple numeric computations), but, as this chapter proves, the technology to build them is there and waiting to be applied. In fact it is already being applied in mashup-like settings, as our Planetary system demonstrates. Compared to other mashup application domains, the technology for mathematical knowledge mashups just took a few more years to consolidate. This is mainly due to the inherent complexity of mathematical formulas, which are a characteristic feature of mathematical applications and distinguish them from other applications.

But the added complexity of the application domain, which has delayed the provision of mashup services, will in our opinion also lead to added usefulness of the mashup systems when they eventually arrive. Math is generally considered hard—partly because of the embedded special-purpose sublanguage of formulas, and we claim that readers can reasonably expect more help in interacting with formulas. This is just what mathematical knowledge mashup services can offer. Indeed, mathematical formulas are often more semantic than mere language, and therefore afford more semantic services that can be mashed up.

**Acknowledgements** The authors would like to thank Paul Libbrecht for his thorough review and constructive feedback.

## References


1. ACTIVEMATH. http://www.activemath.org
2. Alama J, Brink K, Mamane L, Urban J (2011) Large formal wikis: issues and solutions. In: [34], pp 133–148
3. American Mathematical Society. http://www.ams.org/mathscinet/help/mr_lookup_help.html
4. American Mathematical Society. MathSciNet mathematical reviews on the net. http://www.ams.org/mathscinet/
5. Ankolekar A, Krötzsch M, Tran T, Vrandečić D (2008) The two cultures: mashing up Web 2.0 and the semantic web. J Web Semant 6(1):70–75
6. Anthony L, Yang J, Koedinger KR (2005) Evaluation of multimodal input for entering mathematical equations on the computer. In: van der Veer G, Gale C (eds) CHI'05 extended abstracts on human factors in computing systems. ACM, New York, pp 1184–1187. doi:10.1145/1056808.1056872
7. Asperti A, Padovani L, Sacerdoti Coen C, Guidi F, Schena I (2003) Mathematical knowledge management in HELM. Ann Math Artif Intell (special issue on mathematical knowledge management) 38(1–3), 27–46
8. Asperti A, Bancerek G, Trybulec A (eds) (2004) Mathematical knowledge management, MKM'04. Lecture notes in artificial intelligence, vol 3119. Springer, Berlin
9. Asperti A, Geuvers H, Natarajan R (2009) Social processes, program verification and all that. Math Struct Comput Sci 19(5):877–896
10. Ausbrooks R, Buswell S, Carlisle D, Chavchanidze G, Dalmas S, Devitt S, Diaz A, Dooley S, Hunter R, Ion P, Kohlhase M, Lazrek A, Libbrecht P, Miller B, Miner R, Sargent M, Smith B, Soiffer N, Sutor R, Watt S (2010) Mathematical markup language (MathML) version 3.0. W3C recommendation, World Wide Web Consortium (W3C). http://www.w3.org/TR/MathML3
11. Autexier S, Campbell J, Rubio J, Sorge V, Suzuki M, Wiedijk F (eds) (2008) Intelligent computer mathematics. Lecture notes in artificial intelligence, vol 5144. Springer, Berlin
12. Autexier S, Calmet J, Delahaye D, Ion PDF, Rideau L, Rioboo R, Sexton AP (eds) (2010) Intelligent computer mathematics. Lecture notes in artificial intelligence, vol 6167. Springer, Berlin
13. Baez J (2010) Math blogs. Notices of the AMS, p 333. http://www.ams.org/notices/201003/rtx100300333p.pdf
14. Barany MJ (2010) '[B]ut this is blog maths and we're free to make up conventions as we go along': Polymath1 and the modalities of 'massively collaborative mathematics'. In: Ayers P, Ortega F (eds) Proceedings of the 6th international symposium on wikis and open collaboration (WikiSym). ACM, New York. http://www.wikisym.org/ws2010/Proceedings/
15. Billingsley WH (2008) The intelligent book: technologies for intelligent and adaptive textbooks, focussing on discrete mathematics. PhD thesis, University of Cambridge. http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-719.pdf
16. Blackwell A, Green T Cognitive dimensions of notations resource site. http://www.cl.cam.ac.uk/~afb21/CognitiveDimensions/
17. Borwein J, Stanway T (2005) Knowledge and community in mathematics. Math Intell 27(2):7–16
18. Buswell S, Caprotti O, Carlisle DP, Dewar MC, Gaëtano M, Kohlhase M (2004) The OpenMath standard, version 2.0. Tech rep, The OpenMath Society. http://www.openmath.org/standard/om20
19. Byrd D (2006) Extremes of conventional music notation. http://www.informatics.indiana.edu/donbyrd/CMNExtremes.htm
20. Caprotti O, Dewar M, Turi D (2004) Mathematical service matching using description logic and OWL. In: [8], pp 73–87
21. Carette J, Farmer W (2009) A review of mathematical knowledge management. In: [22], pp 233–246

22. Carette J, Dixon L, Sacerdoti Coen C, Watt SM (eds) (2009) MKM/Calculemus proceedings. Lecture notes in artificial intelligence, vol 5625. Springer, Berlin
23. Cartier P (2010) Can we make mathematics universal as well as fully reliable? In: [12], invited talk
24. Cîrlănaru M, Ginev D, Lange C (2011) Authoring and publishing of units and quantities in semantic documents. In: García Castro R, Fensel D, Antoniou G (eds) The semantic web: ESWC 2011 workshops. Lecture notes in computer science, vol 7117. Springer, Heidelberg, pp 202–216. http://kwarc.info/clange/pubs/eswc2011-units.pdf
25. Codescu M, Horozal F, Kohlhase M, Mossakowski T, Rabe F (2011) Project abstract: logic atlas and integrator (latin). In: [34], pp 289–291
26. Cohen AM, Cuypers H, Verrijzer R (2010) Mathematical context in interactive documents. Math Comput Sci 3(3):331–347
27. Conference on intelligent computer mathematics (CICM). http://cicm-conference.org
28. Connexions. http://cnx.org
29. Connexions MathML editor. http://cnx.org/matheditor
30. Connexions—XML languages. http://cnx.org/help/authoring/xml
31. Corlosquet S, Delbru R, Clark T, Polleres A, Decker S (2009) Produce and consume linked data with Drupal! In: Bernstein A, Karger DR, Heath T, Feigenbaum L, Maynard D, Motta E, Thirunarayan K (eds) The semantic web—ISWC 2009. Lecture notes in computer science, vol 5823. Springer, Berlin, pp 763–778
32. Costantini M, Konovalov A, Nicosia M, Solomon A (2011) GAP package OpenMath. http://www.gap-system.org/Packages/openmath.html
33. Datamasher. http://www.datamasher.org
34. Davenport J, Farmer W, Rabe F, Urban J (eds) (2011) Intelligent computer mathematics. Lecture notes in artificial intelligence, vol 6824. Springer, Berlin
35. David C, Lange C, Rabe F (2010) Interactive documents as interfaces to computer algebra systems: JOBAD and Wolfram—Alpha. In: Delahaye D, Rioboo R (eds) Calculemus (emerging trends). Centre d'Étude et de Recherche en Informatique du CNAM (Cédric), Paris, pp 13–30. https://svn.omdoc.org/repos/jomdoc/doc/pubs/calculemus10/jobad-cas.pdf
36. David C, Jucovschi C, Kohlhase A, Kohlhase M (2012) Semantic alliance: a framework for semantic allies. In: [61], pp 49–64. http://kwarc.info/kohlhase/submit/mkm12-SAlly.pdf
37. DBpedia. http://dbpedia.org
38. Developer apps showcase—data.gov. http://www.data.gov/developers/showcase
39. Digital library of mathematical functions. http://dlmf.nist.gov
40. Ennals R, Gay D (2007) User-friendly functional programming for web mashups. In: Proceedings of the 12th ACM SIGPLAN international conference on functional programming, ICFP'07. ACM, New York, pp 223–234. doi:10.1145/1291151.1291187
41. Farmer WM (2004) MKM: a new interdisciplinary field of research. Bull ACM (special interest group on symbolic and automated mathematics—SIGSAM) 38(2):47–52
42. flot—attractive JavaScript plotting for jQuery. http://flot.googlecode.com
43. Giceva J, Lange C, Rabe F (2009) Integrating web services into active mathematical documents. In: [22], pp 279–293. https://svn.omdoc.org/repos/jomdoc/doc/pubs/mkm09/jobad/jobad-server.pdf
44. Ginev D. The LaTeXml daemon: editable math for the collaborative web. http://latexml.mathweb.org
45. Ginev D (2011) The structure of mathematical expressions. Master's thesis, Jacobs University Bremen. http://kwarc.info/people/dginev/publications/DeyanGinev_MScThesis.pdf
46. Ginev D, Stamerjohanns H, Kohlhase M (2011) The LaTeXML daemon: editable math on the collaborative web. In: [34], pp 292–294. https://svn.kwarc.info/repos/arXMLiv/doc/cicm-systems11/paper.pdf
47. González Palomo A (2006) QMath: a human-oriented language and batch formatter for OMDoc. In: [67], Chap 26.2. http://omdoc.org/pubs/omdoc1.2.pdf
48. González Palomo A (2006) Sentido: an authoring environment for OMDoc. In: [67], Chap 26.3. http://omdoc.org/pubs/omdoc1.2.pdf

49. Hammond K, Horn P, Konovalov A, Linton S, Roozemond D, Zain AA, Trinder P (2010) Easy composition of symbolic computation software: a new lingua franca for symbolic computation. In: Proceedings of the 2010 international symposium on symbolic and algebraic computation (ISSAC). ACM, New York, pp 339–346
50. Heath T, Bizer C (2011) Linked data: evolving the web into a global data space, 1st edn. Synthesis lectures on the semantic web: theory and technology. Morgan & Claypool, San Rafael. http://linkeddatabook.com
51. Heintz B (2000) Die Innenwelt der Mathematik. Zur Kultur und Praxis einer beweisenden Disziplin. Springer, Vienna
52. HELM. http://helm.cs.unibo.it
53. Hendriks M, Libbrecht P, Creus-Mir A, Dietrich M (2008) Metadata specification. Deliverable D2.4. Intergeo. http://i2geo.net/files/deliverables/D2.4-Metadata-Spec.pdf
54. Hickson I (2012) HTML5. W3C working draft, World Wide Web Consortium (W3C). http://www.w3.org/TR/2011/WD-html5-20120329/
55. Horn P. MuPAD OpenMath package. http://mupad.symcomp.org/
56. Horn P, Roozemond D (2009) OpenMath in SCIEnce: SCSCP and POPCORN. In: [22], pp 474–479
57. i2geo—i2g metadata. http://i2geo.net/xwiki/bin/view/About/I2GMetadata
58. i2geo—interoperable interactive geometry for Europe. http://i2geo.net
59. IEEE Learning Technology Standards Committee (2002) Standard for learning object metadata. Tech Rep 1484.12.1, IEEE
60. ISSAC—international symposium on symbolic and algebraic computation. http://www.issac-conference.org/
61. Jeuring J, Campbell JA, Carette J, Dos Reis G, Sojka P, Wenzel M, Sorge V (eds) (2012) Intelligent computer mathematics. Lecture notes in artificial intelligence, vol 7362. Springer, Berlin
62. JOBAD framework—JavaScript API for OMDoc-based active documents. http://jobad.omdoc.org
63. JOMDoc project—Java library for OMDoc documents. http://jomdoc.omdoc.org
64. JSXGraph—dynamic mathematics with JavaScript. http://jsxgraph.org
65. Kawata T, Kataoka M, Kai H, Tamura Y (2008) A MathML content markup editor on the XFY. In: Smirnova E, Watt SM (eds) Applications for computer algebra
66. Knowledge management (2009). http://en.wikipedia.org/w/index.php?title=Knowledge_management&oldid=329227520
67. Kohlhase M (2006) OMDoc—an open markup format for mathematical documents [version 1.2]. Lecture notes in artificial intelligence, vol 4180. Springer, Berlin. http://omdoc.org/pubs/omdoc1.2.pdf
68. Kohlhase M (2008) Using LaTeX as a semantic markup format. Math Comput Sci 2(2):279–304. https://svn.kwarc.info/repos/stex/doc/mcs08/stex.pdf
69. Kohlhase A, Kohlhase M (2004) CPoint: dissolving the author's dilemma. In: [8], pp 175–189. http://kwarc.info/kohlhase/papers/mkm04.pdf
70. Kohlhase A, Kohlhase M (2007) Reexamining the MKM value proposition: from math web search to math web research. In: Kauers M, Kerber M, Miner R, Windsteiger W (eds) Towards mechanized mathematical assistants. MKM/Calculemus. Lecture notes in artificial intelligence, vol 4573. Springer, Berlin, pp 266–279. http://mathweb.org/projects/mws/pubs/mkm07.pdf
71. Kohlhase M, Müller C, Rabe F (2008) Notations for living mathematical documents. In: [11], pp 504–519. http://omdoc.org/pubs/mkm08-notations.pdf
72. Kohlhase M, Rabe F, Zholudev V (2010) Towards MKM in the large: modular representation and scalable software architecture. In: Autexier S, Calmet J, Delahaye D, Ion P, Rideau L, Rioboo R, Sexton A (eds) Intelligent computer mathematics. Lecture notes in computer science, vol 6167. Springer, Berlin, pp 370–384
73. Kohlhase M, Corneli J, David C, Ginev D, Jucovschi C, Kohlhase A, Lange C, Matican B, Mirea S, Zholudev V (2011) The planetary system: Web 3.0 & active documents for stem.

Proc Comput Sci 4:598–607. doi:10.1016/j.procs.2011.04.063, https://svn.mathweb.org/repos/planetary/doc/epc11/paper.pdf (finalist at the Executable Paper Grand Challenge)
74. Kovalchuk A, Levitsky V, Samolyuk I, Yanchuk V (2010) The formulator MathML editor project: user-friendly authoring of content markup documents. In: [12], pp 385–397
75. Lakatos I (1976) Proofs and refutations. Cambridge University Press, Cambridge
76. Lange C (2011) Enabling collaboration on semiformal mathematical knowledge by semantic web integration. Studies on the semantic web, vol 11. AKA Verlag/IOS Press, Heidelberg/Amsterdam
77. Lange C (2011) Krextor—an extensible framework for contributing content math to the web of data. In: [34], pp 304–306. http://kwarc.info/clange/pubs/krextor-system.pdf
78. Lange C (2012) Ontologies and languages for representing mathematical knowledge on the semantic web. Semant Web J 4(2):119–158. doi:10.3233/SW-2012-0059, http://www.semantic-web-journal.net/content/ontologies-and-languages-representing-mathematical-knowledge-semantic-web
79. Lange C, Ion P, Dimou A, Bratsas C, Sperber W, Kohlhase M, Antoniou I (2012) Bringing mathematics to the web of data: the case of the mathematics subject classification. In: Simperl E, Cimiano P, Polleres A, Corcho O, Presutti V (eds) The semantic web. Lecture notes in computer science, vol 7295. Springer, Berlin, pp 763–777. doi:10.1007/978-3-642-30284-8_58, http://kwarc.info/clange/pubs/eswc2012-msc-skos.pdf
80. Le-Phuoc D, Polleres A, Hauswirth M, Tummarello G, Morbidoni C (2009) Rapid prototyping of semantic mash-ups through semantic web pipes. In: Quemada J, León G, Maarek YS, Nejdl W (eds) Proceedings of the 17th WWW conference. ACM, New York, pp 581–590
81. Libbrecht P (2010) What you check is what you get: authoring with jEditOQMath. In: 10th international conference on advanced learning technologies (ICALT). IEEE Press, New York, pp 682–686
82. Libbrecht P, Kortenkamp U, Mercat C (2009) I2Geo: a web-library of interactive geometry. In: Sojka P (ed) Towards digital mathematics library. Masaryk University Press, Brno, pp 95–106
83. Manzoor S, Libbrecht P, Ullrich C, Melis E (2006) Authoring presentation for OpenMath. In: Kohlhase M (ed) Mathematical knowledge management, MKM'05. Lecture notes in artificial intelligence, vol 3863. Springer, Berlin, pp 33–48
84. Marchiori M (2003) The mathematical semantic web. In: Asperti A, Buchberger B, Davenport JH (eds) Mathematical knowledge management, MKM'03. Lecture notes in computer science, vol 2594. Springer, Berlin, pp 216–223 (keynote)
85. Marquès D, Eixarch R, Casanellas G, Martínez B (2006) WIRIS OM tools: a semantic formula editor. In: Libbrecht P (ed) Mathematical user interfaces workshop 2006. http://www.activemath.org/~paul/MathUI06
86. MathDox—interactive mathematics. http://www.mathdox.org
87. Mathematica. http://www.wolfram.com/products/mathematica/
88. Mathematics subject classification MSC2010 (2010). http://msc2010.org
89. Mathematics subject classification (MSC) SKOS (2012). http://msc2010.org/resources/MSC/2010/info/
90. MathJax: beautiful math in all browsers. http://mathjax.com
91. MathML (software support/Web browsers) (2012). http://en.wikipedia.org/w/index.php?title=MathML&oldid=482267822#Web_browsers
92. MathML software—converters. http://www.w3.org/Math/Software/mathml_software_cat_converters.html
93. MathML software—editors. http://www.w3.org/Math/Software/mathml_software_cat_editors.html
94. MathOverflow. http://mathoverflow.net
95. MathPlayer. http://www.dessci.com/en/products/mathplayer
96. McCabe D, Garrett A. MediaWiki—api. http://www.mediawiki.org/w/index.php?title=API:Main_page&oldid=574291

97. Melis E, Andrés E, Büdenbender J, Frischauf A, Goguadze G, Libbrecht P, Pollet M, Ullrich C (2001) ActiveMath: a generic and adaptive web-base learning environment. Int J Artif Intell Educ 12(4):385–407
98. Melis E, Weber M, Andrès E (2003) Lessons for (pedagogic) usability of eLearning systems. In: Rossett A (ed) Proceedings of world conference on e-learning in corporate, government, healthcare, and higher education. AACE, Chesapeake, pp 281–284
99. Melis E, Goguadze G, Libbrecht P, Ullrich C (2009) Culturally adapted mathematics education with ActiveMath. Artif Intell Soc 24(3):251–265
100. Miller B. LaTeXML: a LaTeX to XML converter. http://dlmf.nist.gov/LaTeXML/
101. Mizar mathematical library. http://www.mizar.org/library
102. MONET—mathematics on the net. http://monet.nag.co.uk
103. Müller C (2010) Adaptation of mathematical documents. PhD thesis, Jacobs University Bremen. http://kwarc.info/cmueller/papers/thesis.pdf
104. Nakano H, Nagai T, Yunpeng J, Wannous M, Kita T (2011) Mashup approach for embedding algebraic manipulations, formulas and graphs in web pages. In: Learning environments and ecosystems in engineering education. IEEE Press, New York, pp 691–694
105. $n$-Category Café. http://golem.ph.utexas.edu/category/
106. nLab. http://ncatlab.org/
107. OMDoc. http://omdoc.org
108. OpenMath Society. OM 2 Presentation MathML XSLT test release. http://www.openmath.org/standard/omxsl/
109. OpenMath software and tools. http://www.openmath.org/software/
110. Padovani L, Solmi R (2004) An investigation on the dynamics of direct-manipulation editors for mathematics. In: [8], pp 302–316
111. PlanetMath.org—math for the people, by the people. http://planetmath.org
112. PlanetMath Redux.org—math for the people, by the people. http://planetmath.mathweb.org
113. Pólya G (1973) How to solve it. Princeton University Press, Princeton
114. Polymath blog. http://polymathprojects.org/
115. Portal: mathematics. http://en.wikipedia.org/w/index.php?title=Portal:Mathematics&oldid=329137789
116. ProgrammableWeb. http://www.programmableweb.com
117. ProofWiki. http://www.proofwiki.org
118. Robinson A, Voronkov A (eds) (2001) Handbook of automated reasoning, vols I–II. Elsevier/MIT Press, Amsterdam/Cambridge
119. SCIEnce project—symbolic computation infrastructure for Europe. http://www.symcomp.org/
120. SCIEnce EU project (2009) The Popcorn OpenMath representation. http://java.symcomp.org/FormalPopcorn.html
121. Semantic markup for LaTeX. http://trac.kwarc.info/sTeX/ (project homepage)
122. Semantic MediaWiki. http://semantic-mediawiki.org
123. Sutcliffe G, Suttner C. The CADE ATP system competition. http://www.cs.miami.edu/~tptp/CASC/
124. Sutcliffe G, Suttner C (2006) The state of CASC. AI Commun 19(1):35–48
125. Tricki. http://www.tricki.org
126. Truenumbers. http://www.truenum.com
127. Trzeciak J (1995) Writing mathematical papers in English. Gdańskie Wydawnictwo Oświatowe, Gdansk
128. Ullrich C (2008) Pedagogically founded courseware generation for web-based learning. Lecture notes in computer science. Springer, Berlin. http://www.springerlink.com/content/k604618p5351/
129. Unicode, Inc (2009) Unicode. http://www.unicode.org/versions/Unicode5.2.0/
130. Urban J, Alama J, Rudnicki P, Geuvers H (2010) A wiki for Mizar: motivation, considerations, and initial prototype. In: [12], pp 455–469

131. Vismor T (2012) Viewing mathematics on the internet. https://vismor.com/documents/site_implementation/viewing_mathematics/
132. Vrandečić D, Lange C, Hausenblas M, Bao J, Ding L (2010) Semantics of governmental statistics data. In: Proceedings of WebSci'10: extending the frontiers of society on-line. Web Science Trust. http://journal.webscience.org/400/
133. W3C math working group: example XSLT code for transforming XML languages for the web. http://web-xslt.googlecode.com
134. When can I use MathML? (2012). http://caniuse.com/mathml
135. WIRIS editor—a tool for graphical edition of mathematical formulas. http://www.wiris.com/content/view/20/
136. Wolfram—Alpha. http://www.wolframalpha.com
137. Wolfram—Alpha widgets. http://developer.wolframalpha.com/widgets/
138. Wolfram MathWorld. http://mathworld.wolfram.com
139. Working with MathML—Wolfram Mathematica 8 documentation (2011). http://reference.wolfram.com/mathematica/XML/tutorial/MathML.html
140. XULRunner runtime environment. https://developer.mozilla.org/en/XULRunner
141. Yacas computer algebra system. http://yacas.sourceforge.net/
142. Zacchiroli S (2007) User interaction widgets for interactive theorem proving. PhD thesis, Università di Bologna
143. Zalewski M (2010) Browser security handbook. Tech rep, Google. http://browsersec.googlecode.com
144. Zentralblatt MATH. http://www.zentralblatt-math.org
145. Zholudev V et al (2010) TNTBase—restful api. http://tntbase.org/wiki/restful
146. Zholudev V, Kohlhase M, Rabe F (2010) A [insert XML format] database for [insert cool application]. In: Proceedings of XML, Prague, 2010, pp 317–339. http://kwarc.info/vzholudev/pubs/XMLPrague.pdf
147. Zimmer J (2008) MathServe—a framework for semantic reasoning services. PhD thesis, Universität des Saarlandes

# Chapter 7
# Speech Mashups

**Giuseppe Di Fabbrizio, Thomas Okken, and Jay Wilpon**

Scotty: *"Computer—"*
Bones steps in quickly, picks up the "Mouse" and shoves it into Scotty's hand.
Scotty looks at the mouse, baffled, then puts it to his lips like a mike.
Scotty: (continuing) *"Hello? Computer...?"*
Nichols: (bewildered) *"Just use the keyboard..."*
Scotty: *"The keyboard... How quaint."*

Star Trek IV—The Voyage Home
Original movie script, March 11, 1986

**Abstract** Speech mashups are an emerging type of mashup that expose cloud-based speech and language processing technologies as web services. They allow researchers, practitioners, and developers to access commercial-grade speech recognition and text-to-speech systems without the need to install, configure, or manage speech processing software or equipment. This approach significantly lowers the barrier to build speech applications by having all the necessary components and tools available in the network. Compared to traditional mashups, they introduce a number of new concepts such as audio capturing, audio play-back, streaming media across the network, and resource configuration management.

## 7.1 Introduction

Speech interaction with computers—featured in many sci-fi movies as the ultimate communication between humans and machines—seemed a distant goal only a few years ago. But recent advances in speech and language processing research, together

G. Di Fabbrizio (✉) · T. Okken · J. Wilpon
Research, AT&T Labs, 180 Park Avenue, Florham Park, NJ, USA
e-mail: pino@research.att.com

T. Okken
e-mail: tokken@research.att.com

J. Wilpon
e-mail: jgw@research.att.com

B. Endres-Niggemeyer (ed.), *Semantic Mashups*,
DOI 10.1007/978-3-642-36403-7_7, 

with the availability of powerful cloud-based computation resources, are making natural spoken communication with computers a practical, reachable goal for a large spectrum of applications and devices.

Nowadays network-connected devices—either mobile or stationary—can easily extend their capabilities by off-loading highly demanding processing tasks to the cloud and focus on the functionalities that are better suited for portability, usability, or lower battery consumption. This paradigm, often defined as *ubiquitous computing* [10], applies to a vast number of scenarios where devices have access to large-scale processing and long-lived storage resources across a low-latency pervasive network at a fraction of the cost of traditional centralized computing solutions.

Converting text to speech to produce natural sounding utterances—text-to-speech (TTS) synthesis—and automatically converting speech into text—automatic speech recognition (ASR)—are the kind of challenging tasks necessary to integrate speech-interaction capabilities into an application. Both TTS and ASR are highly intensive processing tasks. They rely on a fairly complex technology involving multidisciplinary methodologies from research fields such as digital signal processing, statistical pattern recognition, machine learning, linguistics, and phonetics.

Mastering speech and language technologies requires, in addition to adequate computational resources, a solid background in these disciplines. Depending on the application, crucial procedures, like optimizing ASR performances or improving TTS intonation and pronunciation, are often closer to art than science. However, some important and recent milestones have made this challenging technology more accessible to a larger audience, and ultimately to developers not necessarily aware of "under the hood" details.

A first historical milestone that made speech technology more accessible to a large group of developers was VoiceXML [28], an XML-based programming language introduced by AT&T, IBM, Lucent, and Motorola in 1999 and published in March 2000 by the World Wide Web Consortium[1] (W3C). VoiceXML had a major impact in the speech application developer community because it introduced a standard interface to speech and telephony functionality in a market dominated by proprietary systems. VoiceXML pages are interpreted within a *voice browser* that exploits programming paradigms analogous to a traditional *web browser* environment, making the concept of spoken dialog interaction similar to web-form interpretation. Typically, in a voice browser, the fine-grained details of media synchronization, resource allocation, audio streaming routing, and telephony signaling are hidden and managed in an orthogonal control layer that handles most of the resource-management *minutiae* [15] while only higher-level functions are exposed through the VoiceXML markup language.

However, developers are still required to understand the underlying reactive nature of the media resource interaction [4]. For example, prompts in VoiceXML are queued and played only when the execution reaches an input state implicitly defined in the form execution. The semantics of VoiceXML markup interpretation hinge on

[1] http://www.w3c.org.

the controversial Form Interpretation Algorithm (FIA), which spans the dynamic behavior of reactive components (e.g., governed by multiple threads of execution and synchronized by event notifications) and implicitly defines an underlining state machine logic. Other approaches to speech services such as *XHTML+Voice* (X+V) [3] and Microsoft's neglected Speech Application Language Tags (SALT) [31] introduce multimodality into the equation so that, in addition to speech, input and output can be conveyed through a graphical user interface usually hosted by a web browser.

Thus, speech services design depends upon a monolithic architecture and closely coupled components that make it difficult to use the speech components in a context different from the traditional telephony execution environment. In the web application domain, the concept of loosely coupled and interoperable components, as found in many service-oriented architectures (SOA) [25], has been a successful programming paradigm for reliable and scalable service composition. Distributed resources are made available as independent services through simple communications protocols and API publication mechanisms (SOAP, WSDL, UDDI, etc.; see [2, Chap. 6] for a detailed description), greatly simplifying the creation of third-party applications by hiding the complexity of the technology in the network. Moreover, *mashups* or *web application hybrids* [32] have become a popular method for aggregating web services, allowing developers to invent new innovative applications by reusing public application programming interfaces (APIs) and RSS[2] data feeds.

These considerations suggest that techniques developed in the area of web services could be beneficial to speech services when using networked devices where speech is directly available in digital form. More specifically, can speech processing be seamlessly integrated into a web environment to leverage such broadly accepted programming paradigms and facilitate the creation of new multimodal services? Can speech processing resources be decoupled and made out of stateless building blocks available through the network with simple APIs? Today several architectures address these challenges by integrating speech and web services into one consistent network-hosted application framework. We categorize them in general as *Voice-as-a-Service* (VaaS) and consider this new trend as a second important milestone that significantly reduces the barrier to building telecommunication and speech processing services. VaaS allows developers to easily and rapidly develop new interactive applications for mobile smartphones, tablets, IPTV set-top boxes, web appliances, etc., without the need to install, configure, or manage speech processing software and equipment.

The rest of this chapter is organized as follows. Section 7.2 provides a brief overview of speech processing fundamentals. Section 7.3 describes telephony and speech mashups. Section 7.4 pinpoints issues related to the use of data streaming over the network, the network APIs, and the supported data formats. Section 7.5 shows the possible architecture configurations, while Sect. 7.6 describes the AT&T web portal used for speech resource management. Section 7.7 illustrates examples of mobile clients, and Sect. 7.8 shows some applications implemented with the AT&T speech mashup. Section 7.10 concludes and summarizes the chapter.

[2] http://www.rssboard.org/rss-specification.

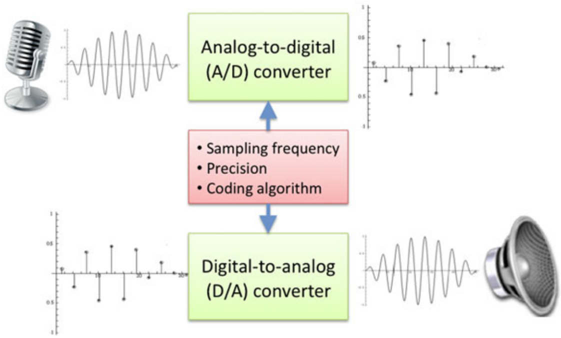



**Fig. 7.1** Analog-to-digital and digital-to-analog conversion

## 7.2 Elements of Speech Processing

This section is a brief overview of the main speech processing concepts necessary to better understand how to develop speech mashup applications. Although far from being a complete tutorial on such a vast topic, it provides pointers for further in-depth analysis.

### *7.2.1 Speech Coding*

In modern digital devices, speech and audio signals are represented as a sequence of numerical values, each number corresponding to the intensity of the signal at a specific instant in time. As shown in Fig. 7.1, analog signals are captured by microphones and sent to *analog-to-digital* (A/D) converters to transform them into sequences of numbers. This process is characterized by three parameters: (1) the sampling frequency—the number of samples per second generated by the converter (typical frequencies are 8,000 and 16,000 samples per second, e.g., 8 kHz and 16 kHz); (2) the numeric precision—the number of *bytes* used to represent each sample (for instance, 1 or 2 bytes); and (3) the coding algorithm—the algorithm used to compress the data stream to reduce the size of the data sample (examples of speech coding algorithms are G.711 μ-law with a data rate of 64 kbit/s, GSM 06.10 with 12.3 kbit/s, and MPEG audio layer III variable between 32 kbit/s and 320 kbit/s).

The reverse process that produces audio from digital samples is called *digital-to-analog* (D/A) conversion and it is done by making a speaker membrane vibrate with the same intensity and frequency of the original digital samples. When using this digital format, signals can be processed and transformed into other forms such a text representation or, conversely, synthetically created from text. Devices or algorithms capable of encoding and decoding signals are often referred as *codecs*.

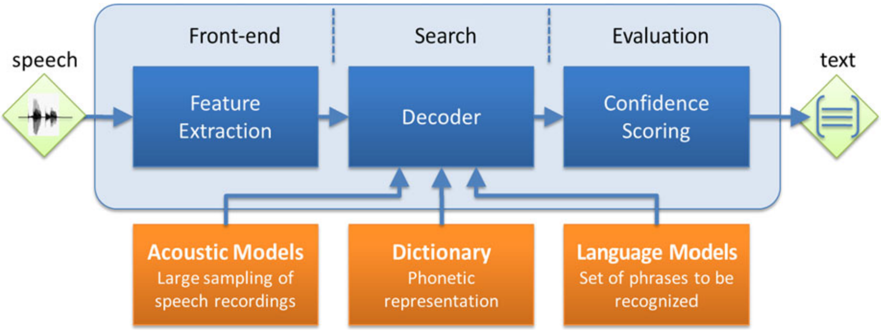



**Fig. 7.2** Automatic speech recognizer components

### *7.2.2 Speech Recognition*

Automatic speech recognition (ASR) converts speech to text. A number of ASR engines are available either as open source [23, 29] or proprietary commercial products (AT&T Speech API,[3] Nuance,[4] Microsoft Tellme[5]) but each ASR engine can be represented with the high-level architecture depicted in Fig. 7.2.

Four main steps are involved in translating speech to text: (1) the input speech in digital form is transformed into audio features describing energy levels at different frequencies; (2) audio features are matched to a large sampling library of speech recordings (*acoustic models*) and transformed into sequences of basic speech sounds called *phonemes*; (3) the resulting candidate phonemes from the previous step are matched and aligned to the sounds of predefined sets of words or phrases (*language models*); (4) the possible candidates are evaluated and selected by maximizing the probability for the best matching phrase.

Although there are only 40 distinct phonemes in English, the acoustic model tries to capture all the possible sound variations by storing thousands of different phonetic variations (*phones*) within the specific context of words (*tri-phones*, or phones-in-context) collected from many speakers and acoustic conditions. The next step consists of matching these phones to words or sentences. The set of all possible words is intractably large, so the search is typically limited to sentences most likely to occur in the context of a particular application. The language model (also called a *grammar*) contains the possible word or sentence combinations that the ASR will try to recognize. While the same acoustic model can be used for all applications of a given language, language models must be customized for each application. Word sequences not defined in the language model cannot be recognized.

There are two types of language models: rule-based and statistical.

[3] http://developer.att.com/API.

[4] http://nuance.com.

[5] http://microsoft.com/en-us/tellme/.

- *Rule-based grammars*—which explicitly define the set of sentences and the order of words that can be recognized. W3C recommendation specifies two standard formats: Augmented BNF (ABNF) and XML-based [24].
- *Statistical language models* (SLMs)—which do not use explicit rules, but instead use the statistical properties of thousands or even millions of transcribed utterances and text samples to help infer the words. The dataset itself (the corpus) does not need to explicitly contain every sentence that can possibly be recognized. Nonetheless, the SLM will have probability estimates for all word combinations of all words from the training text, so good vocabulary coverage in the training text is very important.

Finally, phonetic dictionaries model the pronunciation of words where each word present in the language model has one entry with one or more possible pronunciations (also called transcriptions). A pronunciation is a sequence of symbols from a phonetic alphabet. For instance, the word *tomato* can have two alternative pronunciations that, using the ARPA phonetic alphabet or ARPABet,[6] are typically both entered in the dictionary as possible pronunciations: *"t ow m ey t ow"* and *"t ah m ey t ow."*

In general, SLMs are trained with a large amount of text samples and used for open-ended tasks such as dictation of SMS messages, but they use more computational resources. When it is possible to constrain speakers to a small set of predefined responses (for instance, the names in an address book), rule-based grammars are preferred since they are faster and give a more predictable response.

### *7.2.3 Text-to-Speech Synthesis*

The task of text-to-speech synthesis is to convert text to speech, speech that is both *natural sounding* and *intelligible*. Several techniques have been developed for this purpose, but one of the more recent and successful approaches is *concatenative synthesis based on unit selection* [6, 7] illustrated in Fig. 7.3.

Modern concatenative synthesis involves three steps. The first step, common to almost all synthesizers, processes the text and converts it to a symbolic form. This is used for guiding the concatenation step that follows. Then there is a final signal processing step.

For the first step, input text is analyzed to detect the document structure and then normalized to identify abbreviations, acronyms, numbers, proper names, times, dates, special characters, and other elements that may require special processing. The text is also examined to extract syntactic or other text-based linguistic information. Once all such information has been identified, the collected information is transformed into a representation that specifies how the read version should sound.

[6] http://en.wikipedia.org/wiki/Arpabet.

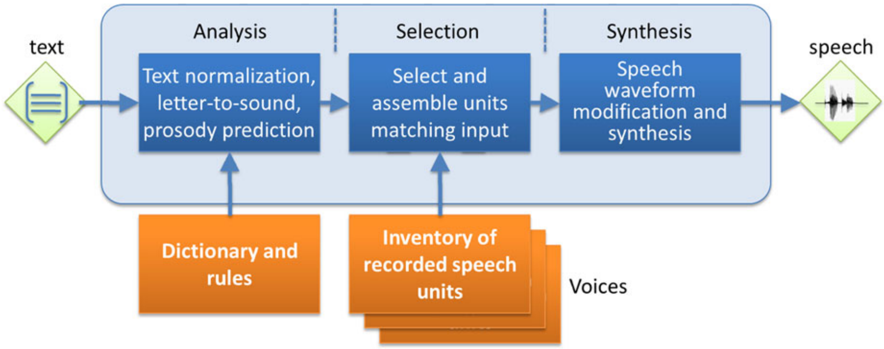



**Fig. 7.3** Concatenative text-to-speech synthesis based on unit selection

Phrases and pauses are identified, and words are mapped to their pronunciations using either a simple lookup dictionary or, for cases where no dictionary entry exists, a set of *letter-to-sound rules*. Finally the linguistic information and the phrasing and pronunciation information are used to generate an appropriate intonation or melody for the text (*prosody* prediction).

For the second step, the unit selection and concatenation step in Fig. 7.3 is one possible approach that relies on a very large data corpus of speech sounds to generate a potentially infinite number of utterances. The inventory of prerecorded and segmented speech units is typically recorded by professional speakers (but it can be generalized to any speaker, see Sect. 7.6.2). Speech units can be selected at different levels of granularity ranging from simple phones to words, or even full sentences, depending on the specific search algorithm chosen for the system. Candidate units are chosen from the speech database based on how well they match the specification derived in the first step. These candidate units are evaluated in the context of their possible neighbors, and an optimal list of units is selected for the next step. Natural quality and high intelligibility depend mostly on unit coverage. In other terms, the number of matching units selected that are closer to the targeted unit pitch, duration, and intensity.

Once the list of units is defined, the last phase is to extract the chosen audio from the database, and join units together as seamlessly as possible. This phase may also make use of signal processing to modify or improve the quality of the output. For large databases only minimal signal processing is required.

### *7.2.4 Configuring ASR and TTS*

Although speech recognizers and synthesizers are fairly complex and controlled by a number of configuration parameters, typical cloud-based deployments used for speech mashups are preconfigured with reasonable default parameter values. Depending on the application, configuring a speech recognizer may require defining acoustic models and language models, adjusting several engine parameters, and even adding pronunciation dictionaries. Additionally, a developer should select a suitable rejection threshold that discards results with low confidence. Configuring a synthesizer, on the other hand, may only require selecting the type of speech (e.g., the speech unit inventory) and eventual extra pronunciations to address anomalous letter-to-sound conversions.

## 7.3 Voice-as-a-Service

Voice-as-a-Service (VaaS) generally refers to telecommunication and speech processing services available in the cloud that may include basic telecommunication functions—placing and receiving phone calls, conferencing, touch-tone detection and generation, SMS and IM messaging, etc.—and speech processing capabilities as illustrated in the previous section. VaaS can be roughly categorized into two types: *telephony mashups* (also called telecommunication or telco mashups) and *speech mashups* (sometimes referred as *voice mashups*). Both categories involve the capability to stream speech *to* and *from* endpoint devices that are able to play back and capture speech either in analog or digital form. The main differences are related to *where* the speech streaming takes place and the type of functionality exposed to developers.

### *7.3.1 Telephony Mashups*

As shown in Fig. 7.4, telephony mashups [11, 18] confine speech processing activities in the traditional public-switched telephone network (PSTN) or, in some cases, through devices using Voice-over-IP-enabled (VoIP), where speech is first converted into digital format and then delivered across the Internet by a VoIP gateway. In both cases, the overall interaction experience is similar to traditional telephony VoiceXML systems mentioned in Sect. 7.1.

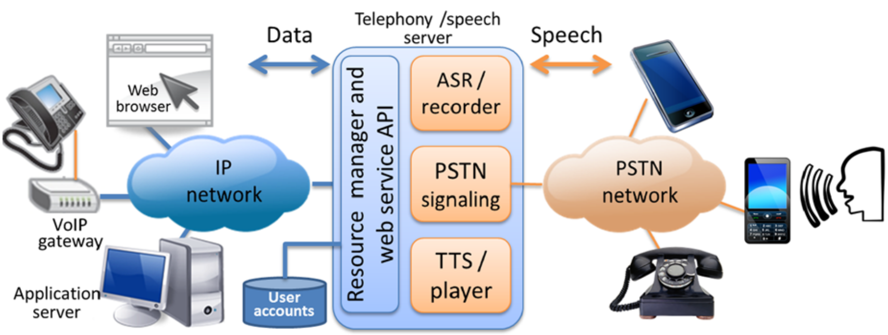



**Fig. 7.4** Telephony mashup architecture

The speech stream is never exposed directly to the developer's API, and speech processing takes place between phones and telephony servers. The web-service interface provides access to high-level functionalities required for basic call-control capabilities including placing, receiving, conferencing, transferring phone calls, and, when available, speech recognition and synthesis. A telephony mashup application typically resides on a web server (application server), and the telephony server is responsible for linking the main application URL to a unique telephony resource identifier (e.g., a PSTN number or a VoIP SIP address). An interaction with a telephony mashup application may involve the following steps:

1. Placing a phone call to the application phone number triggers an HTTP GET or POST request to the application URL specified in the application configuration stored in the user's account database.
2. The telephony server passes parameters into the body of the POST request or as an argument of the GET request and retrieves the response from the web server.
3. The response includes a set of commands appropriate for the telephony platform. This may include playing a prompt and requesting an input from the user.
4. The commands are executed by the telephony server, and the result of the execution is sent to application servers as a new request. The process goes next to the previous step and iterates till the user hangs up or the application decides to terminate the phone call.

Some examples of currently available telephony mashups are described in the following sections.

#### 7.3.1.1 Tropo

Tropo[7] is an open-source telephony mashup platform based on Voxeo[8] cloud communication API. It provides free developer sandbox access to create and test applications with a rich-set telephony and speech API supported by a variety of scripting languages (Ruby, Node.js, PHP, Python) and REST API. When developers register an account, Tropo allows them to associate each application to a regular PSTN number, a SIP address, a Skype ID, or a browser-based phone address (Phono[9]) running on a desktop.

The PHP code snippet in Fig. 7.5 shows a simple example of a telephony mashup that returns the local weather forecast by querying the Yahoo! weather API. After answering the call, Tropo sequentially executes the code in Fig. 7.5. First, the function `ask` plays the welcome prompt and asks for a zip code using a female TTS voice. Then, the caller can either say the zip code or enter the five-digit number using the telephone keypad. Once the input is entered, the `onChoice` callback function,

[7] http://tropo.com.

[8] http://voxeo.com.

[9] http://phono.com.


**Fig. 7.5** Example snippet of Tropo telephony mashup code using rssWeather.com and Yahoo! Pipes tool

assigned to `sayWeather`, is executed, and the location identified by the zip code is sent to a Yahoo! Pipes[10] application graphically pictured on the right. The pipes application calls the rssWeather.com weather API to retrieve an XML document with the forecast information for that area. `getWeather` parses and extracts the information necessary, returning as a result a text document that is synthesized back to the caller by the TTS (for instance: *The current weather condition in Florham Park, NJ, is "mostly cloudy", with a temperature of 66 degrees Fahrenheit and a humidity of 94 %. Tomorrow, the weather-forecast expects: "chance of storm" with a temperature ranging from 63 to 82 degrees. Goodbye.*). Note that the high-level abstraction of the API hides the complexity of the interaction from the developer, although a finer-grain control can be achieved by overriding default parameters (e.g., taking specific action when no input is provided by using `onTimeout` callback, changing the TTS voice to British English by setting `'voice' => 'Elizabeth'`, or specifying a full XML-based grammar for speech recognition).

#### 7.3.1.2 Other Telephony Mashup Services

Twilio[11] is a telephony mashup platform that runs on the Amazon Elastic Compute Cloud (Amazon EC2[12]). In terms of capabilities, Twilio is similar to Tropo but does not support speech recognition. Applications can be implemented either by a REST API with libraries supported in PHP, C#, and Python, or with a proprietary XML markup language called TwiML.

[10]http://pipes.yahoo.com.

[11]http://twilio.com.

[12]http://aws.amazon.com/ec2/.

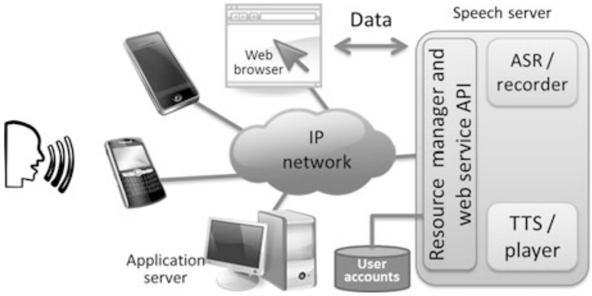



**Fig. 7.6** Speech mashup architecture

Plivo[13] is an open-source cloud-based telephony mashup built around FreeSWITCH,[14] a cross-platform open-source telephony platform. Developers can implement their applications with the help of libraries (Java, PHP, Python, and Ruby) or directly using REST APIs and Plivo RESTXML markup language. Similarly, Teleku[15] with PhoneML and CallFire[16] with CallFire XML.

Other interesting approaches described in [8, 26, 27] try to reuse the existing VoiceXML infrastructure and expand it with multimodal interaction based on a more traditional telephony model.

### *7.3.2 Speech Mashups*

Speech mashups are web services that implement speech technologies, including automatic speech recognition and text-to-speech synthesis, for web applications. They enable users of an application to use voice commands to make requests or to convert text to audio. They also allow multimodal inputs (such as gestures and taps) to be conveyed as well. Speech mashups work by relaying audio or text from an application running on a mobile device, a web browser, or any media and network-enabled device to servers in the network where the appropriate conversion takes place. The result of the process, either text or audio, is returned to the client application.

The concept behind the speech mashup technology is intuitive and similar to the familiar web application approach. Communication between devices and the service platform is established over the packet-switched network; no traditional PSTN circuit switching sessions take place. As pictured in Fig. 7.6, in a typical speech recognition interaction, the speech is first captured on a communication device—*the client*—through the microphone and compressed using one of the available speech codecs. A typical speech compression method used for mobile phones is the

[13] http://plivo.org.

[14] http://freeswitch.org.

[15] http://teleku.com.

[16] http://callfire.com.

Adaptive Multi-Rate[17] codec that has good speech quality with only a 12.2 kbit/s transmission rate. Then an HTTP (Hypertext Transfer Protocol) [5] connection—*the transport*—is established with the speech mashup resource manager—*the server*—which delivers the bit stream to a speech recognizer engine along with a set of parameters including the reference to the grammar or language model used to recognize the utterance. The recognition results are then returned back to the client and used by the client to take the next action. Depending on the complexity of the task, a further processing stage that aims to identify the meaning of the user's request could be added to the results, so that the natural language variation of the same user's intent can be interpreted properly. Optionally, an application or web mashup server could provide additional service logic that is not strictly related to speech processing.

To illustrate these concepts, we will mostly refer to the AT&T speech mashup implementation, but the same principles apply to other implementations with minor changes.

#### 7.3.2.1 AT&T Speech Mashup

The AT&T speech mashup platform [12, 14, 22], is a cloud-based service first deployed in January 2008 and available for non-commercial use. The general architecture is similar to the one depicted in Fig. 7.6. The resource manager or speech mashup manager (SMM) layer implements a multi-user, multi-application hosting platform with two main functions: (1) run-time resource allocation management, (2) off-line user account management. In the former case, it makes the AT&T WATSON[SM] speech recognition engine [19] and the AT&T Natural Voices™ text-to-speech synthesis engine [6] accessible through any network as web services and exposes them through an HTTP-based API. In the latter function, the SMM provides web interfaces to upload and compile users' grammars, and to create new synthetic voices. The SMM also provides data loggers to trace the service activities, and tools for utterance transcription.

From the architecture perspective, a speech mashup application consists of three main components:

1. The AT&T speech mashup server, including the speech mashup manager (SMM), the WATSON servers for ASR, and the Natural Voices servers for TTS. The SMM opens and manages direct connections to the appropriate AT&T speech servers on behalf of the client, including resolving device-dependency issues and performing authentication and general accounting.
2. A speech mashup client that relays audio to the AT&T speech mashup servers and accepts the recognition result, and/or a client that receives audio buffers and plays them through the audio interface. Examples of speech mashup clients are

[17] http://en.wikipedia.org/wiki/Adaptive_multi-rate_compression.

available for Java ME[18] devices, the iPhone iOS, Android devices, Linux-based desktops, and web browsers through Java Applet clients.
3. A main application server (e.g., Apache or Tomcat) that provides access to the application's back-end database, performs data aggregation processing with other application servers, and, depending on the adopted mashup configuration, can implement the application logic.

The SMM component is implemented as a Java servlet with a relational database back-end, while the other components are native Linux processes. No special hardware is required; all processes run on Linux servers and have also been tested in the Amazon EC2 cloud for large-scale deployments.

Figure 7.7 shows an example of weather forecast speech mashups analogous to the telephony mashup example described in Sect. 7.3.1.1. In this case, the client is MTALK [22], a multimodal browser running on iPhone, where, in addition to the standard web browser functionalities, a JavaScript API implements the communication with the speech mashup server. In the first screenshot, the user taps on the *"Speak"* button to start the speech interaction. The API call `talk.asr('citystate', ...)` is executed, and the speech starts to be captured and streamed to the server. The recognizer receives a set of configuration parameters, including the *citystate* grammar, and begins processing the speech samples. The user says a U.S. location identified by city and state names: *"Florham Park, New Jersey."* In the second snapshot, the recognizer returns the result that is parsed and used to populate the fields in the form. Finally, the result triggers an API call to the weather service implemented with Yahoo! Pipes (right panel in Fig. 7.7). The Yahoo! Pipes returns the resulting forecast in XML and visualizes it in a suitable mobile format in the third panel of Fig. 7.7.

#### 7.3.2.2 Other Speech Mashups

A similar speech mashup approach is the WAMI (web-accessible multimodal applications) toolkit [20]. WAMI is an open-source initiative that proposes a general framework for multimodal applications. It is based on MIT speech processing technology encapsulated around a web server and a number of clients including web browsers and smartphones.

Other commercial initiatives include Nuance's Dragon Mobile SDK[19] for ASR and TTS on mobile, Google Speech API for ASR and TTS on Android devices, Nexiwave[20] for off-line speech recognition and Acapela[21] for multilingual text-to-speech synthesis.

[18] Java Micro Edition—http://www.oracle.com/technetwork/java/javame.

[19] http://dragonmobile.nuancemobiledeveloper.com.

[20] http://nexiwave.com.

[21] http://www.acapela-group.com.

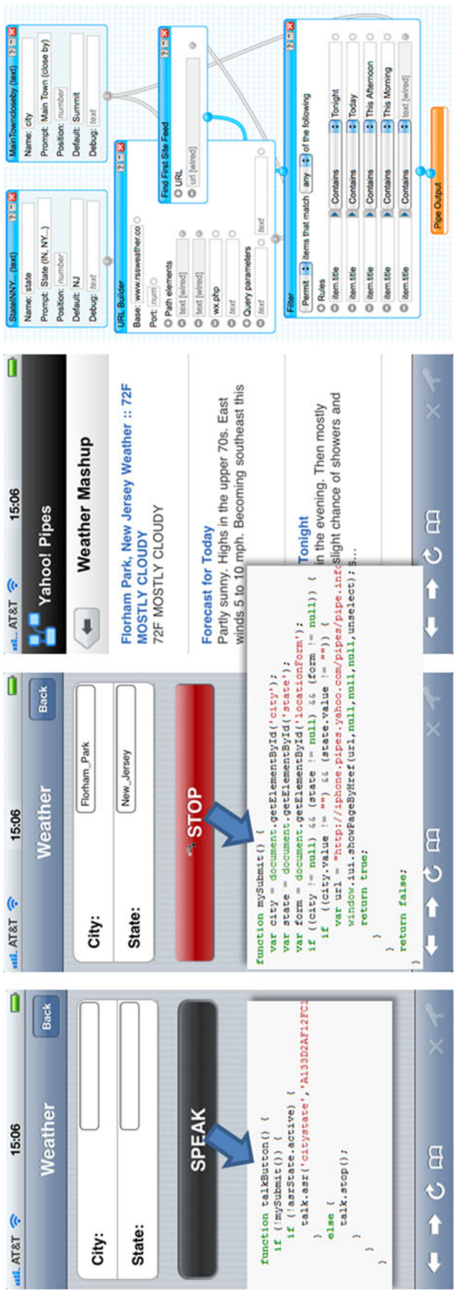



**Fig. 7.7** Example of weather information speech mashups using rssWeather.com and Yahoo! Pipes tool

## 7.4 Network Transport

Streaming continuous media over IP networks is a difficult problem due to the real-time nature of the media generated by client devices. Moreover, network latencies introduced by routers and gateways between devices and switched networks complicate communication even further.

### *7.4.1 Speech over the Data Network*

A variety of protocols are available for real-time media transmission over IP networks. The media resource control protocol version 2 (MRCP v2) [30], for example, is an IETF recommendation for speech-server communication that relies on real-time transport protocol (RTP) and session initiation protocol (SIP). While MRCPcv2 has been designed to deliver real-time media over IP, it was not expressly designed for the wireless network. Moreover, the implementation complexity of both client and server side might be overkill for speech mashup applications.

In case of mobile networks, bandwidth is a scarce resource. The wireless third generation (3G) network, for instance, specifies a maximum data rate of 2.4 Mbit/s on the download link and 153.6 kbit/s on the upload link shared by all users within a single cell sector. These are usually ideal bandwidth conditions that tend to degrade due to environmental conditions such as fading and interference or other conditions like the device travel speed and the density of devices per cell. To deliver media over wireless networks, it is important that mobile clients be carefully designed to reduce latency by compressing media and reducing the number of retransmissions.

Based on the previous considerations, HTTP [5] is one of the most interesting options as transport protocol for speech mashups. HTTP had been originally designed to support HTML, but with the recent version 1.1 extensions, it is often used as a main choice for media protocols on the web. HTTP 1.1 supports persistent connections, allowing sockets to be reused over lengthy transmissions, eliminating the overhead of creating new connections for every transaction. HTTP 1.1 also implements *chunked encoding*, which permits HTTP messages to be broken into several parts. This is very important for delivering and receiving speech segments as they are produced or consumed by the wireless client. Although real-time streaming is not guaranteed, message chunking and persistent connections greatly reduce latency, achieving real-time performance in the majority of working conditions. In essence, HTTP, although not strictly designed for real-time media streaming, turns out to be a practical trade-off between simplicity and efficiency.

### *7.4.2 REST-Based APIs*

Speech mashup APIs follow the Representational State Transfer (REST) [17] network architecture principles. REST refers to resources in the network as uniquely

**Table 7.1** Examples of ASR REST APIs

| Parameter | Value | Description |
|---|---|---|
| uuid | string | Required. Unique user ID assigned at registration. |
| resultFormat | string | Optional. Result format, which can be EMMA, JSON, XML. |
| appname | string | Required. Application namespace. |
| cmd | string | Required. One of the following command strings: |
| | | oneshot—Starts ASR in stateless mode; the request body will contain the entire audio stream. |
| | | start—Starts ASR in stateful mode; the audio stream will be sent using one or more audio or rawaudio requests. |
| | | stop—Stops stateful ASR and returns the result. |
| | | audio—Sends a chunk of audio for stateful ASR. |
| audioFormat | string | Optional. Audio data format supplied by the client. Possible values are: |
| | | amr—Adaptive Multi-Rate (AMR), narrow-band only |
| | | au—Sun AU, μ-law or 16-bit linear |
| | | caf—Apple Core Audio Format, μ-law or linear |
| | | wav—Microsoft/IBM Wave, μ-law or linear |
| | | ... |

identified by a global identifier (e.g., a URI or Uniform Resource Identifier on HTTP). For example, for the AT&T speech mashup, `http://service.research.att.com/smm/asr?uuid=[uuid]&grammar=citystate&format=emma` is a request directed to the speech recognizer resource (`/smm/asr`) with the grammar `citystate` by the registered user identified by the unique user ID (`uuid`). Requests for resource manipulation are sent to components in the network that are responsible for responding to the request with a *representation* of the request, for instance an XML document containing the result of the speech recognition process following the Extensible MultiModal Annotation (EMMA) markup language [21] standard and declared as `format=emma` in the previous example. In the case of a speech recognition interaction, a RESTful call is carried out by the client over an HTTP POST request where the API parameters are in the HTTP header and the speech data attached to the HTTP body. Multiple chunks of data are fleshed out in a sequence of HTTP POSTs by the client as the speech is captured by the microphone. When the utterance is completed, the server generates a response in the specified output format. Table 7.1 shows an example of APIs for the ASR resource.

In the simplest form, a client request can be carried out by the traditional Unix `wget` command (Fig. 7.8) where the input speech is stored in a file (`sample-speech.amr`) and the result written to an EMMA document in a text file (`response.emma`).

Conversely, a TTS interaction starts by posting the text to synthesize and the server responds with a chunked data stream over a persistent data connection. TTS

```
wget \
 --post-file=sample-speech.amr \
 --header 'Content-Type: audio/amr' \
 --server-response 'http://service.research.att.com/smm/asr? \
         grammar=citystate&uuid=[uuid]&resultFormat=emma' \
 -O response.emma
```

**Fig. 7.8** Example of `wget` speech recognition client request

**Table 7.2** Examples of TTS REST APIs

| Parameter | Value | Description |
|---|---|---|
| uuid | string | Required. Unique user ID assigned at registration. |
| text | string | Optional. Text may also be supplied in the body of a POST request. |
| audioFormat | string | Optional. This specifies the format of the audio data supplied by the client. Possible values are: |
| | | amr—Adaptive Multi-Rate (AMR), narrow-band only |
| | | mulaw—AU with μ-law encoding |
| | | alaw—AU with A-law encoding |
| | | linear—AU, 16-bit linear |
| voice | string | Optional. Crystal (default) or Mike. |
| sampleRate | integer | Optional. The audio data sample rate. Defaults to 8,000 Hz. |
| ssml | true/false | Optional. Set this parameter to true when text contains SSML tags. (When set to the default, false, each word is pronounced, including SSML tags.) |
| | | ... |

bookmarks can be interleaved with the speech chunks as well. The client has the responsibility for playing the speech with the proper sample rate and for dispatching the bookmarks at the proper time. Table 7.2 shows the REST API for the TTS resource.

### *7.4.3 Supported Data Formats*

In the AT&T case, SMM returns or accepts structured data related to the execution of a specific speech processing task to or from the client. For an ASR task, speech recognition results and detailed information about the parameter values and settings used during the recognition are described in different equivalent formats that can be selected based on the client capabilities or preferences. There are three standard output formats: XML, JavaScript Object Notation (JSON) [13], and EMMA markup language [21]. JSON is a data serialization format that maps directly into JavaScript objects and it is particularly convenient in a browser-based client environment where a JavaScript interpreter is natively available. EMMA is a richer notation recommended by W3C for interoperable input format representation for multimodal systems. EMMA facilitates plug-and-play of systems components and

```
<emma:emma version="1.0">
<emma:grammar id="gram1" ref="smm:grammar=ipizza&amp;UUID=[uuid]"/>
<emma:model id="model1" ref="smm:file=pizzahut.xsd&amp;UUID=[uuid]"/>
<emma:info>
  <session_id>33C07738-DA61-4814-B60D-4D374F143D8D</session_id>
</emma:info>
<emma:one-of id="one-of1" emma:medium="acoustic"
 emma:mode="voice" emma:function="dialog"
 emma:verbal="true" emma:lang="en-US"
 emma:start="1246437000" emma:end="1246437008"
 emma:grammar-ref="gram1"
 emma:signal="smm:UUID=[uuid]&amp;file=audio-454907.amr"
 emma:signal-size="8070"
 emma:media-type="audio/amr; rate=8000"
 emma:source="smm:platform=null&amp;device_id=null"
 emma:process="smm:type=asr&amp;version=watson-6.3.0000"
 emma:duration="6150"
 emma:model-ref="model1"
 emma:dialog-turn="33C07738-DA61-4814-B60D-4D374F143D8D:1">
<emma:interpretation id="nbest1" emma:confidence="1.0"
 emma:tokens="a large pizza with pepperoni sausage two
Diet Pepsis and a root beer">
 <item>a
        <size_LG>large</size_LG>
        <type_own>pizza with
                <topm_MPE>pepperoni </topm_MPE>
                <topm_MIS>sausage</topm_MIS>
        </type_own>
 </item>
 <item>
        <qt_2>two</qt_2>
        <drinks_DP>Diet Pepsis</drinks_DP>
 </item>and
 <item>
        <qt_1>a</qt_1>
        <drinks_RB>root beer</drinks_RB>
 </item>
</emma:interpretation>
</emma:one-of>
</emma:emma>
```

**Fig. 7.9** Sample EMMA document

is suitable for annotating various stages of the processing of the user's input. An example of an EMMA document is shown in Fig. 7.9. In this case, the ASR recognized the utterance *"a large pizza with pepperoni sausage two Diet Pepsis and a root beer"* which is captured in the element `emma:interpretation` both as literal text (i.e., attribute `emma:tokens`) and a semantically tagged interpretation (i.e., body of the element `interpretation` with semantic XML markups).

For a TTS task, the input text follows the W3C Speech Synthesis Markup Language (SSML) [9] standard. SSML is an XML markup language for modifying the way text is processed by TTS engines. The SSML tags are instructions for normalizing text and controlling emphasis and other speaking qualities (prosody). For example, the attribute `type = telephone` treats the text as a telephone number so that the SSML fragment `<say-as type = "telephone"> 9081234567 </say-as>` is synthesized as *"nine zero eight [pause] one two three [pause] four five six eight."*

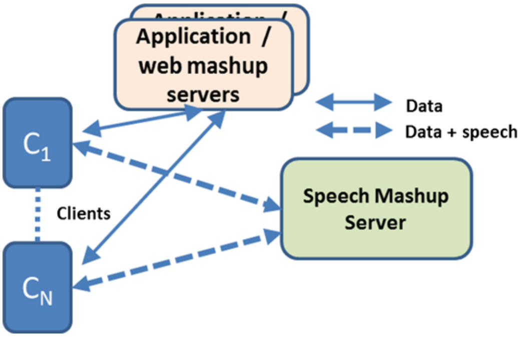



**Fig. 7.10** Client-side speech mashup

## 7.5 Speech Mashup Configurations

Speech mashups can be configured in different ways depending on the application complexity, the wireless network latency, and the level of client authentication required. Figure 7.10 shows a typical *client-side* configuration where the $N$ clients $(C_1, \ldots, C_N)$ implement the logic of the application and are responsible for the communication between the speech mashup server (SMS) and the application/web mashup servers (AS). All the network connections are originated by the mobile clients in the wireless packet-switched network (typically a GPRS[22] or an EDGE[23] network). Speech data requests (dotted arrows) are directed to the SMS component and text data requests (solid arrows), such as database queries, are sent to the AS modules. Each service is identified by a URL and, as described in Sect. 7.4.2, HTTP is the communication link between the client and the speech mashup server.

If the wireless network is congested, creating an HTTP link could take a significant amount of time, limiting the use of this configuration to simpler applications where the number of client round trips is small or a faster wireless network (WiFi) is available. Also, device authentication, when required, is delegated to the SMM component, which may involve some application-dependent authentication model directly implemented in the speech mashup server.

Figure 7.11 illustrates a *server-side* configuration. Here the mobile devices are connecting directly to the main application server where the content aggregation with other content servers takes place. In this case, the clients are facing one server with a single URL, allowing the application to reuse the connections for multiple interactions, saving on the cost of creating multiple connections. The main application server can use the faster IP network to aggregate the data and format the output for specific device display capabilities, saving further round trips and processing time on the device. The disadvantage in this configuration is that the speech chunks delivered over HTTP by the client (or produced by the TTS from SMM) have to be relayed to the SMM component (or transferred from the SMM to the client) through the application server. The application server has to implement the HTTP chunking

[22]General Packet Radio Service.

[23]Enhanced Data Rates for GSM Evolution.

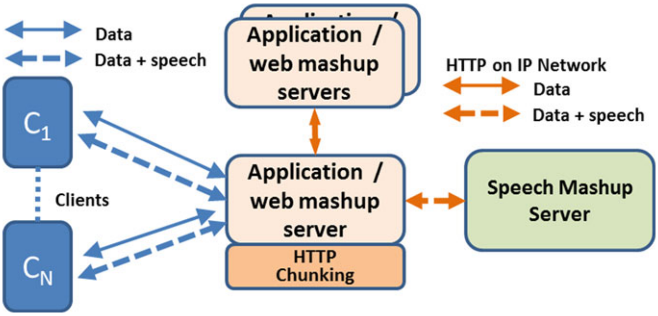


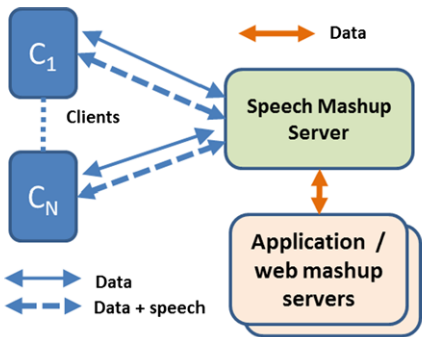

**Fig. 7.11** Server-side speech mashup

streaming module to pass the speech to the SMM (or vice versa for the TTS) in a timely manner.

Finally, Fig. 7.12 sketches the *back-to-back* configuration. As previously mentioned, multiple handset client connections (i.e., HTTP sessions) are significantly more expensive than one connection to accomplish a given task. Passing a large amount of data from a server to a handset for the purpose of forwarding that data to another server (server-side configuration) is far more expensive than having an intermediate server act as a data broker. To address these concerns, SMM allows a single incoming HTTP POST request from a client to specify a preprocessing URL, a speech action (such as a recognition), and a post-processing URL. The preprocessing and post-processing URLs are optional. When the SMM receives a request, it first makes an HTTP POST request to the preprocessing URL, passing the body of incoming request as the body of the POST.

SMM performs the speech action using the response from the preprocessing URL. Then it makes a POST request to the post-processing URL, passing the result of the speech action as the body of the POST. Finally, it returns the response from this POST to the client as the HTTP response to the client's initial request. This is equivalent to *piping* requests from the wireless network to the IP network and minimizes the number of round-trips over the wireless network.

**Fig. 7.12** Back-to-back speech mashup



## 7.6 Speech Mashup Management

In order to centralize the speech processing resources into the network, a web-based portal is usually necessary to manage user accounts, ASR grammars, TTS voices, disk space quota management, load balancing across ASR and TTS servers, and logging. Typically, once the speech mashup user registers to the portal, a unique account ID is emailed to the user, a grammar compilation area is created and the user can immediately start using the ASR by referring to the default built-in grammars and the TTS by selecting the available voices. The next sections refer to the AT&T speech mashup portal.

### *7.6.1 ASR Management*

The AT&T portal supports any language model that can be represented as a weighted finite state machine (FSM), and compiles both rule-based and stochastic grammars into FSMs. In general, rule-based grammars are used for lower complexity tasks and when training data are not available or are scarce. Examples of rule-based grammars included in the portal are *dates*, *times*, *confirmation*, *phone numbers*, and *U.S. city/states*. Rule-based grammars can be represented either in SRGS (Speech Recognition Grammar Specification) [24] format or in a more compact proprietary representation similar to the Backus–Naur Form notation. Corpus-based stochastic language models are used for more complex tasks such as customer care call routing or voice search and are compiled into a weighted FSM with Katz back-off smoothing [1] (other smoothing methods are also available). A typical natural language customer care task requires an average of 5,000 transcribed utterance samples to reach operational performance level ($>$ 75 % word accuracy).

Each grammar can be assigned to a different user's account scope: (1) *My grammars*—a user's own grammars that are not accessible to anyone else; (2) *Shared*

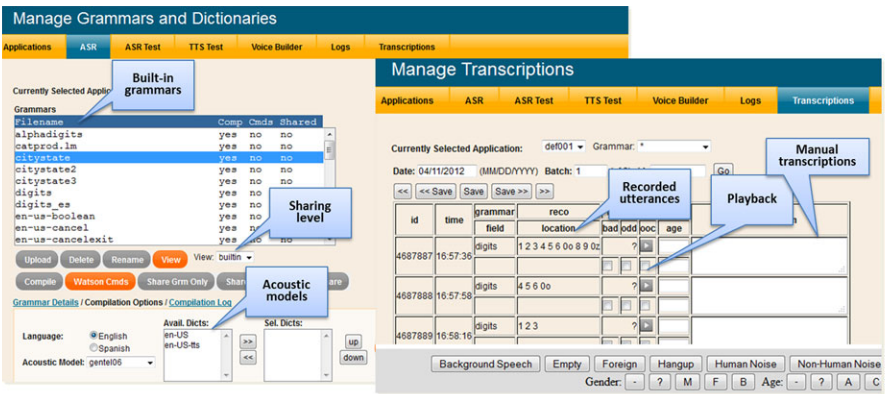



**Fig. 7.13** AT&T speech mashup portal grammar manager and utterance transcription screenshots

*grammars*—grammars created by others and then made available to all speech mashup users; (3) *Built-in grammars*—general-use grammars shared for all the accounts (e.g., SMS dictation, web search, U.S. cities and states, digits, etc.). The portal also exposes logging files for debugging and tracing purposes and a web-based transcription tool (Fig. 7.13) to listen and transcribe the recorded utterances in real time. Utterance transcriptions can be periodically downloaded and used as training data to improve the speech recognition performance.

### *7.6.2 Text-to-Speech Management*

Building a new voice for TTS has historically been a very labor-intensive process, segmenting the voice actor's audio recordings into phonemes manually, one by one, requiring specialized software tools and skilled technicians. With high-quality automatic speech recognition (ASR), however, much of this process can be automated, as the speech recognition engine can identify the starting and ending points of each recognized phoneme, thus making it possible to perform the segmentation task mechanically.

The VoiceBuilder for AT&T Natural Voices TTS is just such an automated system. With it, anyone with access to a basic computer with a microphone can record sentences and obtain a voice suitable for automatic text-to-speech.

Figure 7.14 shows the interface that prompts users to speak a sentence; the user clicks *"Record"* to start recording an utterance and users click it again to end the recording; the system will respond by showing the ASR confidence score. The users may attempt to record a sentence as many times as is necessary to obtain a high

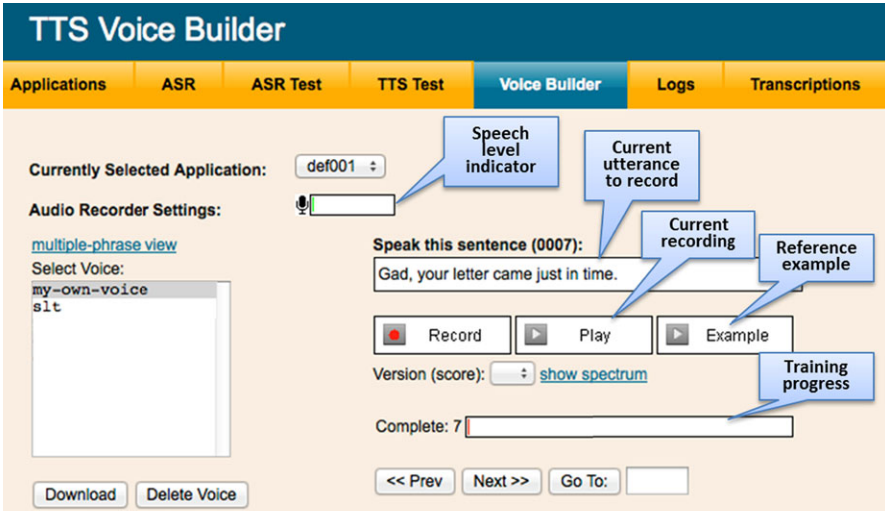



**Fig. 7.14** Example of TTS voice creation interface taken from the AT&T VoiceBuilder web page

score, and they can click *"Example"* to hear the sentence spoken by a professional voice actor in case they are unsure about pronunciation.

In addition to recording using the web interface, recordings made off-line can be uploaded all at once (and scored in batch mode) by using the *"Upload Audio"* section at the bottom of the screen.

When the user clicks *"Build Voice"*, the system will present a histogram of confidence scores for all recorded sentences; if the number of high scores is sufficient, they may then proceed to start the actual voice-building process. This process may take a long time, up to several hours (depending on the number of sentences that were recorded). It is not necessary to remain on-line while this happens; the VoiceBuilder will continue running even after the user logs off. The user will receive an email confirmation once the VoiceBuilder has finished, and if the voice is successfully built, the new voice will then be available in the user's account, and for use with the `/smm/tts` web service.

## 7.7 Mobile Clients

A speech mashup client (Fig. 7.15) is the part of a speech-enabled application that runs on the user's mobile device or, in principle, on any voice-enabled and networked device. Its role is to capture and relay speech or text to the speech mashup manager, which in turn handles authentication, accounting, and communication with the AT&T speech servers. The AT&T speech mashup portal website provides the following client examples downloadable in source code through the sample code link: (1) a Java ME client that can be used for most Java-enabled mobile devices; (2) a native client application for the iPhone and Android; (3) Java applets for any web browser.

The client code samples include three main modules: (1) an audio manager to capture real-time speech from the device microphone; (2) a REST client that implements HTTP and the speech mashup server API; (3) a simple graphical user interface (GUI manager) that enables a 'Speak' button on the device (either an actual button on the mobile keypad or a touch screen soft button). After the user hits the speak

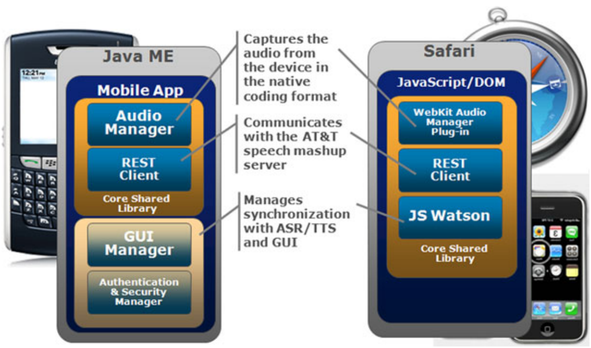



**Fig. 7.15** Typical speech mashup mobile client architecture for Java ME and iOS browsers

button, the audio-capture process starts and the speech samples are streamed over HTTP to the ASR through the speech mashup server. When the button is released, the mashup returns an XML or JSON document containing the speech recognition results, which are displayed on the device screen. In general, an interaction manager would be responsible for taking the next action to present the result to the user.

## 7.8 Sample Applications

This section illustrates three multimodal mobile examples implementing the three architecture configurations described in Sect. 7.5. Input modalities are either speech or typed text. The first is a native iPhone application for local business search freely available in the Apple Store. The second is a prototype iPhone service for pizza ordering. The last example is a version of mobile local business search for BlackBerry phones.

### *7.8.1 Speak4it*

Speak4it[24] [16] (Fig. 7.16) is a native iPhone application for local business search for accessing around 20M entries from the yp.com search engine. It is based on the *server-side* configuration, where the application server combines the speech requests with a local business search engine and a map location information server. The user inputs a request by holding the Push & Talk button and uttering a natural language query such as *"Find Japanese restaurants in Glendale California."* Voice search queries can be either by business name or category with a natural language understanding module extracting the information needed to populate the query fields for the search engine. If the location is omitted, the iPhone location system information is used as default coordinates. The result of the search is displayed in the form of a listing or as a scrollable map. The user can call any item in the list by tapping on the displayed phone numbers.

### *7.8.2 iPizza*

iPizza (Fig. 7.17) implements a multimodal interface for on-line pizza ordering based on the *client-side* configuration. It runs as a native application on an iPhone and combines speech and touch inputs with graphical interaction to enable users to rapidly select and edit menu items. Users can speak naturally by tapping on the 'Talk' button and request multiple items at the same time. For example: *"I'd like*

[24] http://www.speak4it.com.


**Fig. 7.16** Speak4it—multimodal local business search


**Fig. 7.17** iPizza—multimodal pizza ordering prototype—initial screen, shopping cart example, and ordered item editing

*to order a pizza with mushrooms and ham, two Diet Pepsis and baked cinnamon sticks.*" The graphical interface allows easy navigation to update the items in the shopping cart by mixing voice and touch inputs at any stage of the ordering process.

Items listed in the shopping cart (Fig. 7.17) can be edited by tapping on a specific item to edit or by saying the new value of a field by using the talk button.

### *7.8.3 JME Local Business Search*

The JME local business search (Fig. 7.18) is similar to Speak4it but implemented as a Java ME application and it runs on the BlackBerry family of smartphones. The user formulates the voice search query by holding down the keypad call button for the duration of the utterance. This example exercises the *back-to-back* configuration


**Fig. 7.18** Multimodal local business search prototype on BlackBerry phone

where most of the application logic is implemented as a web application, but only text messages are exchanged across. In this configuration, the speech recognition results are first analyzed by a natural language understanding component that identifies the relevant parts of the user's query (e.g., business name or category and location). Then, the annotated result is delivered by the speech mashup manager to a post-processing URL that identifies the web application responsible for handling the user's input. The web application queries the local business search engine and a map location service. Then it combines the resulting business listing and the mapping information into a format suitable for the visualization capabilities of the device and returns the response to the client though the same HTTP request. In this configuration, the web application does not handle the speech stream; it is invoked only after the speech processing stage is completed, providing a significantly simpler and more tradition web application framework.

## 7.9 Emerging Standards

This section covers some currently emerging technologies and standards, and discusses how they may be used to expand the capabilities of speech mashup platforms.

### *7.9.1 HTML5 and Speech Interaction*

The current HTML standard specification, HTML 4.01, offers very little support for speech-enabled applications. Features not covered by the standard are usually addressed by plug-ins, which are third-party code modules loaded by web browsers at run time. For speech-enabled applications, the non-standard functionality that is required is audio recording and fine-grained control over audio play-back, and this functionality may be provided by several third-party plug-ins such as Adobe Flash, Microsoft Silverlight, and Oracle Java.

Browser plug-ins rely on the availability of software development kits (SDKs) to augment the browser functionalities with natively compiled code that exposes lower-level components not normally accessible through the JavaScript interface.

This situation, while workable for most desktop computing platforms, has significant drawbacks. First, in many cases appropriate browser plug-ins are not available, e.g., Microsoft Silverlight is not supported for all browsers in Mac OS X, and not at all on Linux or iOS; Adobe Flash is not supported on iOS, and the Java plug-in is currently not supported on any smartphone platform. Second, even if a platform is supported, many users will not be able to install plug-ins because of security policies or lack of technical knowledge.

The draft HTML5 specification[25] addresses some of these issues. It supports an `<audio>` element that allows play-back and control of audio files without the need for browser plug-ins. This element can be used to create text-to-speech applications that will work on any HTML5-compliant browser. HTML5 does provide support for audio recording, which is of course needed for speech recognition (ASR). There is a proposal sponsored by Google for speech-enabling HTML text fields by adding a "speech" attribute,[26] but currently this is only implemented in Google Chrome, and the feature is also tied to Google's back-end ASR service. In order for this approach to appeal to web application developers, it will need to be supported by more browser vendors, and application developers should be able to use alternative ASR service providers.

A more comprehensive proposal has been drafted by the W3C HTML Speech Incubator Group.[27] The Speech Incubator proposal specifies a pair of HTML elements, `<reco>` and `<tts>`, for speech recognition and speech synthesis, and it specifies an elaborate JavaScript API for interacting with these elements, providing full control over such aspects as language, model selection, endpointing, intermediate results (for ASR), and dictionaries and SSML mark handling (for TTS). It also specifies a WebSockets-based speech protocol for the communications between the browser and the back-end ASR and TTS servers. This proposal is not yet part of any standards track, but Microsoft is releasing a prototype implementation[28] with additional specifications for media-capture capabilities.[29]

---

[25] http://dev.w3.org/html5/spec.

[26] https://docs.google.com/View?id=dcfg79pz_5dhnp23f5.

[27] http://www.w3.org/2005/Incubator/htmlspeech.

[28] http://html5labs.interoperabilitybridges.com/prototypes/media-capture-api/media-capture-api/info.

[29] http://lists.w3.org/Archives/Public/www-archive/2011Mar/att-0001/microsoft-api-draft-final.html#capture_api_extensions.

### 7.9.2 WebRTC

WebRTC[30] is a set of evolving proposals for Real-Time Communications (RTC) support in web browsers. It is focused on providing multimedia capture, local and remote transmission, and display. While the main interest of WebRTC is to provide standards-compliant web interfaces for voice telephony and video conferencing and the like, WebRTC also provides a way for automated speech and language processing components to join in audio and video conversations, interacting with humans using speech recognition and speech synthesis.

WebRTC supports multimedia communications through several JavaScript APIs. The primary API is `MediaStream`, which encapsulates a complete multimedia stream, including all its audio and video tracks; the `MediaStream` objects provide one or more `MediaStreamTrack` objects, each encapsulating one video stream or one group of audio channels (e.g., an audio track can be stereo, and so have two channels).

To obtain a media stream capturing audio and/or video from microphones or cameras attached to the local computer, the browser provides the `LocalMediaStream` API. Instances of this API are obtained from `getUserMedia()`, which is a proposal currently being designed by the W3C Media Capture Task Force. Media can be recorded from a `MediaStream` object using the `MediaStreamRecorder` API, and media play-back is accomplished using the HTML5 `<audio>` and `<video>` elements.

WebRTC clients communicate with peers using the `PeerConnection` API. The `PeerConnection` API provides a communications channel that can traverse routers using Network Address Translation (NAT) in both directions using STUN or TURN protocols, with connections negotiated using the ICE protocol. Once the connection is established, the communication protocol proposed by the IETF WebRTC group will use SCTP to provide unordered, unreliable message passing, over DTLS to provide message security, over UDP.

The STUN[31] (Session Traversal Utilities for NAT) protocol is a simple mapping service which, for a given host and port that are behind a NAT router, provides a host and port on the Internet that are configured to be forwarded to the hidden host and port. For a host to be reachable in this manner, all NAT routers between it and the Internet must be configured to forward IP traffic for the appropriate ports, and the host/port mapping must be entered into a publicly visible STUN server.

The TURN[32] (Traversal Using Relays around NAT) protocol deals with NAT traversal by eliminating the need for a host on the Internet to initiate a connection to hosts behind a NAT router, by using a relay on the Internet, that both endpoints can contact to establish a connection. Under this scheme, there is no true peer-to-peer communication between the endpoints; all communications are routed

[30] http://dev.w3.org/2011/webrtc/editor/webrtc.html.

[31] http://tools.ietf.org/html/rfc5389.

[32] http://tools.ietf.org/html/rfc5766.

through the TURN relay. The ICE[33] (Interactive Connectivity Establishment) protocol is used to negotiate STUN or TURN connections, providing a high-level abstraction for clients. The SCTP[34] (Stream Control Transmission Protocol) is a transport layer protocol that provides reliable message-based communications, combining the strengths of TCP (reliability) and UDP (message-based, possibly out-of-order transmission and delivery). For secure communications, the datagram layer DTLS[35] (Datagram Transport Layer Security) can be used on top of SCTP.

While WebRTC provides no explicit support for speech recognition and synthesis, it does offer a rich and flexible framework for interactive peer-to-peer and client–server multimedia streaming and control, which is well suited for multimodal applications.

## 7.10 Conclusions

The speech mashup architecture proposes a general framework for mashing up speech and web services in a multimodal computation environment. It provides a web-service-based access to speech processing resources (ASR and TTS) in the "cloud" and different architecture configurations to minimize network latencies and increase security. Current mashup systems support application development for mobiles, desktops, and potentially any networked device, using native and web-based APIs. Ongoing advances in the standardization of network-based multimedia communications protocols will continue to lower the barrier of entry for rapid development, not to mention improving the interoperability upon which mashup architectures rely. We have discussed several building blocks for mashup applications, and shown how these blocks are realized in one publicly accessible web-based portal. This portal grants easy access to grammar tools, voice building, and service administration.

**Acknowledgements** We would like to thank Linda Crane and Amanda Stent for their contributions and continuous unconditional support.

## References

1. Allauzen C, Mohri M, Roark B (2005) The design principles and algorithms of a weighted grammar library. Int J Found Comput Sci 16:403–421
2. Alonso G, Casati F, Kunoand H, Machiraju V (2003) Web services. Springer, Berlin
3. Axelsson J, Cross C, Lie HW, McCobb G, Raman TV, Wilson L (2001) XHTML+voice profile 1.0. http://www.w3.org/TR/2001/NOTE-xhtml+voice-20011221

[33] http://tools.ietf.org/html/rfc5245.

[34] http://tools.ietf.org/html/rfc2960.

[35] http://tools.ietf.org/html/rfc6083.

4. Beckham JL, Di Fabbrizio G, Klarlund N (2001) Towards SMIL as a foundation for multimodal, multimedia applications. In: Eurospeech 2001, European conference on speech communication and technology
5. Berners-Lee T, Fielding RT, Frystyk Nielsen H (1996) Hypertext transfer protocol—HTTP/1.0. http://www.w3.org/Protocols/HTTP/1.0/spec.html
6. Beutnagel M, Conkie A, Schroeter J, Stylianou Y, Syrdal A (1999) The AT&T next-gen TTS system. In: Joint meeting of ASA, EAA, and DAGA
7. Black AW (2002) Perfect synthesis for all of the people all of the time. In: IEEE 2002 workshop on speech synthesis
8. Blechschmitt E, Strödecke C (2002) An architecture to provide adaptive, synchronized and multimodal human computer interaction. In: Multimedia'02: proceedings of the tenth ACM international conference on multimedia, pp 287–290
9. Burnett DC, Walker MR, Hunt A (2004) Speech synthesis markup language (SSML) version 1.0. http://www.w3.org/TR/2004/REC-speech-synthesis-20040907/
10. Caceres R, Friday A (2012) Ubicomp systems at 20: progress, opportunities, and challenges. IEEE Pervasive Comput 11:14–21
11. Chudnovskyy O, Weinhold F, Gebhardt H, Gaedke M (2012) Integration of telco services into enterprise mashup applications. In: Current trends in web engineering. Lecture notes in computer science, vol 7059, pp 37–48
12. Conkie A, Okken T, Kim Y-J, Di Fabbrizio G (2012) Building text-to-speech voices in the cloud. In: Eight international conference on language resources and evaluation (LREC'12)
13. Crockford D (2006) The application/json media type for JavaScript object notation (JSON). RFC 4627. http://www.json.org/
14. Di Fabbrizio G, Okken T, Wilpon JG (2009) A speech mashup framework for multimodal mobile services. In: ICMI-MLMI 2009, pp 71–78
15. Eberman B, Carter J, Meyer D, Goddeau D (2002) Building VoiceXML browsers with OpenVXI. In: WWW 2002: proceedings of the 11th international conference on world wide web, pp 713–717
16. Feng J, Bangalore S, Gilbert M (2009) Role of natural language understanding in voice local search. In: 10th annual conference of the international speech communication association (Interspeech 2009)
17. Fielding RT (2000) REST: architectural styles and the design of network-based software architectures. PhD thesis, Univ. of California, Irvine. http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
18. Gebhardt H, Gaedke M, Daniel F, Soi S, Casati F, Iglesias CA, Wilson S (2012) From mashups to telco mashups: a survey. IEEE Internet Comput 16(3):70–76
19. Goffin V, Allauzen C, Bocchieri E, Hakkani-Tür D, Ljolje A, Parthasarathy S, Rahim M, Riccardi G, Saraclar M (2005) The AT&T WATSON speech recognizer. In: IEEE international conference on acoustics, speech and signal processing (ICASSP)
20. Gruenstein A, McGraw I, Badr I (2008) The WAMI toolkit for developing, deploying, and evaluating web-accessible multimodal interfaces. In: ICMI 2008: proceedings of the 10th international conference on multimodal interfaces, pp 141–148
21. Johnston M (2009) EMMA: extensible multimodal annotation markup language. http://www.w3.org/TR/emma/
22. Johnston M, Di Fabbrizio G, Urbanek S (2011) mTalk—a multimodal browser for mobile services. In: Interspeech 2011, 12th annual conference of the International Speech Communication Association, pp 3261–3264
23. Lee K-F, Hon H-W, Reddy R (1990) An overview of the SPHINX speech recognition system. IEEE Trans Acoust Speech Signal Process 38(1):35–45
24. McGlashan S, Hunt A (2004) Speech recognition grammar specification version 1.0. http://www.w3.org/TR/2004/REC-speech-grammar-20040316/
25. Newcomer E, Lomow G (2004) Understanding SOA with web services. Addison-Wesley Professional, Reading

26. Niklfeld G, Finan R, Pucher M (2001) Architecture for adaptive multimodal dialog systems based on VoiceXML. In: EuroSpeech 2001
27. Niklfeld G, Pucher M, Finan R, Eckhart E (2002) Mobile multi-modal data services for GPRS phones and beyond. In: Fourth IEEE international conference on multimodal interfaces, pp 337–342
28. Porter B, McGlashan S, Burke D, Burnett DC, Candell E, Auburn RJ, Baggia P, Carter J, Rehor K, Oshry M, Bodell M, Lee A (2007) Voice extensible markup language (VoiceXML) 2.1. http://www.w3.org/TR/2007/REC-voicexml21-20070619/
29. Povey D, Ghoshal A, Boulianne G, Burget L, Glembek O, Goel N, Hannemann M, Motlicek P, Qian Y, Schwarz P, Silovsky J, Stemmer G, Vesely K (2011) The Kaldi speech recognition toolkit. In: IEEE 2011 workshop on automatic speech recognition and understanding
30. Shanmugham S, Burnett D (2009) Media resource control protocol version 2 (MRCPv2). tools.ietf.org/html/draft-ietf-speechsc-mrcpv2-18
31. Wang K (2002) SALT: an XML application for web-based multimodal dialog management. In: NLPXML 2002: 2nd workshop on NLP and XML, pp 1–8
32. Yee R (2008) Pro Web 2.0 mashups: remixing data and web services. APress, New York

# Chapter 8
# Mashups for the Emergency Management Domain

**Axel Schulz and Heiko Paulheim**

**Abstract** Emergency management applications support a command staff in disruptive disaster situations, such as earthquakes, large-scale floodings or fires. One crucial requirement to emergency management systems is to provide decision makers with the relevant information to support their decisions. Mashups can help here by providing flexible and easily understandable views on up-to-date information. In this chapter, we introduce a number of mashups from the domain of emergency management. An in-depth study of the mashup *MICI* shows how mashups can combine valuable information for ranking and filtering emergency calls to cope with information shortage and overload. We further discuss the use of Linked Open Data both as a source of additional information and a means for more intelligent filtering.

## 8.1 Introduction

Emergency management deals with coordinating operations in situations such as earthquakes, large-scale fires and floodings, or epidemics. There is a large variety of IT solutions supporting emergency management, ranging from intelligent messaging solutions to planning support systems and geospatial infrastructures. Mashups in this domain are especially helpful to provide information to the command staff.

Information is the most important resource in emergency management as it is necessary for decision making. Decision makers must know what is happening where and at which time, and they need to know the severity and relevance of all

A. Schulz (✉)
SAP Research, Darmstadt, Germany
e-mail: axel.schulz@sap.com

A. Schulz
Telecooperation Group, Technische Universität Darmstadt, Darmstadt, Germany

H. Paulheim
Research Group Data and Web Science, Universität Mannheim, B6, 26, 68159 Mannheim, Germany
e-mail: heiko@informatik.uni-mannheim.de

B. Endres-Niggemeyer (ed.), *Semantic Mashups*,

DOI 10.1007/978-3-642-36403-7_8, 

incidents. Hence, obtaining and communicating important information in real time is a crucial task.

Milis and van de Walle show that, despite the existence of numerous research projects on the topic, IT is rarely used in emergency management nowadays for information management. The main problem is that IT solutions are applicable only if there is “a member in the crisis management with an IT background” [38], but the members of emergency management organizations are often non-IT experts [9]. Thus, the barriers for using IT solutions need to be significantly lowered.

With the advent of the Web 2.0, information across the web has been combined in various light-weight mashups, which often have a steeper learning curve than the heavy-weight IT solutions existing in the field. Quickly creating representations of multiple information sources that can be viewed in a web browser became very popular over the last years. Especially in emergency management, where decision makers have to identify and react to risks of disasters, this easy understandable way of summarizing information may be valuable to improve the situational picture and minimize the risk of wrong decisions.

While the Web and especially the Web 2.0 is a rich and comprehensive source of information, the potentially valuable information often remains unused because the sheer amount of information cannot be handled efficiently, as it exceeds human cognitive limits. Especially in this case, mashups can help by processing and visualizing that information in a useful, structured way.

In this chapter, we present several mashup approaches for filtering and presenting information relevant for decision makers in emergency management. In an in-depth study, we present our prototype MICI which helps classifying and structuring an unordered stream of data from the Seattle fire brigade.

The rest of this chapter is structured as follows: In Sect. 8.2, we describe the domain of emergency management and discuss particular requirements to mashups in that domain. In the subsequent sections, we discuss examples for mashup solutions in the field. Section 8.4 discusses an in-depth view on the mashup application *MICI*. We conclude the chapter with a short summary and an outlook on future work.

## 8.2 The Emergency Management Domain

Emergency management is “the discipline and profession of applying science, technology, planning and management to deal with extreme events that can injure or kill large numbers of people, do extensive damage to property, and disrupt community life” [18]. Improving situational awareness, i.e., “the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future” [19], is one of the main goals for efficient decision making as it is critical for the success of the operation in the first 72 hours of a disaster [20].

At the beginning of a crisis, the emergency management staff begins to gather and analyze information from multiple sources. These might be web sites, governmental information, and information provided by units in the field via radio or mobile phone. In this case, human reconnaissance is required for two reasons [29]: first, to know the current state of an object (e.g., a bridge: is it bending?) and second, to compare it to the "normal" conditions (does the bridge usually bend that way?). In this case, the decision maker has to cope with a concurrent information overload and shortage. While information provided in social media can be manifold, this kind of information is unstructured and has further to be verified. On the other hand, the amount of information from on-site rescue squads and bystanders is low. Using mashups, spatial, temporal, and thematic information about events can be used for increasing situational awareness.

Recent research shows the crucial importance of information for emergency management. With the increasing adoption of smartphones with multiple sensors and a constant Internet connection [14], humans as soft sensors became valuable sources of crisis information in participatory sensing environments. Social media has been widely used for sharing messages about incidents, e.g. Twitter was used to report on incidents like the Oklahoma grass fires and the Red River floods in April 2009 [48] or the terrorist attacks on Mumbai [23]. Plenty of useful situational information was provided by citizens in these cases. Furthermore, the analysis of social media is valuable for detecting events in real time or tracking diseases [42, 46]. Social media reports are often more timely and detailed than reports on official news sites, but on the other hand more noisy and more difficult to handle in terms of quantity.

In addition to generating new information, crowd-sourcing mechanisms have been applied several times in emergency management to filter and aggregate information, e.g. during the Haitian earthquake. In this case, crowds can be used to significantly decrease the time for generating a map of the affected area [31].

Beside social media, the Semantic Web and Linked Open Data (LOD) have grown as another branch of information. While the social web consists of text, images, and videos, all of which cannot be processed by intelligent agents easily, the Linked Open Data cloud contains semantically annotated, formally captured information. Especially the increasing amount of Linked Open Government Data (LOGD) from the public sector [11] contributes about 42 % to the LOD cloud [12]. This information source could also provide valuable background information for decision making. However, methods and tools for using those data in emergency management are still being developed.

There are various characteristics that make mashups attractive for the emergency management domain. First, emergency management relies on *timely information*. Mashups involving Web 2.0 and social media sources such as Twitter can provide more timely information than official or commercial news web sites [10]. Second, emergencies are *disruptive and unforeseeable* by nature, which makes it hard to combine all the required information sources upfront. The flexible nature of mashups helps combining the required sources of information in an on-the-fly manner to serve the right information at the right time.

On the other hand, introducing mashups also introduces a set of challenges. One particular requirement of emergency management is the need for *reliable information*, another particular challenge is dealing with *information overload*, which can be easily introduced when presenting the command staff with an unfiltered stream of Twitter messages. Thus, mashups in the emergency management domain require *intelligent information filtering and ranking* mechanisms. In the following chapters, we will show how mashups are used to address information needs in the emergency management domain, and discuss approaches to address the above challenges.

## 8.3 Overview of Mashups for the Emergency Management Domain

Various approaches exist for using mashups in the emergency management domain. Those approaches differ on several dimensions. For categorizing these, we use the following three dimensions: information sources, processing methods, and user interfaces.

A mashup usually comprises many *information sources*:

- Static web content, e.g. text snippets taken from web sites as well as structured information from the web.
- User-generated content from Web 2.0 sources like tweets from Twitter or pictures on Flickr.
- Open Government Data, e.g. real-time or historical information from http://www.data.gov/.
- Linked Open Data as a source of structured information.
- Sensor devices such as water level or earthquake sensors.
- Human observations provided as soft sensors, e.g. traffic conditions. In contrast to Web 2.0 sources specialized interfaces for the intended application domain are used.

The information from various sources needs to be further *processed* in order to get meaningful insights from the raw data. In this chapter, we concentrate on intelligent mashups. We review approaches where the information sources are automatically or semi-automatically enriched with further knowledge. Natural language processing (NLP) methods are often used for extracting relevant topics and/or information snippets from text, or for classifying text messages. Information items may be semantically annotated (tagged and linked) with metadata and external knowledge, such as Linked Open Data sources. For structuring data and identifying doubles or outliers, machine learning and clustering methods can be applied. If problems are too difficult for being fully automated, crowd-sourcing may help to solve them, e.g., by asking humans to categorize information items.

The processed information can then be presented to the emergency management staff in various different *user interfaces*. While map interfaces are the most wide spread, some user interfaces also include multi-media data. Furthermore, other interfaces exist, such as expert query interfaces.

Table 8.1 summarizes the approaches discussed in this chapter, and shows which information sources, processing methods, and user interfaces they comprise. In the subsequent sections, we discuss those approaches in more detail.

**Table 8.1** Overview of mashup approaches

| Mashup | Information sources | | | | | | Processing methods | | | | | User interface | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Static web content | Web 2.0 | Govern-ment data | Linked Open Data | Sensors | Citizen sensors | NLP, infor-mation extrac-tion | Seman-tic annota-tion | Machine learning, cluster-ing | Infor-mation filtering | Crowd sourc-ing | Map | Multi-media | Other |
| Twitris | | X | | X | | X | X | X | X | | | X | X | |
| SemSor | | X | | X | | | | X | X | X | | X | X | X |
| Disaster 2.0 | | | | | | X | | | | | X | X | X | |
| Linked Sensor Middleware | | | | X | X | | | X | | X | | X | X | X |
| Twitcident | | X | | X | | | X | X | | X | | | | X |
| Repopulation Indicators for New Orleans | | | X | | | | | | | | | X | | |
| IBISEYE | | | | | X | | | | | | | X | | |
| EDIS | | | X | | X | | | | | X | X | X | | |
| Live Earthquake | | | | | X | | | | | | | X | | |
| Healthmap | X | | X | | | | | | | X | | X | | |
| MediISys | X | X | | | | | X | | | X | | X | | X |
| LA Fire Tweets, SwineFluTweets, Iran Protest Tweets | | X | | | | | | | | | | X | | |
| WikiCrimes | | | | | | X | | | | X | | X | | |
| LA Fires, Bushfire Incidents, BushfireConnext | X | | X | | | | | | | X | | X | | |
| PakReport | | | | | | X | | | | | | X | | |
| Ushahidi | | X | | | | X | | | | X | X | X | | |
| FindShelter | X | | | | | X | | | X | X | | X | | |
| MICI | | | X | X | | | | X | | X | | X | | |

### 8.3.1 Twitris

Sheth et al. present an approach for real-time monitoring of tweets, e.g. for the identification of disaster events [28]. Twitris[1] extracts citizen perceptions of these events and presents related information. In this case, the Twitris system uses spatial, temporal, thematic and sentiment analysis on tweets and SMS to analyze events. The idea of the approach is to extract context information from microblogs and SMS and to take background information into account, like news, pictures, video or Wikipedia. Unfortunately NLP techniques do not work well on short informal text, and in addition one has to cater for domain-specific language use on social platforms such as interposed hashtags, @'s, or fancy user names. The whole pipe consists of six steps [45]:

1. The Twitter Search API is used for fetching tweets, while the Ushahidi API is used for collecting SMS. For crawling social media, a set of keywords is extracted using a semantic model from DBpedia [7] and a statistical analysis. Related concepts from DBpedia to the events at hand are used. The statistical analysis is based on trends, e.g. hashtags used by users and analysis from Google Insights.
2. The text is analyzed with spatio-temporal-thematic (STT) bias to extract event Descriptors. To identify N-gram summaries for tweets, first, a Spatio-Temporal cluster for an event is created. TFIDF is used to generate the N-grams from these data. Second, spatial, temporal and thematic bias to the N-grams are used to weight the relevance for the events. Last, domain models are created using the event context.
3. Semantics are captured from internal (annotations, related posts), external (external sources) and mined internal context (sentiment analysis).
4. Domain models are created automatically to understand the meaning of event descriptors.
5. Entities are semantically annotated for knowledge discovery. Twarql [37] is used to represent social media as LOD.
6. External information sources are integrated using the semantic similarity between the contexts.

The UI has components for theme, time and space. All descriptors for events are represented on this map using the spatial attribute and the date. Additional information is represented using widgets, e.g. for media items, graphs, images, videos, and sentiment analysis. An example screenshot of the UI is shown in Fig. 8.1.

### 8.3.2 SemSor

The SemSor project supports the situational assessment for emergency management based on social media analysis [26]. The system constantly crawls social media

[1] Twitris is available on http://twitris.knoesis.org.

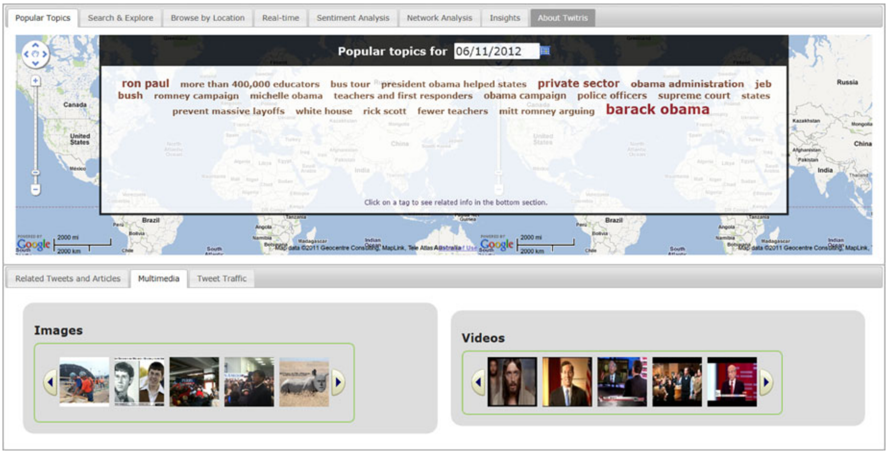



**Fig. 8.1** Twitris user interface showing topic analysis and related multi-media content

sources like Twitter, Flickr or YouTube. The textual entries of the social media items are annotated with links to entities in the semantic web (the LOD cloud).

Datasets in the LOD cloud are used to start the Spreading activation [17]. This enables the retrieval of related entries to the previously identified entities that might also be useful. The access on Linked Open Data resources using SPARQL can trigger the spreading activation. Furthermore, contextual information like location, time, or weather data can be used to change the mechanisms. The user's actions can be used to refine the link weights and to adapt the weight of links and the spread results. Since spreading activation is not limited to a specific domain or dataset, the SemSor approach is feasible for all LOD datasets.

The search query and the results are presented in different views as shown in Fig. 8.2. In those views, the search query can also be manipulated, which causes a changing of the weights used by the spreading activation mechanism, which might identify other relevant information. A result browser is showing all results so that it targets their semantic relevance. Furthermore, all relevant entries can be viewed on a map or on a timeline. The map and the timeline views can be used to define additional temporal and location constraints to filter the information.

### *8.3.3 Disaster 2.0*

The Disaster 2.0 system follows the idea of using citizen sensors and crowd-sourcing to manage information about natural disasters [41]. To that end, the main entities like events (fire, flood), allocated resources (policemen, firemen) and damages (victims) are explicitly modeled in the system with unique identifiers. Information about those entities can be obtained by asking citizens.

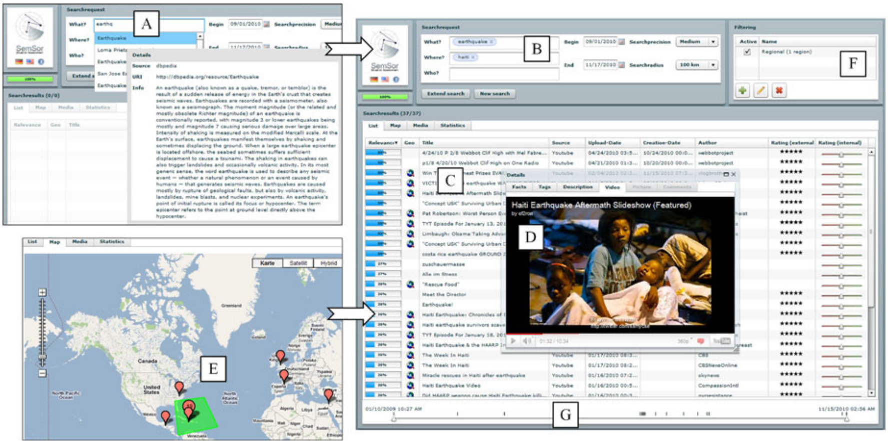



**Fig. 8.2** Screenshots of different components of the SemSor UI [26]: forming a search query (**A**, **B**). Relevant information is displayed in a list (**C**) or in a separate window (**D**). Spatial (**E**, **F**) and temporal filters (**G**) can be applied

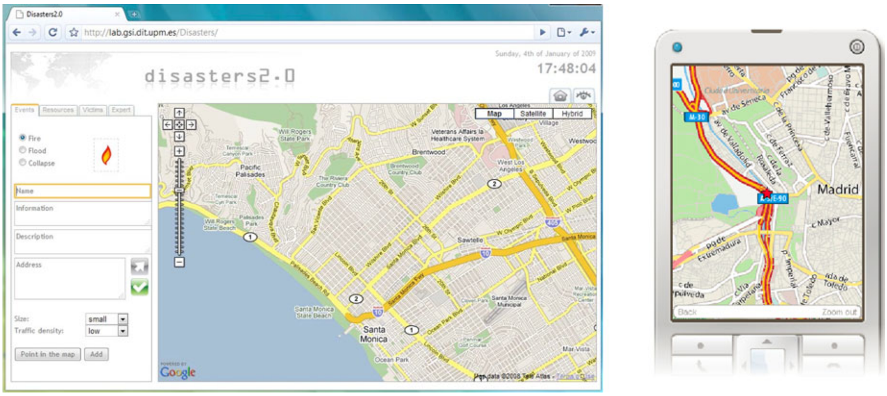



**Fig. 8.3** Screenshot of the web client of Disaster 2.0 and the mobile UI for displaying disasters on a map [41]

The Disaster 2.0 interface visualizes the information using a Google Map view as shown in Fig. 8.3. On this, information sources are provided as markers with additional descriptions and the corresponding location. Additional information might be the magnitude of the disaster or the traffic around the affected area. Furthermore, the map enables the manual association of different information sources by dragging items on each other. For example, victims can be related to disasters this way. Important buildings for the rescue operations can also be displayed. In this case, a simple selection is possible based on the information provided by Google Maps.

A mobile application has been developed in the project as another way of displaying the contents of the map. This application can be used to refine this information, like the correct position of resources. With this mobile application, crowd-sourced information can be obtained on a larger-scale, using the workforce of many citizens in the disaster area.

For the support of rescue forces, a rule engine is used to automatically assign incidents to the closest available resources. The rules use information such as the type and size of the incident, the number of victims, and traffic information. These rules can be edited by emergency personnel manually. This expert system works in real time, i.e., updates of available resources are directly triggered to disasters and vice versa. Furthermore, a multiagent system for supporting the decision making has been developed. This system enables rescue squads to query the active disasters and to assign to these. Furthermore, this system enables the coordination of all rescue squads as these send their status and location information to the systems.

### *8.3.4 Linked Sensor Middleware*

The Linked Sensor Middleware (LSM) platform is designed to combine sensor data and the Semantic Web based on Linked Data principles [35]. The purpose of the middleware is to make sensor streams usable by integrating them with existing information. The whole concept is based on the idea of Linked Stream Data [44] to provide a possibility to publish heterogeneous sensor data as Linked Data to enable the connection to other Linked Data collections. The middleware is divided into four layers:

1. The data acquisition layer provides a wrapper to collect sensor data. These data are transformed to RDF, annotated with links and stored in a database. Based on the Semantic Sensor Ontology,[2] two layers are defined for annotating sensor data: the static layer describes the sensor metadata, while the dynamic layer contains a graph-based stream from the various sensor readings.
2. The Linked Data layer provides access to the sensor data and links to additional information in the LOD cloud. For example links are provided to external sources like DBpedia [7], Geonames,[3] or LinkedGeoData [8] to identify points of interest near a sensor location.
3. The Data Access Layer enables the querying of resources based on a remote SPARQL endpoint. This can be used to filter or integrate existing data into new streams.
4. The application layer provides access to the information offered by the system. For example, the sensor information can be displayed on a map where information like weather, departure times of trains, street cameras are shown. An example visualization is shown in Fig. 8.4.

[2] http://purl.oclc.org/NET/ssnx/ssn.

[3] http://www.geonames.org/.

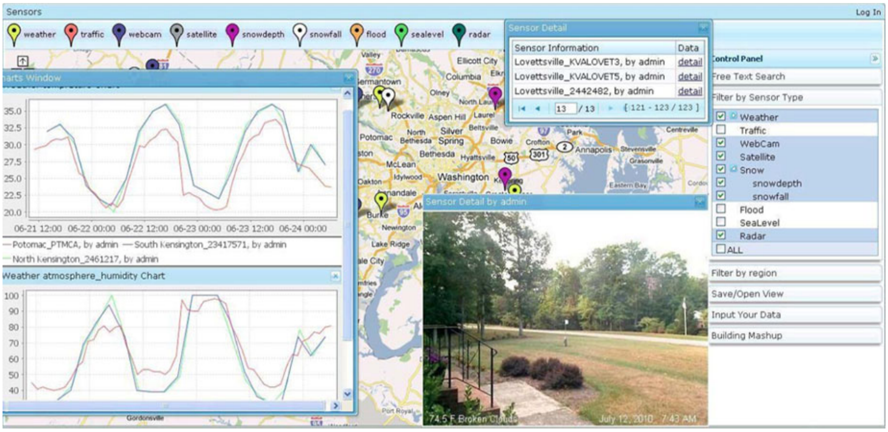



**Fig. 8.4** Screenshot of the UI of the Linked Stream Middleware [34]

The Live Linked Open Sensor Database project [34] is based on the Linked Sensor Middleware. Two application scenarios for emergency management are mentioned: Possible threats can be taken into account, if information from sensor sources is available like the wind force or temperature in case of a fire. Furthermore, during diseases the combination of user-generated content and sensor data might help to prevent the spread of a disease.

Currently 200,000 sensor readings have been integrated into the platform. The different sensors are displayed on a map [22] that enables the filtering of data sources with browsing facets. In this case, text input, a taxonomy or spatial information can be used for filtering. Additionally, the SWEET ontology[4] is used to provide vocabularies for natural phenomena like blizzards or snow fall, which can be used for filtering too. Using the view, current and historical sensor data can be compared. The DERI Pipes engine[5] has been integrated, to enable the user to visually combine sensor data sources for publishing them as new sensors. The view is shown in Fig. 8.5.

### 8.3.5 *Twitcident*

Twitcident[6] is a mashup for filtering, searching and analyzing social media information about incidents [2]. The purpose is to automatically filter relevant information from social media and making these information accessible and findable dependent on the given context.

[4]http://sweet.jpl.nasa.gov/.

[5]http://pipes.deri.org/.

[6]http://wis.ewi.tudelft.nl/twitcident/.

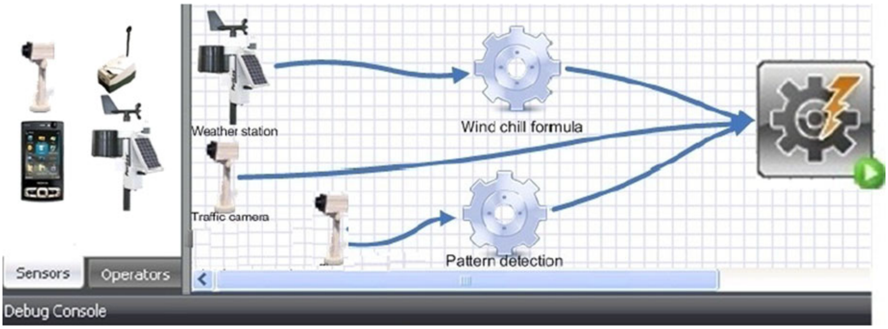



**Fig. 8.5** Screenshot of the Sensor Composer based on DERI pipes [34]

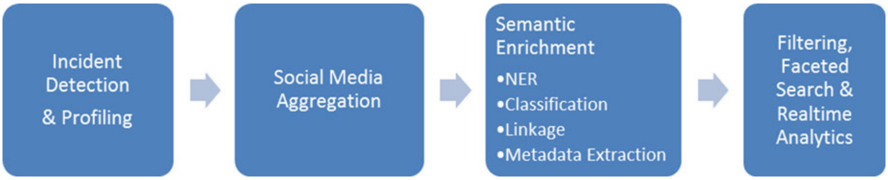



**Fig. 8.6** The Twitcident processing pipeline based on [2]

For incident detection, information about incidents published in the P2000 network[7] is collected, which is used by the public emergency services in the Netherlands. The network provides information about the type of incident, its location and its temporal attributes. Furthermore, the incident is scaled for severity in this system. The Twitcident framework transforms these information into an initial query for collecting potentially relevant Twitter messages (tweets) using the Twitter Streaming API.[8]

Four steps as shown in Fig. 8.6 are applied for improving the filtering of tweets [3].

First, based on the incident information, an incident profile is created which describes the attributes of the incidents. This integrates related concepts like locations or persons and weights of their importance for the incident. These profiles are constantly updated with changing information.

As a second step, a social media aggregation module retrieves tweets based on the incident profile, related pictures and videos to a post.

As a third step, the Semantic Enrichment module consists of four different components for processing the information. Before applying the enrichment, non-English tweets are translated to English using the Google Translate API,[9] because

[7] http://en.wikipedia.org/wiki/P2000_(network).

[8] https://dev.twitter.com/docs/streaming-apis.

[9] https://developers.google.com/translate/.

the components only work in this language. The NER component uses OpenCalais,[10] DBpedia [7], AlchemiAPI[11] and Zemanta[12] to detect entities like persons and locations in tweets. The classification component classifies each tweet into different types of reports, using categories such as different damages and risks. Furthermore, the different senses of the tweeter are differentiated based on manually defined rules containing the basic keywords of the senses. It can be identified if the tweeter sees, feels, hears or smells something. A linkage component extracts links from tweets. These links are used to provide additional context information. To that end, the content of linked web pages is analyzed. In a last component, metadata about tweets are collected. For this purpose, the metadata like the user profile information provided by the Streaming API are used.

As a fourth step, keyword-based and semantic filtering can be applied to automatically identify relevant tweets for an incident. The keyword-based filtering can be used to define a query. The query itself is evaluated using a relevance model based on RM2 [33]. The semantic filtering is based on the enriched tweets.

The user is enabled to search the information based on faceted search [1] and analytics capabilities. A faceted search can be used to define pairs of attributes that have to be matched by all returned tweets. Different strategies for querying can be applied in this case [2]. The information found can be analyzed using different widgets, for example the geographical impact can be identified. The filters applied in the faceted search will also be applied on the analytics widgets. The search and analytics functionality is available to client users in a web front-end as shown in Fig. 8.7.

The framework has been evaluated with respect to the filtering capabilities. As a result, the faceted search seems to be the most appropriate approach for filtering tweets relevant for an incident.

### *8.3.6 Crisis Map Mashups*

Map mashups are the combination of different sources of information that are displayed in some geographic visualization. Recently updated resources like governmental data or crowd-sourced information can be visualized on the user interface as spatially and temporally changing events. In this case, crisis map mashups can be used to report, assist and manage emergencies. Zang et al. [49] describe map mashups as a common form of data mashups, because they are the most visual and adaptable of the mashup options. Several analyses on crisis map mashups have been made e.g. by [36].

Generally, crisis map mashups can be roughly categorized into mashups that show information from only one source, e.g., Twitter messages on a given topic, and

[10] http://www.opencalais.com/.

[11] http://www.alchemyapi.com/.

[12] http://www.zemanta.com.

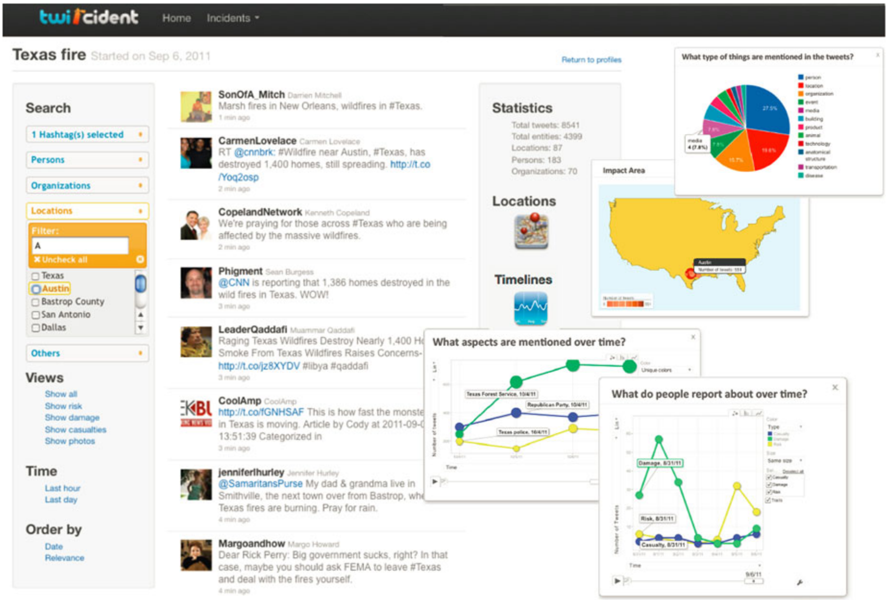



**Fig. 8.7** Screenshot of the Twitcident interface, showing the search and filtering (*left*), the retrieved social media (*middle*) and the analytic components (*right*) [2]

those that aggregate information from different sources. Furthermore, the mashups often employ some intelligent processing for aggregation, e.g., topic and entity detection for relating pieces of information from different sources.

#### 8.3.6.1 Map Mashups Using a Single Source of Information

The *Repopulation Indicators for New Orleans* [25] is intended to visualize the repopulation in New Orleans, USA after the Hurricane Katrina in 2005 [36]. To that purpose, information from the U.S. Postal Service Delivery Statistics and a Housing and Urban Development Census[13] is used. The *IBISEYE* [27] mashup provides an overview about current storms, their strengths and current weather alerts. In this case, information from WeatherFlow[14] is used.

The combination of user-generated content with geographic maps has become popular during the last years. *Los Angeles Fire Tweets* [4] displays all tweets related to the string “fire” in a 100 mile radius around Los Angeles, USA. *Swine Flu Tweets* [6] shows the distribution of swine flu in the world using the Twitter API. *Iran Protest Tweets* [5] displays tweets related to the Iran protests.

[13] http://www.hudhdx.info/.

[14] http://www.weatherflow.com/.

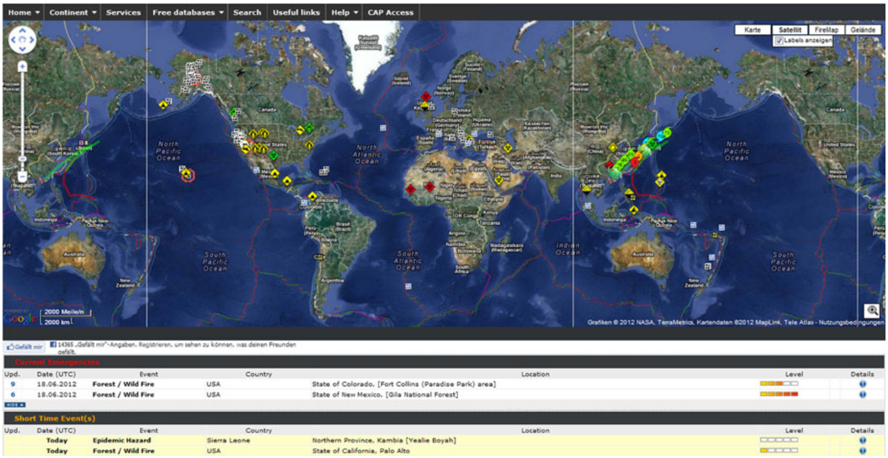



**Fig. 8.8** Screenshot of the RSOE AlertMap

*Vikalpa* [15] is an initiative launched in 2008 to gather and display user-generated reports on election related violence. Reports are created by local journalists and shown on a map to locate related incidents.

#### 8.3.6.2 Map Mashups Combining Multiple Information Sources

The Hungarian National Association of Radio Distress-Signalling and Infocommunications (RSOE) operates the Emergency and Disaster Information Service (EDIS) which provides a so called Disaster and Emergency AlertMap [39]. This map (see Fig. 8.8) displays emergencies obtained from governmental sources around the world, like wild fires, explosions as well as tsunami and earthquake warnings. Furthermore, detailed information about the events can be displayed in a separate view, e.g. the damage level and the number of affected people. The Live Earthquake [16] mashup is more specific for displaying earthquakes of the last seven days on a map and a timeline. The mashup uses data from the U.S. Geological Survey,[15] the European–Mediterranean Seismological Centre[16] and the GFZ Potsdam[17] that are integrated.

Other governmental sources are used to present data from the public health sector. Healthmap [13] is provided by the Children's Hospital Boston, US, as a mashup for disease outbreak monitoring. In this case, official sources like ProMED Mail,[18]

[15] http://www.usgs.gov/.

[16] http://www.emsc-csem.org/Earthquake/.

[17] http://geofon.gfz-potsdam.de/eqinfo/eqinfo.php.

[18] http://www.promedmail.org/.

the WHO[19] and the FAO[20] are used. Furthermore, news feeds from Google and Chinese search engines are aggregated. All available information like the time, the type of a disease and the affected species are displayed on a map, where every push-pin displays the related news. All information can furthermore be accessed using the Outbreaks Near Me mobile application, where the same information as on the website is displayed. This app can also be used for reporting about new known outbreaks. MediISys [30] follows a similar approach to display information about diseases on a map. Furthermore, statistical analyses of recent and past events are provided. News from the WHO, CDC[21] and EMM[22] are aggregated and displayed in a list. All provided information about alerts and important topics can further be filtered manually.

WikiCrimes [47] was created by the University of Fortaleza, Brazil. It allows the posting of user-generated reports about criminal activities. These reports are aggregated and displayed on a map showing the hotspots of different types of crimes (e.g. theft, robbery). The view can be filtered based on the crime types, temporal attributes and the credibility of the reports. Furthermore, one can receive alerts about crimes in a predefined location.

For displaying warnings about wildfires, the Los Angeles Fires[23] mashup developed by the LA Times integrates several information sources. Reports from LA Times reporters, satellite images and viewer's comments are combined. Bushfire Incidents[24] follows the same approach for Australia. Additionally, weather information and the current position of the fire brigades can be displayed. BushfireConnect[25] is another approach for aggregating official information about fires in Australia with user-generated reports. Information about incidents and news can be displayed on a map and in a list. These can be filtered by the type of media and different categories like the types of the emergencies, official reports or community updates.

Another form of incident reporting has been used in PakReport,[26] where users send information about floods via SMS. This information is displayed according to helpful background information like places with water or hospitals.

### *8.3.7 Ushahidi*

The Ushahidi platform is an open-source crisis management application that was developed in 2008 [40]. Ushahidi uses crowd-sourcing for incident reporting and

[19] http://apps.who.int/ghodata/.

[20] http://www.fao.org/corp/statistics/en/.

[21] http://www.cdc.gov/DataStatistics/.

[22] http://www.hse.gov.uk/events/index.htm.

[23] http://www.latimes.com/news/local/la-me-la-fire-map-html,0,7464337.Htmlstory/.

[24] http://www.aus-emaps.com/fires.php.

[25] http://bushfireconnect.org/.

[26] http://pakreport.org/flood2010/.

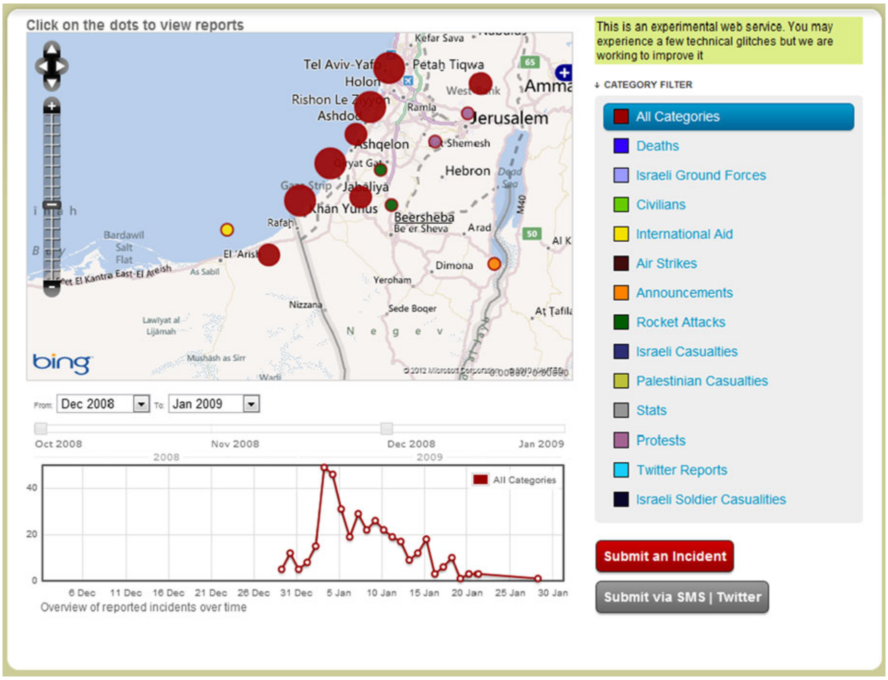



**Fig. 8.9** Adoption of the Ushahidi platform from the Al Jazeera Labs for tracking events around the Gaza strip. http://labs.aljazeera.net/warongaza/

aggregation and became world-wide known from the Haitian earthquake, where the platform was used for translating Creole text messages by crowd-sourcing mechanisms. The platform has now been used for several campaigns over the last years, e.g. for tracking events around the Gaza Strip, as shown in Fig. 8.9.

In a first scenario during the elections 2008 in Kenya, citizens could post reports of incidents to the platform [32]. Reports can be sent via SMS, mail or Twitter. In a simple mashup based on the Kohana framework[27] and Google maps, those user-generated incident reports are displayed. Using three different timelines, the information can be filtered [36]. The first timeline filters the temporal and spatial information. The second timeline shows the number of incidents in a time period and can be used to filter temporally. The third timeline shows the changes in the number of reports over time while showing the corresponding reports on the map.

For keeping the reliability of information high, reports from incidents which are potentially not trustable have to go through an approval process conducted by the team from Ushahidi [40]. In this case, reports from anonymous people are checked against other sources like the news. The code of the platform is freely available to adapt it to any situation. For example, the framework behind Ushahidi has been

[27] http://kohanaframework.org/.

published as a cloud service as “Crowdmap” and can be used by anyone for mashup creation.

With the increasing amount of data on the platform, the Ushahidi, Inc. developed the SwiftRiver toolset as a complementary product [24]. The SwiftRiver toolset provides several APIs for filtering and structuring data from multiple information sources. Social media, blogs or mobile applications can be used as an information source. The fetched data can be processed by the different APIs. For example, the location of the information is identified. Content is analyzed lexically and duplicates are removed. As a result, contextualized messages are created, which can be processed by asking questions like “Where is the person at” or “What has happened” based on the predefined tags provided in a taxonomy.

### *8.3.8 FindShelter*

*FindShelter*, the approach described by [21] is based on crawling web pages for important information like emergency accommodations as this source of information cannot easily be aggregated by decision makers. For this, relevant information about entities is retrieved and provided as a web feature service (WFS). The aggregation process is structured as follows.

First, information sources are defined manually. In this case, web pages of a specific domain can be used. As a second step, relevant information is crawled from the web site using breadth-first search. In a third step, information of the same type is extracted and aggregated. To that end, regular expressions are used to extract the relevant information, e.g., for phone numbers and addresses. For completion of the retrieved data it is proposed to use external sources, e.g. using the Google Geocoding API to complete geographic information, and yellow pages for providing additional contact information. In a fourth step, data from different web sites are merged. For every retrieved entity, missing data are added and the attributes are replaced with the newest information. For describing entities, more detailed information can be added using this approach. For example, the different areas of expertise of a hospital can be added. In this case, a synonym table is used to map retrieved terms to standard denotations. As a last step, the retrieved data are published as RDF and as an OGC-standard web feature service and can be displayed on a map view, as shown in Fig. 8.10.

While it is possible to search directly for hospitals with a search engine like Google Maps, the FindShelter approach is also capable of retrieving and aggregating information on more complex entity types, such as emergency shelters, which can comprise, e.g., schools, convention centers, and sports halls.

## 8.4 Case Study: MICI

MICI (Mashup for Identifying Critical Infrastructure) [43] is a mashup application which showcases how Linked Open Data can be used as relevant background knowl-


**Fig. 8.10** FindShelter using a visualization based on Gaia. *Black points* are hospitals, *white points* are emergency accommodations

edge in emergency applications. It addresses several central information needs of emergency response staff:

- Many incoming messages need to be prioritized. In situations where the resources are limited and a large number of individual incidents exist, it is necessary to decide which ones need response most urgently.
- Background knowledge about incidents is required. For example, it is important to know which nearby schools possibly need evacuation.
- Information overload has to be avoided. Despite the large number of incidents and the even larger amount of potentially relevant background knowledge, the command staff still has to be able to get a clear situational picture.

MICI reads Open Government Data, e.g., as RSS data, which contain information about incidents. Users may define rules which classify the severity of an incident, e.g.: *if there is a fire within 50 m radius of a gas station or a gas pipeline, the severity is high*. For evaluating such a rule, the background information about nearby objects, such as gas stations, is taken from Linked Geo Data [8], a dataset within Linked Open Data which contains information about several objects with geocoordinates. For our prototype scenario, we use the fire call dataset from the city of Seattle,[28] which provides a list of fire calls with type and coordinates, among others.

[28] http://data.seattle.gov/Public-Safety/Seattle-Real-Time-Fire-911-Calls/kzjm-xkqj.

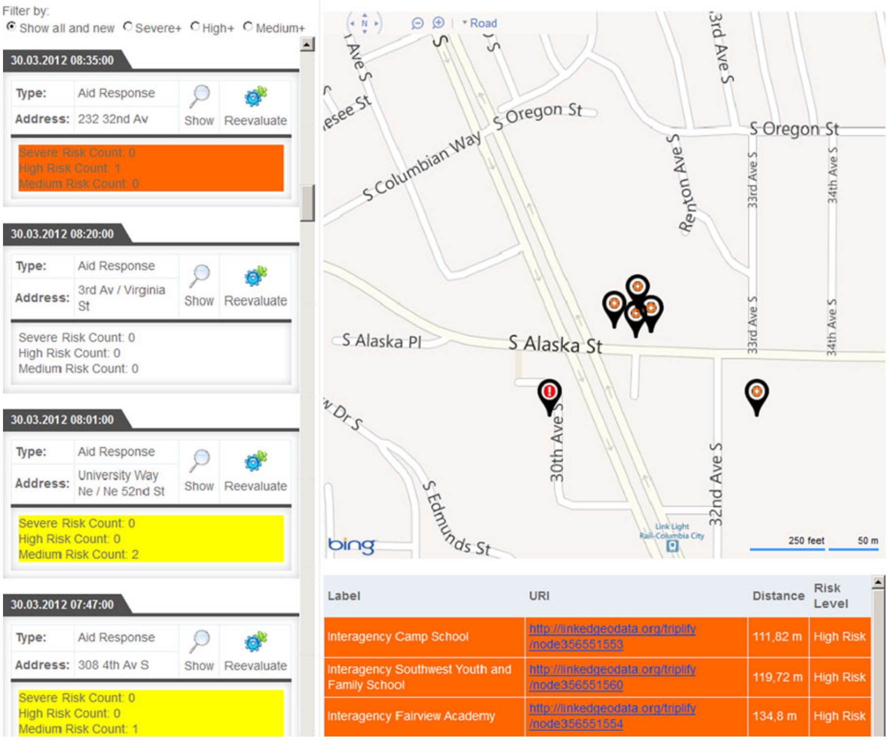



**Fig. 8.11** MICI main screen, showing a list of color-coded incidents (*left*), the map view of an incident including potentially affected infrastructure (*top right*) and a list view of those infrastructure objects (*bottom right*)

### *8.4.1 User Interface*

The main screen of the prototype shows two views on the current situational picture, as depicted in Fig. 8.11. On the left-hand side, a list of incidents is displayed, which can be filtered according to the incidents' severeness. The right hand side shows a map view. When the user selects an incident in the list, that incident and all affected infrastructure objects are displayed in the map.

Rules are used both to find relevant infrastructure objects, as well as to calculate the severeness of an incident. To define which objects are relevant for which type of incident, users may create their own rules on the rule panel. To that end, they assign a set of object types to an incident type and define a radius and a degree of severity. For defining a rule, the user needs to know about the potential types of objects on which background knowledge is available, i.e., the ontology of Linked Geo Data defining object types such as gas stations, schools, etc. This ontology is used as a vocabulary for the user to define rules, while the vocabulary for the incident types depends on the RSS source used.

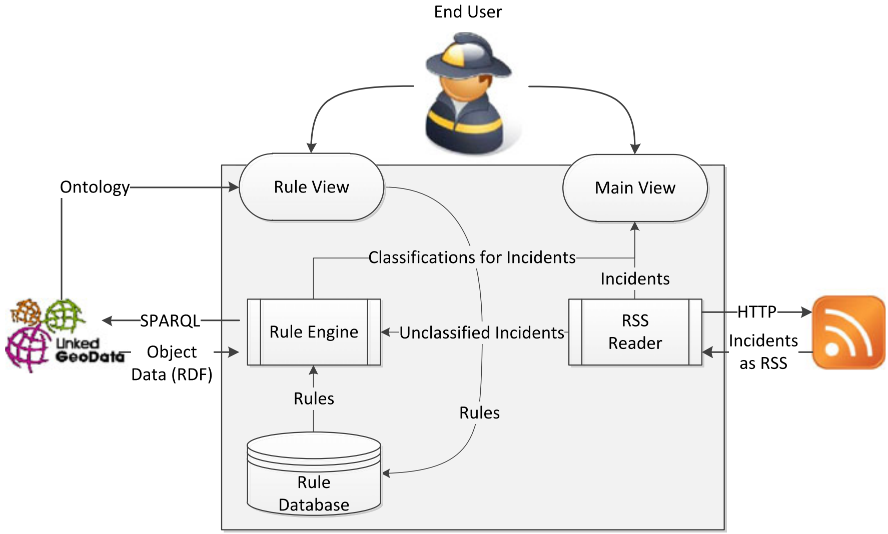



**Fig. 8.12** Prototype architecture

As the rule sets are relatively constant, editing the set of rules is something which is typically done once, and not at the time of an incident. During an emergency, the command staff will rather work with the main screen, which provides a clear situational picture to the emergency staff.

## 8.4.2 Architecture

Figure 8.12 shows the architecture of our prototype. The RSS reader reads incidents from an RSS source. The rule engine is responsible for classifying incidents based on rules entered by the users, using background knowledge from Linked Geo Data. A rule is a tuple of the form:

$$\langle incident\_type, radius, object\_types, class \rangle \tag{8.1}$$

The incident type is provided by the RSS schema of the incident source, the object types are classes from Linked Geo Data, and the classifications are an ordered set of degrees of severity. The above example would be formalized as follows:

$$\left\langle Fire, 50, \{\mathtt{lgdo:Fuel}, \mathtt{lgdo:Pipeline}\}, severe \right\rangle \tag{8.2}$$

For classifying incidents, the rule engine uses the maximum of all radii defined for formulating a SPARQL[29] query to retrieve objects nearby the incident. Each rule is then evaluated against the set of retrieved objects, and the maximum classification of

[29] http://www.w3.org/TR/rdf-sparql-query/.

all firing rules is used (for example, if one rule states that an incident is severe, while two more state that it is medium, the overall assessment is severe). Furthermore, those objects that made a rule fire are included in the incident information.

It is noteworthy that each rule serves two purposes:

1. Classifying incidents. An unordered list of incident messages is classified into different categories of severity. This allows the emergency staff to identify the most urgent messages.
2. Relevant information objects are picked from Linked Open Data. Since only the objects that made a rule fire are attached to the incident, rules help separating important information objects (such as gas stations and schools) from non-important ones (such as trash bins or traffic lights).

The outcome of the rule processing step is a set of classified incidents, augmented with relevant information objects. These processed incidents can then be used to provide intelligent views. For example, incidents may be filtered by severity, marked with different colors and/or symbols on a map, etc. By adding the corresponding objects responsible for making a rule fire, information on potentially harmed infrastructure (such as gas stations and pipelines) can also be provided to the end user, e.g., in a map view. Thus, the user may not only see that an incident is severe, but also get direct access to the objects that caused that rating, i.e., the name and the position of a specific gas station.

## 8.5 Conclusions and Future Perspectives

In this paper, we have shown a survey of mashup approaches for emergency management. The need to have timely, accurate, and complete information is a paramount requirement in emergency management. At the same time, it is hard to define the actual information sources in detail upfront, since emergencies are most often unforeseeable. Thus, the flexible and light-weight nature of mashups is a good fit for complementing IT landscapes for emergency management.

Existing mashup infrastructures make use of diverse information sources, including classical web sources as well as Web 2.0, sensor information and Linked Open Data. Citizen sensors also come into play when specific information is required which can only be delivered by a human on site. However, most mashups only use one or two types of information sources. Richer information infrastructures still need to be developed. They should leverage the information contained in the deep web together with timely Web 2.0 and sensor data as well as official government data, integrating all input into one clear situational picture. How to build complex information infrastructures like these, using sophisticated information filtering and aggregation mechanisms, while at the same time maintaining the light-weight paradigm of web mashups, is still subject to research.

While finding, aggregating, and filtering information is a topic already well covered by the mashup research in the domain, another crucial requirement in emergency management is information reliability. Some first works for addressing these issues, e.g., by crowd-sourcing, are currently emerging, but they are still rare. For establishing mashups in the emergency management domain, addressing information quality will be a crucial issue. Apart from crowd-sourcing, other possible approaches are weighting information sources or searching for mutual evidence in diverse information sources.

The information used in mashups is most often very timely, especially when using Web 2.0 information sources. However, presenting an analysis in real time is often very difficult. Recognizing a possible trend in hindsight is easy when looking at past data, but recognizing a beginning disruption is a harder problem. For example, the authors of *Twitris* admit that their spatio-temporal-thematic analysis lags a week behind. This shows that timely information processing is a topic of active research.

In summary, this chapter has shown that mashups can provide a clear value in the emergency management domain. As this is an active research area, future solutions will be even more powerful than the examples shown in this chapter, and eventually, mashups will be used in the future by emergency management staffs to help them make the right decisions.

## References

1. Abel F, Celik I, Houben G (2011) Leveraging the semantics of tweets for adaptive faceted search on Twitter. In: Aroyo L, Welty C, Alani H, Taylor J, Bernstein A, Kagal L, Noy N, Blomqvist E (eds) Proceedings of the 10th international conference on the semantic web, ISWC'11. Lecture notes in computer science, vol 7031. Springer, Berlin, pp 1–17
2. Abel F, Hauff C, Houben G (2012) Twitcident: fighting fire with information from social web streams. In: Proceedings of international conference on hypertext and social media, Milwaukee
3. Abel F, Hauff C, Stronkman R (2012) Semantics + filtering + search = Twitcident exploring information in social web streams categories and subject descriptors. In: International conference on hypertext and social media, Milwaukee
4. Ajmani V (2008) Los Angeles fire tweets. http://www.mibazaar.com/lafires.php
5. Ajmani V (2009) Iran election protests. http://www.mibazaar.com/irantweets.html
6. Ajmani V (2009). Swine flu tweets. http://www.mibazaar.com/swineflu.html
7. Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z (2007) DBpedia: a nucleus for a web of open data. In: Proceedings of the 6th international semantic web conference, ISWC, Busan, pp 722–735
8. Auer S, Lehmann J, Hellmann S (2009) LinkedGeoData—adding a spatial dimension to the web of data. In: International semantic web conference (ISWC)
9. Babitski G, Bergweiler S, Grebner O, Oberle D, Paulheim H, Probst F (2011) SoKNOS—using semantic technologies in disaster management software. In: The semantic web: research and applications (ESWC 2011), Part II, pp 183–197
10. Beaumont C (2009) New York plane crash: Twitter breaks the news, again. http://www.telegraph.co.uk/technology/twitter/4269765/New-York-plane-crash-Twitter-breaks-the-news-again.html
11. Bizer C (2009) The emerging web of linked data. IEEE Intell Syst 24(5):87–92
12. Bizer C, Jentzsch A, Cyganiak R (2001) State of the LOD cloud. http://www4.wiwiss.fu-berlin.de/lodcloud/state/
13. Brownstein J, Freifeld C (2007) Healthmap: the development of automated real-time internet surveillance for epidemic intelligence. Euro Surveill 12(48):3322–3324

14. Burke J, Estrin D, Hansen M, Parker A, Ramanathan N, Reddy S, Srivastava MB (2006) Participatory sensing. In: Workshop on world-sensor-web (WSW'06): mobile device centric sensor networks and applications, pp 117–134
15. Centre for Monitoring Election Violence (2008) Election day violence. http://cmev.wordpress.com/maps/
16. Clausen J (2012) Live Earthquake mashup. http://www.oe-files.de/gmaps/eqmashup.html
17. Collins A, Loftus E (1975) A spreading-activation theory of semantic processing. Psychol Rev 82(6):407–428
18. Drabek TE, Hoetmer GJ (1991) Emergency management: principles and practice for local government. International City Management Association, Washington
19. Endsley MR (1988) Design and evaluation for situation awareness enhancement. In: Aerospace systems: situation awareness in aircraft systems. Proceedings of the Human Factors Society annual meeting, vol 32. Human Factors and Ergonomics Society, Anaheim, pp 97–101
20. Federal Emergency Management Agency (2011). FEMA 2011–2014 strategic plan overview. http://www.fema.gov/about/2011_14_strategic_plan_overview.shtm
21. Fritz C, Kirschner C, Reker D, Wisplinghoff A (2010) Geospatial web mining for emergency management. In: GIScience 2010—extended abstracts, pp 1–5
22. Galway D (2009) Sensor Masher examples. http://sensormasher.deri.org/
23. Goolsby R (2009) Lifting elephants: Twitter and blogging in global perspective. In: Social computing and behavioral modeling. Springer, New York, pp 1–6
24. Gosier J (2011) Swift river 2011. http://www.slideshare.net/Ushahidi/swiftriver-2011-overview
25. Greater New Orleans Community Data Center (GNOCDC) (2009) Population and loss of children across the New Orleans metro area. http://www.gnocdc.org/repopulation
26. Heim P, Thom D (2011) SemSor: combining social and semantic web to support the analysis of emergency situations. In: Proceedings of the 2nd workshop on semantic models for adaptive interactive systems (SEMAIS)
27. Herald-Tribune (2012) IbisEye: your 2012 hurricane source. http://ibiseye.com/
28. Jadhav A, Wang W, Mutharaju R, Anantharam P (2009) Twitris: socially influenced browsing. In: Semantic web challenge 2009, demo at 8th international semantic web conference, Washington
29. Johnson D, Zagorecki A, Gelman JM, Comfort LK (2011) Improved situational awareness in emergency management through automated data analysis and modeling. J Homeland Secur Emerg Manag 8(1). doi:10.2202/1547-7355.1873
30. JRC European Commission (2012) MedISys. http://medusa.jrc.it/medisys
31. Kamenetz A (2010) Open source maps are helping the World Bank save lives in Haiti. http://www.fastcompany.com/1554072/open-source-developers-help-the-world-bank-save-lives-in-haiti
32. Keay A (2010) Ushahidi and Crowdmap: micro streaming as time-binding media. 3 PM J Digital Res Publ 1(2):116–125
33. Lavrenko V, Croft WB (2001) Relevance based language models. In: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR 01, pp 120–127
34. Le-Phuoc D, Parreira JX, Hausenblas M, Han Y, Hauswirth M (2010) Live linked open sensor database. In: Proceedings of the 6th international conference on semantic systems, I-Semantics'10, p 1
35. Le-Phuoc D, Quoc H, Parreira J (2011) The Linked Sensor Middleware—connecting the real world and the semantic web. In: Semantic web challenge 2011, ISWC 2011
36. Lui S, Palen L (2010) The new cartographers: crisis map mashups and the emergence of neogeographic practice. Cartogr Geogr Inf Sci 37(1):69–90
37. Mendes P, Passant A, Kapanipathi P (2010) Twarql: tapping into the wisdom of the crowd. In: Proceedings of the 6th international conference on semantic systems (I-Semantics 2010)

38. Milis K, De Walle B (2007) IT for corporate crisis management: findings from a survey in 6 different industries on management attention, intention and actual use. In: 40th annual Hawaii international conference on system sciences, HICSS07, p 24
39. National Association of Radio Distress-Signalling and Infocommunications (2012) RSOE EDIS. http://hisz.rsoe.hu/
40. Okolloh O (2008) Ushahidi, or 'testimony': Web 2.0 tools for crowdsourcing crisis information. Particip Learn Action 59(January):65–70
41. Puras JC, Iglesias CA (2009) Disasters 2.0: application of Web 2.0 technologies in emergency situations. In: Lundgren J, Jul S (eds) Proceedings of the 6th international ISCRAM conference, vol 37
42. Sakaki T, Okazaki M (2010) Earthquake shakes Twitter users: real-time event detection by social sensors. In: Proceedings of the 6th international ISCRAM conference, Gothenburg, pp 1–10
43. Schulz A, Paulheim H (2012) Combining government and linked open data in emergency management. In: AI mashup challenge 2012
44. Sequeda JF, Corcho O (2009) Linked Stream Data: a position paper. In: Taylor K, Ayyagari A, Roure DD (eds) International workshop on semantic sensor networks, SSN 2009, vol 522. CEUR, Washington, pp 148–157
45. Sheth A, Chen L (2011) Understanding events through analysis of social media. Tech rep, Kno.e.sis
46. Signorini A, Segre AM, Polgreen PM (2011) The use of Twitter to track levels of disease activity and public concern in the U.S. during the influenza A H1N1 pandemic. PLoS ONE 6(5):1–10. doi:10.1371/journal.pone.0019467
47. University of Fortaleza (2012) WikiCrimes—mapping crimes collaboratively. http://www.wikicrimes.org
48. Vieweg S, Hughes AL, Starbird K, Palen L (2010) Microblogging during two natural hazards events. In: Proceedings of the 28th international conference on human factors in computing systems, CHI'10. ACM, New York, pp 1079–1088
49. Zang N, Rosson M, Nasser V (2008) Mashups: who? what? why? In: CHI'08 extended abstracts on human factors in computing systems, pp 3171–3176

# Chapter 9
# Similarity Mashups for Recommendation

**Arturs Sosins and Martins Zviedris**

**Abstract** Recommendation systems are becoming a state of the art for web-based systems as they produce additional product exposure and customer satisfaction. The Semantic Web and mashups can improve recommendation systems and provide new ways for their creation. In the web it is possible to analyze product descriptions and to use Linked Data for characterizing the similarity of objects or of objects and user interests. In this chapter, we give a brief overview of existing technical approaches and tools for creating recommendation systems that can be used to create mashups as recommendation systems.

## 9.1 Introduction

Product and service recommendation systems are becoming a state of the art for vendor systems as they improve the opportunities for business. Product vendors get more exposures to their products, thus increasing the possibility to sell them, while consumers get an insight into engaging products based on their interests. Major drawbacks for recommendation systems are a lack of statistical information at the start of the system as most systems depend on available data. For newly opened e-shops or other sites like blogs it is an obstacle to successfully operating from the beginning because they just do not have enough statistical data to provide recommendations. Thus new recommendation approaches are developed to overcome this hurdle. These models try to use resources available on the Internet to determine similarity between different concepts and to enrich product content.

This chapter of the book will provide the reader with insight into the basics of similarity determining models, cover existing technical approaches and describe popular models for solving the similarity problem, starting from an introduction to the most popular techniques—collaborative filtering and content-based

A. Sosins (✉) · M. Zviedris
University of Latvia, Raina blv 19, Riga, Latvia
e-mail: ar2rsawseen@gmail.com

M. Zviedris
e-mail: Martins.Zviedris@lumii.lv

B. Endres-Niggemeyer (ed.), *Semantic Mashups*,
DOI 10.1007/978-3-642-36403-7_9, 

recommendations—and proceeding to mashup examples that use content-based recommendation techniques and combine multiple information sources available on the Internet to determine similarities between objects.

The second section introduces readers to the similarity problem—why one needs to determine similarities between objects, how the problem is commonly solved, and why existing solutions are not meant for everyone so that novel approaches are required.

The third section characterizes the problem domain where the proposed solutions can be applied. It discusses what type of problems current recommendation system have for specific domains and how these problems can be solved, and why it might be useful to use mashups of multiple information sources that combine in one recommendation system.

The fourth section describes specifics of the problem in more detail. The Tversky model is explained, showing how concept features relate to similarities and their usage in similarity models. Then we move to content-based recommendation systems, explaining the algorithm and the drawbacks. We describe how to determine similarity based on textual descriptions by extracting keywords that represent facts about objects, and comparing keyword sets, and other approaches that take advantage of information available in the semantic web, and using these linked data to establish connections between objects.

We also overlook a set of tools and methods that can be used to implement similarity approaches that are based on the semantic web. One can also gain valuable ideas from a tool overview on how to improve the existing approaches or combine them in a novel way to create a better mashup to determine similarities between objects.

In the end, a couple of mashup examples are presented. These mashups are

- dbrec, which finds similarities between music groups based on entries in the DBpedia
- Gamedipper, which presents similar video games based on their textual description from game reviews
- BlogAlike service, which generates lists of similar articles of the same blog
- Zemanta, used to enrich blog content with external media and related links

This section describes how they work and what their information sources are, and in case of Gamedipper, there is some more technical information of the implementation and evaluation.

## 9.2 Background

With all available products and enormous information flow, it is becoming hard to find things that suit your interests, which is why many websites are using recommendation systems to adapt to users' interests. Systems of this kind help website customers to find and choose things that they may deem necessary or desirable.

There are a few approaches for developing recommendation systems [1], but the two main ones are collaborative filtering and content-based recommendations.

The first approach is based on collaborative filtering [2] to determine possible similarity between objects. Basically it compares your interest (usually expressed as purchase or ratings) in products with other user interests, finds the most similar one, based on the quantity of similarities between users, and it offers products that this similar user has expressed interest in.

To find the most similar user, we need to compare all products they expressed their interest in. For example, let us say that we are comparing user ratings for different products. In that case, the first thing to do is to calculate the Euclidean distance between each pair of users. But we also need to take differences in consideration. For example, if one user has rated 50 products and another one 30 products, but the users only match in five products, that should definitely affect the similarity between them. That is why we multiply the result with the ratio between the amount of common products and the maximal product quantity that both users can match in, which in our case was 30 products.

From a product recommendation point of view, this approach may be dependent on buying trends, popularity of products and other none-similarity-related factors. Thus it recommends similar actions, for example, following trends, not similar objects. Whether it is good or not depends on the purpose of the system and the system's users' needs. For example, if your goal is to increase sales, then you probably should recommend trending products.

We should also note that the collaborative filtering strongly depends on the amount of statistical data. Thus the initial threshold of the approach is embarrassing for smaller websites that do not have enough user data to produce good enough recommendations. As it is sometimes the chicken or egg dilemma that with good recommendations the user base grows faster, but with the small existing user base and the statistics that it produces, it is impossible to provide good recommendations, some other ways of recommendation approaches are needed to successfully overcome the threshold.

As you may be aware, large scale systems with high volumes of user traffics each day usually have good recommendations for you as user. These kinds of systems are eBay, Amazon, Netflix, Pandora, Last.fm and even StumbleUpon. And even they are looking for the ways to increase the quality of provided results. For example, Netflix offers a 1 million dollar contest. But have you noticed that small local on-line shops do not have any recommendations, or if they do, it is usually a recommendation from the same product category? They just do not have the resources and statistics needed to create recommendations with plausible results.

Another approach involves more information about the object itself, the object's attributes and even textual descriptions. Provided that a user likes the current object, it is possible to find similar ones in the hope that they will be valuable to the user [3]. As opposed to collaborative filtering, where we are comparing different user models, in content-based recommendation systems, we need to establish links between user and content (whether the user likes it) and then compare objects with each other, in order to find the most similar ones to the object that the user liked.

These similarities are found by matching feature sets of different objects, for example by determining how many attributes that one object has, and apply them to other objects also matching the attribute values. Of course, this raises some problems when comparing completely different types of objects that do not have any attribute types in common. For example, how would you compare a movie to a book? This problem does not persist with collaborative filtering, because user interest in a book or a movie may be expressed in the same way. But in content-based recommendation systems the most popular solution for comparing different types of objects is to use their textual descriptions. Which leads us to another problem—how to compare textual information about objects. Usually it is done by generating meaningful keywords, which represent facts about objects from given text and use them as feature sets in the matching process. Like this it is possible to compare products of different types and sets of properties only by their textual descriptions.

Results of comparing keyword sets are probabilistic and strongly depend on description quality. Thus it is important to gather only suitable descriptions and to develop a sophisticated keyword generation processes. A solution for increasing the quality of results is to gather reviews only from trusted sources that have already produced them and to take more factual reports rather than subjective reports on likes and dislikes. Finding trusted sources depends on the niche you are working on and is usually done manually, by exploring the niche and finding the leading portals with catalogs and reviews about a specific subject.

Since both approaches (user interaction statistics for collaborative filtering and quality content for content-based recommendations) have drawbacks, developers are constantly looking for alternatives. One of the directions is to experiment with hybrid systems [1] that use both collaborative filtering and content-based recommendations. Another direction is to use different sources of information already available on the Internet and combine them in mashups [4], which is most suitable for content-based recommendations. Thus the semantic web can help to tackle the problem as it provides relations between different objects that can be used to determine the similarity between them. This chapter will mostly concentrate on hybrid systems that try to build mashups and it will provide insights into some of the approaches one can use to build a mashup similarity recommendation engine for different concepts.

## 9.3 The Domain of Recommendation Systems

Determination of similarities between two different concepts is a great addition to any system that provides different opportunities to its users. Showing items that are similar to ones that a user liked will positively increase overall user experience and save users' time on searching and deciding for specific items, which is exactly what every system wants. The main domain of recommendation systems, specifically content-based ones, are the websites rich with information about each item, and not enough resources to generate required interaction statistics, or without opportunities for users to express their interaction.

The first example of these websites are blogs or news portals. Portals of this kind are usually without options to identify users and their actions across different viewing sessions. The only reliable interaction fact for a specific item is that this user is reading this article in this very moment. So it would make sense to generate lists of similar articles based on the current articles and present these lists to the user. There is a great chance that if a user liked the current article he or she would also read similar ones offered by system, making the user visit one more page and increase visiting statistics, and view more ads or improve whatever is the purpose or the monetization model of the current website. Also we should note that news items are mostly factual. Thus generated keywords will not express subjective likes or dislikes that vary in news articles. With blog situations this is a bit different but still appropriate as blogs mostly express likes or dislikes in a consistent way. Thus generated keywords will express the same feeling.

The second example are on-line stores. There are ways to record user interactions, through buying histories and provided ratings, which can easily be used by collaborative filtering, but in many cases it might not be enough to rely only on them. Content-based recommendation can be provided from the first day an on-line store opens, whereas collaborative filtering would require some time for statistical data to gather and a lot of customers to get data from.

One web application may have many potential domains for recommendation systems, for example if it involves search engines implemented in systems as blogs, news portals or on-line stores. By providing intelligent recommendations for searching terms or results, it is possible to save user time and frustration during searching for information he or she needs. These recommendations can be achieved by comparing the importance of provided keywords with stored own ones and returning results ordered by sum of importance of all found keywords.

The last example we want to mention are the information enrichment capabilities. When writing blog posts or news articles, it might be useful to include links to similar articles, to similar topics for both your own websites and other bloggers or news portals. Maybe one could even advance it to other media types as images and videos, or simply include generated tags extracted from your content. Because after all, visitors came to your website to look for information—the more related information you provide, the more returning visitors you will have.

## 9.4 Mashups of the Domain

This section describes technical details and background knowledge you would need if you want to create a mashup recommendation system. It also presents a set of tools that the authors have used and that might be helpful. In the end of the section there are some issues and open points presented for further research in this field.

### *9.4.1 Characterization*

The power of mashups lies in an ability to combine multiple resources. Exploiting this ability, we can gather information from multiple sources that provides valuable facts about specific objects, thus providing a powerful way to implement content recommendation systems using mashups. This section discusses several ways of gathering information and how to apply it to existing similarity models, in order to create a content-based mashup recommendation system.

### *9.4.2 Purpose*

The whole purpose of similarities is to help an end user to find related information faster and easier, by providing lists of similar items a user may like, or categorizing search results in the same manner.

The purpose of the similarity mashups is quite the same, if not greater. Some mashups that acquire information from multiple sources, not only can generate lists of similar items, but also provide it as a service to other consumers or mashups. Possibly information is enriched so that users can understand it faster.

### *9.4.3 Technical Background*

This subsection describes some approaches to how one can evaluate similarities between different concepts. Choosing a usable approach may depend on numerous factors, like type of objects that will be used, available information and resources, etc. A final choice may not necessarily fall only into one of the approaches. It is possible to combine them so that they will complement each other giving even better results.

#### 9.4.3.1 Information Sources

This section describes how to find proper information sources on your specific topic, and how to retrieve the needed information, once it is found. The easiest way to retrieve information is that your information source provides an API through different protocols as SOAP or REST, thus presenting all available content usually in JSON or XML formats. Otherwise it gets harder to retrieve information, because one would need to scrape the web content and then properly parse it to retrieve the desired information. This can usually be done through regular expression parsing or detecting areas of change of different pages (processing semi-structured content as HTML), or using different languages for specific formats as Xquery for XML and Sparql for

RDF (processing structured content). More about these options is described later in this section.

Content-based recommendations rely on information about items that we want to recommend. The more information we can gather, the better results we can achieve. Thus it is important to combine different information sources and to know whether information quality is acceptable. Thus, we must select information types and sources really carefully. Finding proper sources and gathering information is one of the hardest parts of creating content-based recommendation systems.

For example, if we would be interested in movies then the best source probably would be IMDB.com using their API to get properties and descriptions of different films. If you are looking for more unstructured information, then in these types of items, reviews are usually good sources of information, so that, for example, Metacritics.com could also be considered as a good quality resource. Finding suitable information sources for a field of interest is not easy because many fields lack catalogs of information sources. Thus it depends on the human factor to find suitable sources.

Once acceptable information sources have been found, it is useful to check for their web services as services allow acquiring information easier. Web services basically are an Application Programming Interface (API) that usually allows browsing and gathering available information in an automatic manner. The APIs allow a combination of multiple services into a single, novel application that can be called a mashup.

Technical ways of information gathering from APIs vary as there are many different protocols and data formats. But when used in a context of web applications, a simple set is most popular: Hypertext Transfer Protocol (HTTP) request messages with defined parameters and response messages usually in Extensible Markup Language (XML) or JavaScript Object Notation (JSON) format.

Popular standards to define Web Services are SOAP, that is, an XML-defined messaging framework and has been designed to be independent of any particular programming model and other implementation specific semantics [5], and Representational State Transfer (REST), which is a server/client architecture-based standard where requests and responses are built around the state representations of resources [6]. We should remark that a recent trend shows[1] that web services are moving away from the Simple Object Access Protocol-based services towards the more direct Representational State Transfer style communication [7].

One approach for automatic information gathering that is usually underestimated is content syndication. The most popular content syndication formats are Really Simple Syndication (RSS) and Atom. Both are XML-based documents, usually called feeds, with discrete content items and web links to the source of the content. It is a common practice for blogs and news portals to provide such automatically generated feeds for recently added content.

[1] ProgrammableWeb: REST vs. SOAP. http://blog.programmableweb.com/2010/06/09/new-job-requirement-experience-building-restful-apis/.

There are two reasons why feeds alone cannot be used for gathering information:

- They provide only short overall information about items.
- They provide only recent content, not all available information.

But content syndications of this type are useful for information extraction. They make discovering new content on the website easier and provide canonical URLs to content that can be used to scrape content from websites, so one does not have to go through all website pages and index their contents to find usable information.

If there is no web service available for a specific source of information, it is an option to scrape its web content. Web content scraping is a process meant to automatically collect information from the websites, which usually involves extracting information from unstructured or half-structured data in HTML format and to transform it to structured data that can be analyzed. The data are usually retrieved from a website as text content over Hypertext Transfer Protocol (HTTP). In many environments, as Java or .NET it is done through environment-implemented HTTP clients, but there are also other libraries, for example, cURL and wget, which are available for most of the platforms, starting from command line interface up to PHP, Ruby, Python, etc.

The web scraping approach can be a fairly questionable process, as is can be perceived as stealing information from a website. Although copying information from a website for personal use is not illegal, some systems may have preventions implemented against automated scraping tools not only for reasons as unknown intentions with collected data, but also because automated crawling may use up their server resources. As an example, there is a case where eBay won an injunction stopping Bidder's Edge from automatically crawling the eBay site to generate auction comparison listings, because the robotic crawler used eBay system resources [8]. So respect your information sources, check if content usage applies with their terms and preserve their resources when collecting their data automatically, for example, by using a timeout between requests or limiting the number of requests per specified time units.

The main problem with content scraping from websites is to properly parse it. By standard web practices all text should be included in proper tags like a paragraph tag. But in practice, not all developers and even frameworks follow best practices, so that one needs to parse all tags to find all valuable information.

There are different approaches for parsing semi-structured HTML to get information. One of the simplest includes a regular expression matching to extract information between specified HTML tags, or on the contrary to ignore some sets of tags. Other approaches include DOM parsing when content is not only retrieved as a string, but also processed by some browser engine, including client-side scripts, forming an HTML document object model (DOM) tree that can be traversed [9]. There are also some semi-structured data query languages, for example, XQuery—a functional programming language that is designed to query collections of XML data, and Hypertext Query Language (HTQL)—a language for the querying and transformation of HTML [10].

Another problem with semi-structured HTML parsing is that websites may include irrelevant information. Each website can have a menu, a footer with a description of the website, headers and other parts of the website that do not provide any information about the specific topic, making automatic gathering of related content a really hard job.

There have been attempts to solve this problem, for example, by including Resource Description Framework in attributes (RDFa), which allows using Friend of a Friend (FOAF), Dublin Core (DC) and other properties for describing documents. Another solution is to use Microformat's classes as HTML attributes in website code. These formats allow defining different types of information as events, contact information, resumes, media, news, feeds, etc. Using these solutions, automatic software would not only be able to easily extract information, but also to add some semantic relations to it.

Another approach is the new HTML5 standard with its so called semantic tags. They allow, for instance, distinguishing main content from headers, footers and side menus. Right now there are 10 such elements with semantic purpose [11]:

- article—Self contained syndicatable or reusable composition
- aside—Sidebar for tangentially related content
- figcaption—Caption for figure
- figure—Figure with optional caption
- footer—Footer for a page or section
- header—Introductory or navigational aids for a page or section
- hgroup—Heading group (on or more of h1, h2, h3, h4, etc.)
- mark—Highlight (phrase or paragraph)
- nav—Section with navigational links
- section—Generic document or application section

The main disadvantage of all described solutions is the amount of work needed to implement any of them. If there is no reason for a system to share its information, implementing RDFa or Microformats would be an additional and needless work for developers. And of course, what to do with existing websites that are already published without any of these solutions?

There is another approach available. When comparing pages of the same website, it is possible to detect areas where textual contents change. For example, a navigational menu or copyright notice in the footer probably would stay at the same place on all web pages of one website. An approach exploiting this steadiness/change opposition is called Wrapper Induction System. It usually consists of Template Detection and Wrapper Generation [12]. First a system trains detection, by grouping training pages into classes, for example, using a URL and assuming that subdirectories of one URL have the same template in common [13], or by comparing similarity among page representations [12]. These groups are processed by some induction module, which generates wrappers for each found class. These wrappers than can be used to extract desired information from web pages.

### 9.4.3.2 Text Processing

Besides comparing structured data as item properties, content-based recommendation systems additionally use unstructured information as description texts to establish similarities. Text may contain different facts about objects that cannot be obtained through properties. But a problem is to retrieve and process those facts to a point where we can compare them between different objects. For this purpose we can extract descriptive keywords from the text, which represent the facts about the objects, and then use them as object properties on which similarity will be based.

#### 9.4.3.2.1 Keyword Extraction

Keyword extraction processes may use numerous approaches to analyze text and generate candidate keywords: starting from statistical approaches [14, 15], to heuristic-based analysis [16]. Each approach has its own advantages and disadvantages, and usage of each one may depend on a domain, available information, available resources or languages used in text. The most usual method is to order words by their frequency in a text. But this also means that the most common words will be more likely make the top keywords. To avoid this, it is a usual practice to filter out common and other unwanted words from text before text analysis. These common words or unwanted words are called stop words, and lists containing them in are usually called stop lists. Filtering text using them will make sure that common or unwanted words will not appear in a top keyword candidate list, leaving place for more important ones.

There are also more sophisticated statistical approaches, for example, keyword extraction using co-occurrence statistical information of words [15], where frequent words are extracted first, and then we analyze co-occurrence of frequent words and all words in the same sentence. And if the probability distribution for co-occurrence is biased to a particular subset of frequent words, then these words may be considered significant. There are also several machine learning approaches, which learn to extract keywords based on provided examples of text and a result set. These kinds of approaches usually use Genetic algorithms or Bayes-based classification methods. The most notable representatives are GenEx [17] and Kea [18] algorithms.

The last approach we would like to discuss is a keyword extraction based on heuristics. It is usually an attempt to extract more semantically related data [4, 16]. For example, proper nouns might be more significant to certain objects than other nouns. Proper nouns might define characters, places and related entities. Thus by using proper nouns as keywords, the significance of matching them in different objects is greater than the significance of matching regular nouns.

The way of extracting proper nouns usually depends on the grammar of a language. The most common approach is to use multiple rules defining proper nouns and extract words abiding these rules. For example, using the first letter as a sign, if it is in uppercase and a word is not in the beginning of the sentence, then this word should be a proper noun. Of course there could be many exceptions and defined rules should be more sophisticated than that.

However, there are grammars in which this method will not work. You will need to find other approaches specific to the language. For example, in German language all nouns are in uppercase, but it is still possible to extract proper nouns because of articles. In formal German texts, there should be definite or indefinite articles before nouns, but they are not used before proper nouns. So if a word's first letter is in uppercase, if the word is in the middle of a sentence and none of definite or indefinite articles are before it, then this word should be a proper noun [19].

#### 9.4.3.2.2 Keyword Processing

When statistically processing all keywords from text, there is a need to group similar ones to increase their potential salience in text. For example, if the same word is used in different forms, different time, it would count as different words, although the meaning is the same. For that purpose, it is a common practice to use a stemming algorithm to reduce words to their stem, thus matching different forms of the same word. And when it comes to keyword processing it is better to allow false positives results [17], so you should choose more aggressive stemming algorithms. A great representative of this type of algorithms is the Lovins algorithm [20].

Another scenario are the synonyms. They are completely different looking words with same meaning. This makes it significant and hard to identify and to group them [21]. There are synonym dictionaries available especially for this purpose.[2]

#### 9.4.3.2.3 Requirements for Text

As similarity is based on keywords from textual description, these texts should meet certain requirements to ensure better results of similarity classification [4].

Only qualitative facts about subjects should be mentioned in a text. For example, describing subject A that it is red and round is a qualitative fact. But describing what subject A is not (not square and not blue), is not a qualitative fact, and it will probably provide the keyword square and blue as describing keywords for subject A. Then by comparing sets of keywords for each subject, the similarities with squared and blue objects might be established, because they have same keywords.

Another important aspect is that a formal language is used in text. When comparing two different texts by keywords, a formal language is like a protocol that ensures that if both texts mention same thing, then their respective words will be similar. There should be no unnecessary abbreviations, jargon or fictitious words, which usually appear in texts found on the Internet.

---

[2]WordNet: http://wordnet.princeton.edu/. Dictionary.com: http://developer.dictionary.com/. Abbreviations.com: http://www.abbreviations.com/api.asp.

And third requirements for texts are that they should be grammatically correct. As in a formal language, if there are the same words in two different texts, and one of them is mistyped, they will not count as similar ones and will lose their importance.

#### 9.4.3.3 Similarity Basics—Tversky Model

In 1977 Tversky described a model in which objects are represented as collections of features and similarity is described as a feature matching process [22]. The more features objects have in common, the more similar they are. Of course the amount of distinctive features is also important. That is why the similarity between objects is expressed as the combination of objects' common and distinctive features. The Tversky model is a basic concept of similarity. Many similarity determination approaches that we will discuss use it in one way or another, for example, by comparing sets of features or relations between different objects to determine how different they really are.

For the sake of simplicity, let us assume that compared objects are products and their features are product properties. We can try to determine two product similarities by comparing their common and distinctive properties. But in order to get proper results, we will need something more sophisticated than that.

Firstly there should be a vast amount of properties to compare with.

Secondly each property should have been weighted. That would define the importance of a property in the determination of object similarity. For example, genre, director and scriptwriter are more important in determining similar movies than the length of the movie, its budget or production year.

Thirdly, these properties should describe an object in a way that would be helpful for similarity determination. For example, if one wants to compare songs then it should be the meaning of the song, the melody that should be compared. For movies, it is the story line, the plot that contains the needed information. Usually these kinds of properties are hard to compare.

#### 9.4.3.4 Classifying Similarities

Once information is gathered and structured, it is now possible to establish similarities between different items. There are a couple of ways to do that. For example, it is possible to divide items into groups and repeat the process until each group contains only instances of one single class. This kind of approach is called Decision trees. Partitioning is performed by matching the items' individual features in their feature sets, for example, the presence or absence of specific keywords [23]. One example of this approach would be an ID3 algorithm, which generates a reasonable good decision tree without much computation [24].

Another approach uses the vector space model to represent similarities between items, by separating all the instances in the multi-dimensional space and using multi-dimensional weight vectors to establish similarities between items. Algorithms using this approach are called linear classifiers [25].

Classifiers of a next type compare sets of features for each item and use a similarity function that produces a "distance" between specified items, thus finding "nearest neighbors" what would also mean finding the most similar items. The similarity function used in nearest neighbor approaches is usually an Euclidean distance [26], but also other non-Euclidean measures are observed, mostly for high-dimensional models, as Value Difference Metric [27] and Cosine Similarity measure [28].

There are also probabilistic approaches with the Naïve Bayesian classifier as the most significant representative, which is adopted in many works and acknowledged as one of the most well-performing text classification algorithms. The main advantages of the Naïve Bayesian classifier are that it is quite simple to implement, it is fast and highly scalable, and it works great on a smaller training sets; and even when the independence assumption does not hold, the Naïve Bayesian classifier usually performs surprisingly well [29–32].

The last approach that we should mention is based on the consideration of previous results returned by the system, by allowing users to provide a feedback, for example, by rating the results. This approach is called Relevance Feedback. One of the most popular representatives is Rocchio's Algorithm. It is widely used in the vector space model. It uses the nearest neighbor metric with differently weighted vector prototypes of relevant and non-relevant items [33].

#### 9.4.3.5 Linked Data

A semantic approach to the information available on the Internet gave us an opportunity to extend the Tversky model and to analyze not only object features, but also relations between objects, which can be coded in Resource Description Framework (RDF) form [34].

Basically the Tversky model describes relations between different objects. These relations can contain not only equal properties, but also provide additional information that cannot be obtained using properties. For example, we have two different movies. They have one equal property—the director, which is also a relation between these movies. But they might also have a relation that could not be a property, for example, they are representing a certain historical event. If we compare these two movies, then it is more significant that they are representing a certain equal historical event than that they have the same director.

Relations may contain additional information, but how should we handle them? Well, similarly to how we could handle properties in the Tversky model. We need a factor of how important this relation is in comparing two objects. Additionally, you can go further and compare objects and properties which relate to objects, and properties that relate to comparable objects. In other words, it is possible to compare direct relations or indirect ones that can give us much deeper insight into how closely two objects are related. We can assign coefficients based on the depth level of relation. For example, if one movie is directly related to a historical event and another movie is indirectly related through another object, then the significance of the first movie relation is much greater than that of the second movie relation. Based on relation values or relation quantities, we can calculate Linked Data Semantic Distance (LDSD) [34, 35].

## 9.4.4 Tools

This subsection describes some well-known tools for different steps of creating recommendation systems, for example for text processing, keyword generation and information sources harvesting. The authors have used all tools themselves and can confirm that they might be helpful to others trying to complete a similar task. Note that this list is not complete and there are a lot more tools not mentioned here.

### 9.4.4.1 OpenCalais

OpenCalais [36] is a tool to analyze text, using natural language processing and machine learning algorithms. It extracts entities from text and displays semantic relations between these entities. It does not only extract these entities, but it also classifies them as, for example, place, person, movie, etc. Currently OpenCalais supports texts in English, Spanish and French.

You can use OpenCalais free of charge, through a Web Service using protocols as REST or SOAP.

### 9.4.4.2 DBpedia

DBpedia is a great source of structured quality information. It is a semantic version of Wikipedia, which allows to perform sophisticated queries against information available in Wikipedia and to link other datasets on the Web to Wikipedia data. Querying uses Sparql—a query language that was created specifically to retrieve and manipulate data in RDF format. Information is accessible through Sparql endpoints using lists of available datasets. DBpedia is a very useful information source for linked data.

## 9.4.5 Dictionaries

Dictionaries are useful for forming stop word lists and grouping words into synonyms. This can help you in keyword generation and assigning importance to each specific keyword. There are many dictionaries with public web services available to query for mashups. Here a couple of them as example:

- WordNet[3] is a large lexical database of English nouns, verbs, adjectives and adverbs, which are grouped into sets of cognitive synonyms. WordNet is one of the most popular dictionaries.
- Dictionary.com[4] provides a wide spectrum of options to search for words, thesaurus, quotes and references.
- Abbreviations.com[5] also provides a wide range of APIs to use, as for abbreviations, phrases, synonyms, quotes, definitions and even rhymes.

### *9.4.6 Issues, Open Points*

In this section, we will discuss some issues of creating mashup recommendation systems that require your attention, and open points for further studies, and we shall provide references to research on some of the specific points.

As mentioned before, there is a lot of research on combining different classifying approaches to get better results, thus creating hybrid recommender systems [1].

Still there are some unsolved issues that you should be aware of. For example, issues regarding information sources. The first thing that should be mentioned is the legal issue of content ownership. You must check how obtained content may be used and comply with the provided terms, no matter whether you obtained the content via an API or via web scraping: can you store a copy of content, can you republish it, etc. Another aspect to consider is that you will not use up server resources of your information source. Most APIs have request call limits that you should check and comply with. You may also consider caching data, by storing it on your server, if it complies with content usability terms.

Another unsolved issue is how to discover new content from websites or APIs. You can use RSS feeds to some extent on websites, but APIs usually do not provide a way to discover new content, and your only option is basically walking through all the content, which again uses resources, both yours and of other servers. Thus there is a need for flagging newly added information from the information owners' point of view or to implement an efficient way to discover changes and new content on websites [37].

There are also unsolved problems of extracting keywords and, for example, finding proper nouns in different languages, especially logographic ones, as Chinese, Japanese, Korean, etc., although there has been some research in that direction [38–40]. Identifying the context of homographs (words with the same spelling, but different meaning) is another common problem. When extracting keywords, it is important to know the context of the homograph, for example, whether the word "race" relates to "car racing" or "ethnicity".

---

[3] WordNet: http://wordnet.princeton.edu/.

[4] Dictionary.com: http://developer.dictionary.com/.

[5] Abbreviations.com: http://www.abbreviations.com/api.asp.

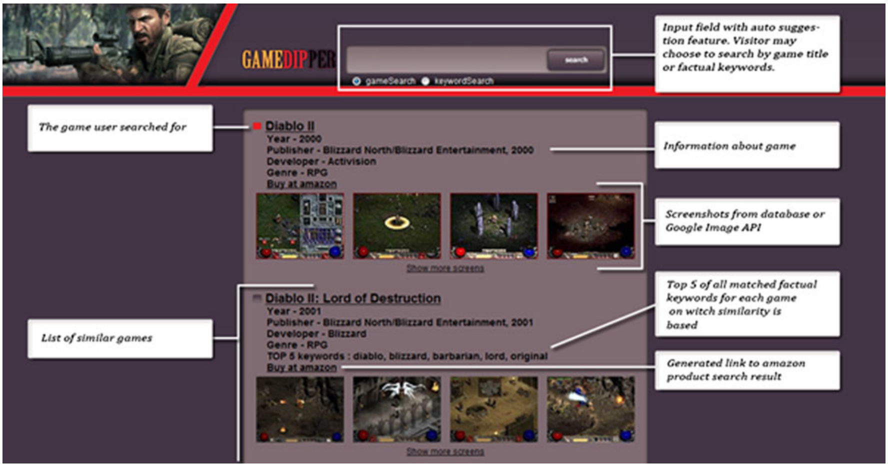



**Fig. 9.1** Screenshot of Gamedipper.com website, March 2012

Another open issue is adding weights to similarity decisions. For example, linked data have different relations and each should be weighted to get better results.

## 9.5 Demo Mashup

### *9.5.1 GameDipper*

GameDipper (http://gamedipper.com/) [4] is a search engine for similar video games. Similarity of video games is established by analyzing video game reviews from different game review portals. An interface screenshot of GameDipper's search engine is provided in Fig. 9.1.

At the time of writing, GameDipper contains more than 20,000 video games, with more than 26,000 video game reviews and more than 50,000 extracted keywords.

As game reviews usually describe a game from different points of view, for example game play, game graphics, sounds, story line and the overall atmosphere, it is a great source for finding facts about a game of interest.

The list of sources for reviews contains Video game database APIs and the most popular video game review portals [41]:

- DBpedia SPARQL Endpoint
- GamerPro API
- GamesRadar API
- gamerDNA API
- GiantBomb API
- Goozex API
- GameSpot RSS Feed
- GameZone RSS Feed

- IGN RSS Feed
- GameSpy RSS Feed

Game reviews are parsed and keywords describing facts about games are extracted. The GameDipper parser marks proper nouns in text and increases their salience coefficient. This heuristic is based on the fact that proper nouns usually represent other important entities, like characters, places or even other video games that authors mentioned as similar ones.

The second heuristic is based on cross-referencing extracted keywords with multiple reviews of the same game, thus decreasing the probability of unrelated keywords to be attached to a specified game.

Since this is a heuristic-based approach, it is always interesting to analyze and assess the results. For example, if game A is similar to game B, it would be logical that game B is similar to game A. Since in our case similarity is not a strict relation, we might have a situation where game A might be most similar to game B, but game B might be most similar to game C, if game A's keyword set does not completely match game B's keyword set. That is why we will analyze the position of games in the top 10 similar games.

In Table 9.1, you can see the result of an experiment where we take 4,000 randomly chosen games and for every game A, we will find a most similar game B, and then we check game's A position in the top 10 similar games for game B. The percentage shows how many times game A was in each top position.

As you see more than 60 % of the most similar games are symmetrical (are in top 10) and the percentage is decreasing from top 1 to 10. The results have an overall trend to symmetry, but it is not perfect. This could be explained by the fact that extracted keyword sets from texts are not symmetrical, authors mention different facts about similar games. If in game A's review game B is mentioned as a similar one, this does not mean that in game B's review, game A will be mentioned, as mostly newer games are related to older ones, but not in other direction.

**Table 9.1** Symmetry of similarity—finding the most similar game B for existing game A, and checking which games that are most similar to game B, and where game A is positioned

| Position | Percentage of all results |
|---|---|
| 1 | 26 % |
| 2 | 11 % |
| 3 | 7 % |
| 4 | 5 % |
| 5 | 4 % |
| 6 | 3 % |
| 7 | 2.4 % |
| 8 | 1.6 % |
| 9 | 1.4 % |
| 10 | 1.3 % |
| Not in top 10 | 37.3 % |

**Table 9.2** Transitivity of similarity—finding the most similar game B for existing game A, and finding the most similar game C for game B, and checking how games A and C relate, in which position game A is for similarity with game C

| Position | Percentage of all results |
|---|---|
| 1 | 10.9 % |
| 2 | 9.1 % |
| 3 | 6.4 % |
| 4 | 5 % |
| 5 | 4.3 % |
| 6 | 3.5 % |
| 7 | 3 % |
| 8 | 2.2 % |
| 9 | 1.9 % |
| 10 | 1.7 % |
| Not in top 10 | 52 % |

Let us expand this relation to three games. If game A is similar to game B, and game B is similar to game C, than there should be some sort of similarity between game A and C, because they share something in common—game B.

So again, we will take 4,000 games and for each game A, we will find the most similar game B, for this game B, we will find the most similar game C and then we will check what is game A's position in game C's top 10 most similar games. The result of this experiment is presented in Table 9.2.

Results show that almost half of all tested games appear in the top 10, meaning there is a correlation between all three tested games. Not all triplets correlate, mostly because of the same reason: symmetry is not perfect—different facts are mentioned about similar games.

The last step to assess the similarity searching engine is to compare it with other systems with the same or different approaches. As our theme is video games, we select systems that also offer similarity of video games.

We will compare GameDipper and more systems: GameSpot, gamerDNA, Giant-Bomb, Amazon.

Amazon uses product recommendation systems which are based on the user's interest in products, that is, on user buying habits, which games people buy with other games, etc.

GiantBomb is a Wiki for video games, where one can change Wiki contents. So users add games that they think are similar to a specified one. Thus the Wiki represents the overall opinion of this video gamer community.

GamerDNA is a social portal for video game players. Users can create own lists of games—games they like, currently play, etc. It is possible to compare these.

Unfortunately there is no information provided on how GameSpot determines similarity between games, and there is no way for users to provide feedback. We can only assume that similarity is determined by website administration.

**Table 9.3** GameDipper comparison—how many GameDipper-recommended games appear in other systems' top 10 recommendations

| Video game | GameSpot | gamerDNA | GiantBomb | Amazon |
|---|---|---|---|---|
| BioShock | 2 | 3 | 3 | 2 |
| Crysis | 0 | 1 | 2 | 2 |
| Deus Ex | 1 | 1 | 2 | 0 |
| Devil May Cry 4 | 0 | 0 | 1 | 0 |
| Diablo 2 | 1 | 2 | 6 | 1 |
| Doom 3 | 0 | 1 | 0 | 1 |
| FIFA World Cup 06 | 7 | 0 | 2 | 1 |
| Gothic 4 | 2 | 1 | 0 | 0 |
| GTA IV | 3 | 3 | 1 | 3 |
| Half-Life 2 | 1 | 1 | 1 | 1 |
| Halo 2 | 0 | 2 | 1 | 1 |
| Left 4 Dead 2 | 1 | 2 | 1 | 2 |
| NFS ProStreet | 2 | 2 | 2 | 3 |
| PoP: The Sands of Time | 0 | 2 | 1 | 0 |
| Runaway: A Road Adventure | 0 | 0 | 0 | 0 |
| Sid Meier's Civilization IV | 2 | 3 | 4 | 4 |
| Syberia II | 3 | 2 | 2 | 2 |
| The Sims 2: Apartment Life | 4 | 4 | | 3 |
| Tom Clancy's Splinter Cell | 0 | 4 | 1 | 4 |
| WOW: Wrath of the Lich King | 4 | 2 | 5 | 2 |
| Average | 17 % | 18 % | 18 % | 16 % |

Because of a lacking APIs and different game naming conventions, there is currently no way of automating the comparing task, so that we compared results manually. We have selected 20 popular games which appear in all recommendation systems. For each game we have selected the top 10 most similar games in each system.

Table 9.3 displays how many GameDipper-recommended games appear in the top 10 positions in other system recommendations.

As one can see, there is quite a big difference in the results. Less than a fifth of the results match. With a slight difference, GameDipper results seem to be most similar to GamerDNA and GiantBomb results created by masses of users.

Let us check if recommendation systems overall results match with each other. To do that, for each system we will check for each game how many times it appeared in other system top 10 recommendations. The result of this experiment is presented in Table 9.4.

Results show that GameDipper has a bit better results than other recommendation systems. Our chosen approach has some perspective.

**Table 9.4** All system comparison

| Game | GameDipper | GameSpot | gamerDNA | GiantBomb | Amazon |
|---|---|---|---|---|---|
| BioShock | 7 | 2 | 4 | 7 | 3 |
| Crysis | 4 | 2 | 1 | 4 | 3 |
| Deus Ex | 2 | 2 | 6 | 8 | 4 |
| Devil May Cry 4 | 1 | 1 | 1 | 1 | 1 |
| Diablo 2 | 7 | 2 | 5 | 8 | 3 |
| Doom 3 | 1 | 2 | 1 | 5 | 4 |
| FIFA World Cup 06 | 7 | 8 | 0 | 4 | 3 |
| Gothic 4 | 6 | 2 | 1 | 3 | 4 |
| GTA IV | 5 | 5 | 5 | 2 | 4 |
| Half-Life 2 | 3 | 1 | 4 | 3 | 3 |
| Halo 2 | 3 | 2 | 2 | 4 | 3 |
| Left 4 Dead 2 | 4 | 3 | 3 | 2 | 6 |
| NFS ProStreet | 5 | 4 | 4 | 3 | 4 |
| PoP: The Sands of Time | 3 | 0 | 2 | 2 | 1 |
| Runaway: A Road Adventure | 0 | 1 | 1 | 1 | 2 |
| Sid Meier's Civilization IV | 6 | 3 | 4 | 5 | 4 |
| Syberia II | 5 | 4 | 3 | 3 | 3 |
| The Sims 2: Apartment Life | 5 | 5 | 8 | 0 | 3 |
| Tom Clancy's Splinter Cell | 6 | 0 | 8 | 5 | 10 |
| WOW: Wrath of the Lich King | 8 | 6 | 3 | 6 | 3 |
| Average | 44 % | 28 % | 33 % | 38 % | 36 % |

## 9.6 Similarity Mashup Examples

### *9.6.1 dbrec*

dbrec (http://dbrec.net/) [34] is a music recommendation engine based on information found using DBpedia. dbrec offers recommendations for more than 39,000 bands and solo artists. Additionally to recommending similar artists, which users might also like, dbrec provides explanations based on Linked Data. Figure 9.2 shows the dbrec recommendation engine.

Similarity between artists and bands is established using Linked Data Semantic Distance (LDSD), which is calculated on direct and indirect (including some third resource) paths through linked URIs.

To calculate LDSD for every artist, all instances of dbpedia:MusicalArtists and dbpedia:Bands were collected, resulting in more than three million triplets, which were then reduced to about 1,600,000 triplets by removing data type properties, redundancy, inconsistencies, and duplicates.

After analyzing their collected data, the authors of dbrec found that there are over 20,000 artists without direct links to other artists, and about 10,000 artists who had


Fig. 9.2 Screenshot of dbrec.net website, March 2012

direct links to three other artists. Thus indirect link computation was involved, to calculate LDSD for unlinked artists.

For evaluation, dbrec recommendations were compared to Last.fm recommendations. Results were rated from 1 to 5 from both systems individually, with later comparison of their overall ratings scores. Results showed that dbrec is a little behind Last.fm recommendations, however, the difference was quite small [34].

### *9.6.2 BlogAlike*

BlogAlike (http://blogalike.com/)—see Fig. 9.3—provides a search service for the most relevant articles of the same blog. BlogAlike users are required to put small JavaScript code on the pages of articles that will then be indexed by the BlogAlike service.

Each visited link which contains BlogAlike JavaScript code will be passed to a recommender service. If the URL is already indexed by the service, the JavaScript code will generate a list of recommended articles from the same domain. If the URL is new, it will be added to BlogAlike crawler's queue.

When indexing web pages, BlogAlike will generate a wrapper for your website to parse the contents, based on webpage parts that are different for the same

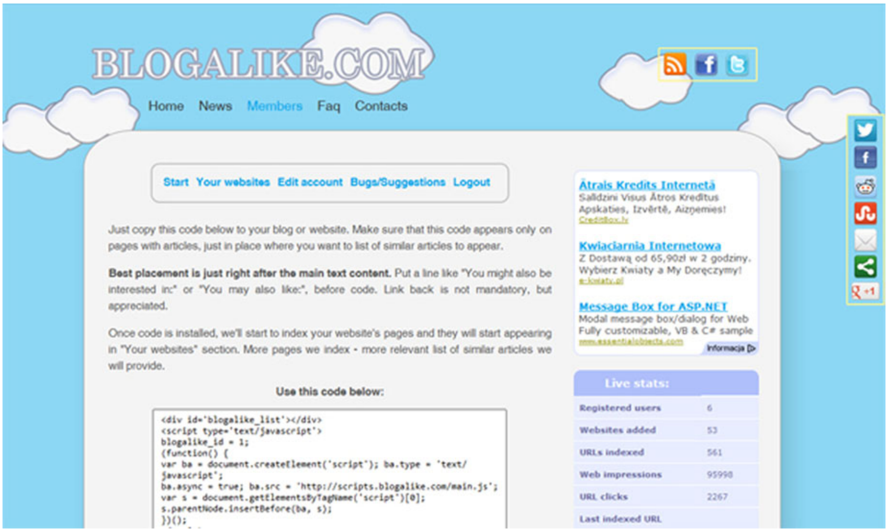



**Fig. 9.3** Screenshot of BlogAlike.com website, March 2012

domain. Then keywords are extracted from parsed texts, similarly to how it is done in GameDipper's case. Using these keywords recommendations are generated at runtime when loading scripts, to ensure that even recently indexed pages may be included if they are relevant.

BlogAlike provides its services in beta mode for more than 6 months, resulting in 160,000 additional page views provided. They are tracked by clicks on lists of similar articles.

### 9.6.3 Zemanta

Zemanta (http://www.zemanta.com/) is a content suggestion engine for bloggers and other content creators. It analyzes user-generated content using natural language processing and semantic technologies.

Zemanta offers related media as images, lists of related articles, in-text links and tags for any textual content. A Zemanta screenshot is shown in Fig. 9.4.

At the core of the Zemanta system there is a collection of semantic entities that represent concepts. Each entity is associated with some explanatory text, which will be used to establish similarities between concepts, and lists of possible aliases, which can be used as anchors for links and possible URLs with concept explanations as link destinations [42].

Current sources for Zemanta's recommendations are: Wikipedia, YouTube, IMDB, Amazon.com, Crunchbase, Flickr, ITIS, Musicbrainz, Mybloglog, Myspace, NCBI, Rottentomatoes, Twitter, Facebook, Snooth, Wikinvest and the blogs of other Zemanta users.

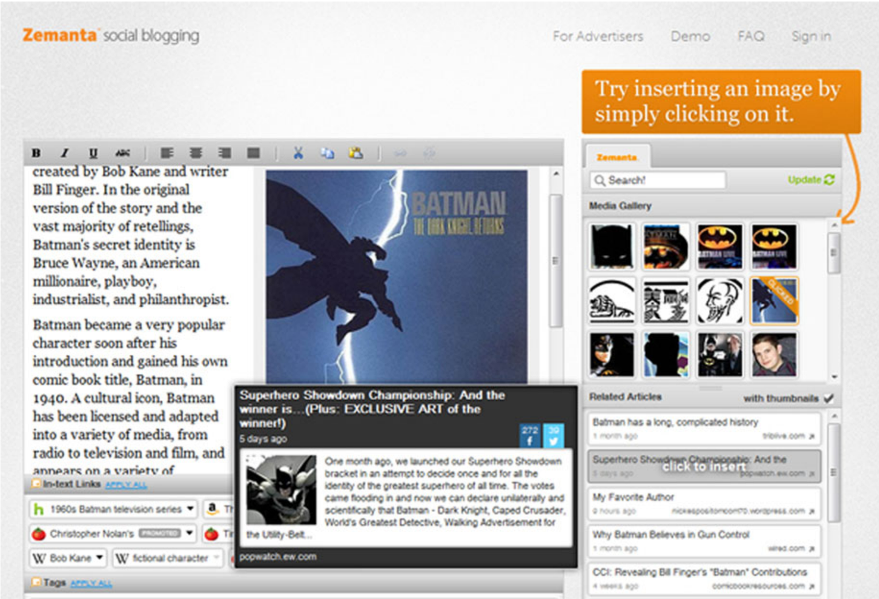



**Fig. 9.4** Screenshot of Zemanta tool example interface, March 2012

## 9.7 Conclusions

With the available approaches for determining content-based similarities between items it becomes easier to implement sophisticated recommendations systems. However, the collection of information on which to base the similarity of items remains a bottleneck.

To achieve better results, the quantity of available information may also matter, so that developers are stimulated to explore ways of automatically collecting available information on the web. There are a lot of web services available solely for this purpose, but for specific topics they may not be enough. Hence new approaches are needed, for example on web content scraping. These may pose their own problems, not only IT related, but also legal ones.

There are existing proofs of concept for new services. They can be developed not only for one limited topic, but also as much more general-purpose recommendations. Examples are Zemanta and BlogAlike services. They help to enrich existing content and provide related content using multiple information sources available on the Internet.

Mashups with the purpose of providing recommendation services by aggregating content from multiple sources face the deficit of available quality sources and services to do that. And although technologies like the semantic web, microformats and some aspects of the HTML5 standard ease this problem, it will still take time until these technologies are adapted by a majority of developers, so that they are generally available and not restricted to specific use cases and systems.

**Acknowledgements** The authors want to thank Raivis Gelsbergs, Aldis Salzemnieks, Edgars Vizulis, Aigars Stepanovs and Karlis Jerins for their support in creating the GameDipper recommendation engine. This work has been partly supported by the European Social Fund within the project Support for Doctoral Studies at the University of Latvia and the project Support for Master Studies at the University of Latvia.

## References

1. Burke R (2007) Hybrid web recommender systems. In: The adaptive web, pp 377–408
2. McFee B, Barrington L, Lanckriet G (2010) Learning similarity from collaborative filters. In: Proceedings of the International Society for Music Information Retrieval, Utrecht, pp 345–350
3. Lops P, de Gemmis M, Semeraro G (2011) Content-based recommender systems: state of the art and trends. In: Recommender systems handbook. Springer, Berlin, pp 73–105
4. Sosins A, Zviedris M, Gelsbergs R, Salzemnieks A, Vizulis E, Stepanovs A, Jerins K (2011) GameDipper: a search engine for similar video games. In: Extended semantic web conference (ESWC), AI mashup challenge
5. Mendes E, Mosley N (2006) Web engineering. Springer, Berlin
6. Fielding R (2000) Architectural styles and the design of network-based software architectures. PhD thesis, University of California, Irvine
7. Wilde E, Pautasso C (2011) REST: from research to practice. Springer, Heidelberg
8. Well T (2005) Internet law—trespass to chattels. http://www.tomwbell.com/NetLaw/Ch06.html. Accessed 18 Mar 2012
9. Gupta S, Kaiser G, Neistadt D, Grimm P (2003) DOMbased content extraction of HTML documents. In: Proceedings of the 12th world wide web conference, Budapest, pp 207–214
10. Chen L (2012) The hyper-text query language (HTQL). http://htql.net/htql-manual.pdf. Accessed 18 Mar 2012
11. Hickson I (2012) A vocabulary and associated APIs for HTML and XHTML. http://www.w3.org/TR/html5/. Accessed 18 Mar 2012
12. Zheng S, Song R, Wen J, Wu D (2007) Joint optimization of wrapper generation and template detection. In: Proceedings, Knowledge discovery and data mining, San Jose, pp 894–902
13. Crescenzi V, Mecca G, Merialdo P (2001) Roadrunner: towards automatic data extraction from large websites. In: Proceedings of the 27th international conference on very large data bases, pp 109–118
14. Jones K (1973) Index term weighting. Inf Storage Retr 9(11):619–633. doi:10.1016/0020-0271(73)90043-0
15. Matsuo Y, Ishizuka M (2003) Keyword extraction from a single document using word co-occurrence statistical information. Int J Artif Intell Tools 13(01):157. doi:10.1142/S0218213004001466
16. Krulwich B, Burkey C (1996) Learning user information interests through the extraction of semantically significant phrases. In: Proceedings of the AAAI spring symposium on machine learning in information access, Stanford
17. Turney PD (2000) Learning algorithms for keyphrase extraction. Inf Retr 2(4):303–336
18. Frank EF, Paynter GW, Witten IH (1999) Domain-specific keyword extraction. Morgan Kaufmann, San Francisco
19. Fabricius-Hansen C, Gallmann P, Eisenberg P, Fiehler R, Peters J (2005) Duden. Die Grammatik, Bd 4, 7. Aufl. Dudenverlag, Mannheim
20. Lovins JB (1968) Development of a stemming algorithm. In: Mechanical translation and computational linguistics, pp 22–31
21. Voorhees EM (1993) Using WordNet to disambiguate word senses for text retrieval. In: Proceedings of the sixteenth annual international ACM SIGIR conference on research and development in information retrieval, pp 171–180

22. Tversky A (1977) Features of similarity. Psychol Rev 84:327–352
23. Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. In: The fourteenth international conference on machine learning (ICML 97). Morgan Kaufmann, San Mateo, pp 412–420
24. Quinlan JR (1986) Induction of decision trees. Mach Learn 1:81–106
25. Lewis DD, Schapire RE, Callan JP, Papka R (1996) Training algorithms for linear text classifiers. In: Proceedings of SIGIR-96, 19th ACM international conference on research and development in information retrieval, Zurich, pp 298–306
26. Tunkelang D (2002) Making the nearest neighbor meaningful. In: SIAM workshop on clustering high dimensional data and its applications
27. Payne T, Edwards P (1998) Implicit feature selection with the value difference metric. In: Proceedings of the European conference on artificial intelligence, ECAI-98. Wiley, Brighton
28. Qian G, Sural S, Gu Y, Pramanik S (2004) Similarity between euclidean and cosine angle distance for nearest neighbor queries. In: Proceedings of the 2004 ACM symposium on applied computing. ACM, New York, pp 1232–1237
29. McCallum A, Nigam K (1998) A comparison of event models for naive Bayes text classification. In: AAAI-98 workshop on learning for text categorization
30. Zhang H, Li D (2007) Naive Bayes text classifier. In: Proceedings of the IEEE international conference on granular computing, p 708
31. Androutsopoulos I, Koutsias J, Chandrinos KV, Paliouras G, Spyropoulos CD (2000) An evaluation of naive Bayesian anti-spam filtering. In: Proceedings of the workshop on machine learning in the new information age, 11th European conference on machine learning (ECML 2000), Barcelona, pp 9–17
32. McCallum AK, Rosenfeld R, Mitchell TM, Ng AY (1998) Improving text classification by shrinkage in a hierarchy of classes. In: Proceedings of ICML-98, 15th international conference on machine learning, Madison, pp 359–367
33. Joachims T (1997) A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In: Proceedings of ICML-97, 14th international conference on machine learning, Nashville, pp 143–151
34. Passant A (2010) dbrec—music recommendations using DBpedia. In: Patel-Schneider PF, Pan Y, Hitzler P, Mika P, Zhang L, Pan JZ et al (eds) Proceedings of the 9th international semantic web conference, ISWC 2010, vol 1380. Springer, Shanghai, pp 1–16
35. Wang Y, Stash N, Aroyo L, Hollink L, Schreiber G (2009) Semantic relations for content-based recommendations. In: Proceedings of the 5th international conference on knowledge capture, K-CAP'09, New York, pp 209–210
36. How does Calais work? http://www.opencalais.com/about. Accessed 18 Mar 2012
37. Liu L, Pu C, Tang W (2000) WebCQ: detecting and delivering information changes on the web. In: Proceedings of the international conference on information and knowledge management, McLean
38. Chien LF (1997) PAT-tree-based adaptive keyphrase extraction for intelligent Chinese information retrieval. In: Proceedings of the 20th annual international ACM/SIGIR conference on research and development in information retrieval (SIGIR'97), Philadelphia
39. Wu D, Xia X (1995) Large-scale automatic extraction of an English–Chinese translation lexicon. In: Machine translation, vol 9. Kluwer Academic, Dordrecht, pp 285–313
40. Fung P (1995) A pattern matching method for finding noun and proper noun translations from noisy parallel corpora. In: Proceedings of ACL 1995, pp 236–243
41. About GameDipper. http://gamedipper.com/about.php. Accessed 18 Mar 2012
42. Solc T (2008) Automatic generation of in-text hyperlinks in web publishing. Tech rep, Zemanta Ltd

# Chapter 10
# Urban Mashups

**Daniele Dell'Aglio, Irene Celino, and Emanuele Della Valle**

**Abstract** Cities are alive: they rise, grow, evolve like living beings. The state of a city changes continuously, influenced by a lot of factors, both human (people moving in the city or extending it) and natural ones (rain or climate changes). Cities are potentially huge sources of data of any kind and for the last years a lot of effort has been put in order to create and extract those sources. This scenario offers a lot of opportunities for mashup developers: by combining and processing the huge amount of data (both public and private) is possible to create new services for urban stakeholders—citizens, tourists, etc. In this chapter, we illustrate the challenges in developing mashups for the urban environments: starting out from the specificity of cities and the availability of urban data and services, we describe a number of scenarios for urban mashups, we present our experience in realizing demonstrators of urban mashups and we discuss the lesson learned and the implications for citizens, tourists and municipalities.

## 10.1 Introduction

Cities are complex environments: they are populated by a number of different actors—citizens, commuters, tourists—and various stakeholders are interested in cities' management—public authorities like municipalities, businesses, transporta-

D. Dell'Aglio (✉) · I. Celino
CEFRIEL, Via Fucini 2, Milan, Italy
e-mail: daniele.dellaglio@cefriel.it

I. Celino
e-mail: irene.celino@cefriel.it

D. Dell'Aglio · E. Della Valle
Dipartimento di Elettronica, Informatica e Bioingegneria—DEIB, Politecnico di Milano,
Via Ponzio 34/5, Milan, Italy

D. Dell'Aglio
e-mail: daniele.dellaglio@mail.polimi.it

E. Della Valle
e-mail: emanuele.dellavalle@polimi.it

B. Endres-Niggemeyer (ed.), *Semantic Mashups*,
DOI 10.1007/978-3-642-36403-7_10, 

tion companies, and so on. Everyday all those parties interact, generating different challenges for the governance of urban environments. During the last few years, novel technological and business trends have led to the publication of huge amounts of location-specific data on the Web: people have supplied the so-called user-generated contents through social networks and blogs; public authorities have released data of public interest according to the Open Data philosophy; promoters of cultural events like concerts and museum exhibitions have produced advertising data. The result is a heterogeneous stock of independent data sources about cities.

Thus the urban scenario offers several opportunities to design and develop location-aware services that "mashup" data from those multiple sources to provide added-value applications for urban actors, such as traffic predictors, touristic route planners, interactive restaurant guides, etc.

Nonetheless, realizing those applications can be troublesome: often huge amounts of data should be effectively processed; data sources can be unreliable or conflicting; information can be incomplete, noisy, outdated or simply incorrect. A solution consists of combining several disciplines like Semantic Web, Machine Learning, Operational Research.

In this chapter we present the urban mashups topics: in Sect. 10.2 we discuss the domain, the challenges and the opportunities that cities and the urban settings offer. In Sect. 10.3 we present a set of axes useful to describe urban mashups; those axes will be then used in Sect. 10.4 to describe some representative demo mashups related to this context. Finally in Sect. 10.5 we conclude with some remarks about the most common open issues that developers can find while designing an urban mashup application.

## 10.2 Background

"Smart city" is the term that defines a vision of future cities in which innovation plays a central role to improve citizens' quality of life. Economical sustainability, low impact on the environment, avoidance of traffic congestions, intelligent transportation systems are only a few examples of the goals that this vision aims to achieve [24].

The smart city vision introduces a set of technological and social challenges, requiring the interplay of several disciplines to improve the cities performance. For those reasons, smart cities have become an exciting topic for researchers: urban open issues represent an ideal environment to experiment research solutions.

From an ICT (Information and Communication Technologies) point of view, smart cities are the research object of Urban computing [30]: the goal of this discipline is to change the way in which people feel the city, designing and developing new applications and services in the urban context. Urban computing is a multi-disciplinary field: its basis and principles can be found in Ubiquitous Computing and Geoinformatics.

Ubiquitous Computing [26] (also known as Pervasive Computing) refers to a new vision of machines: no longer a "traditional" computer with keyboards (and mouses) to give inputs and a monitor to watch the outputs, but a new kind of sensors and actuators integrated in several devices [40].

Geoinformatics [15], as the name suggests, researches on application of computer science in the geographical domain: two examples that are relevant to the Urban computing context are geocoding and geolocation. Geocoding [32] is the process to join generic data with information about location (i.e. coordinates); geolocation is a set of methodologies and instruments to discover the geographic location of different devices (often mobile devices like PDAs, notebooks and smartphones).

A fundamental requirement of Urban Computing is that cities should have input/output devices—sensors and actuators—to interact with environment and citizens [22, 33]. The basis to achieve this requirement can be found in the spread of smartphones and tablets during the last few years, enabling the citizen-as-a-sensor paradigm [25]. Those devices offer a set of sensors like GPS, camera and UMTS connections that can be used to supply data (acting like sensors) or to access services (acting like actuators). Example of Urban computing applications exploiting the mobile devices are the location-based services, such as Google Maps,[1] Foursquare[2] and Waze.[3]

The urban domain offers opportunities for mashup development: there are a lot of data sources that can be combined and processed in order to create new services for urban stakeholders (citizens, tourists, public administrators, etc.). Additionally, the urban scenario proposes several challenges—integration of heterogeneous data, (near-)real time processing, user-centered customizations—that can be solved applying techniques from different research topics such as Semantic Web, Artificial Intelligence and Data Mining. In the following of this chapter we will discuss the problem of designing mashups in this context, supplying examples of problems and scenarios.

## 10.3 Introducing Urban Mashups

Urban mashups are applications that put together different data and technologies to address the smart cities challenges outlined in the previous section. When developing this kind of mashups, it is important to take a holistic view by considering conceptual, technological and social aspects and their interplay. Hereafter, we offer a possible key to describe and analyze urban mashups, taking into consideration both the specificity of city environments and the technological and development choices in realizing those applications.

Table 10.1 introduces the four analysis dimensions that are illustrated in the following and that will be used in the rest of the chapter to describe urban mashup case studies. To ease the explanation, we use as running example the Urban LarKC [18],

[1] Cf. http://maps.google.com.

[2] Cf. http://foursquare.com.

[3] Cf. http://www.waze.com.

**Table 10.1** Urban mashup analysis method

| Stakeholders | Mashup Data | Mashup Processing | Mashup Delivery |
|---|---|---|---|
| Those who benefit from the mashups, either offering or using them | Datasets about the urban environment used or generated via the mashup | Technologies and services used to process urban data in the mashup, with specific reference to AI | Tool, interface or service to offer mashup functionalities to users |

a simple yet archetypal application that puts together topographical, tourism, event data, and services about Milan. Through its Web interface, the Urban LarKC allows users to find interesting tourist places and relevant events happening on a specific date, and to compute the shortest path to reach those destinations, in order to plan their visit in the city of Milan.

## *10.3.1 Stakeholders*

In urban contexts, several stakeholders can be interested in the realization of urban mashups. Stakeholders are governments and public authorities: they often own data and services about cities and struggle to find the most suitable way to fulfill their citizens' requirements and requests. Smart cities in this respect means smart government: better understanding and monitoring the cities to improve quality of life of their citizens. In our Urban LarKC application, we got Milan topological data from the Municipality agency that orchestrates and controls mobility and environment.[4] This agency could benefit from urban mashups based on their data and services to better monitor the city life and, as a consequence, improve their action (e.g. by fine-tuning the traffic-lights timing to reduce traffic).

Beside institutional actors, local businesses represent another important stakeholder. In their case, the aim of developing urban mashups could be oriented to improve business visibility to potential customers. In the Urban LarKC mashup, local events organized by both public and private actors are retrieved and suggested to potential visitors: this kind of mashup could then constitute an additional marketing or advertising channel.

Last but not least, urban mashup stakeholders are all the "final users" of those applications, citizens commuters and tourists. Since urban computing aims to change the way people feel the city, mashups in this context constitute a new class of applications that could be provided to people living or moving in the cities, for example enabling new communication ways [6] or search features by "querying" the cities like databases. In the Urban LarKC mashup, the target user is a tourist interested in exploring the urban environment: the popularity of location-based services for mobile devices proves the growing interest and potential exploitation of this family of mashups for all the stakeholders described above.

[4]Cf. http://amat-mi.it.

### 10.3.2 Mashup Data

Datasets about cities are more and more available, also thanks to initiatives like the social and political Open Data movement [34]. Data come from both public and private sources and range across a large number of topics (maps, points of interest, sensors data, people activity, user-generated contents, etc.) and formats (structured data and unstructured content; relational databases, XML or ESRI shapefiles [21]; etc.).

This fact represents an important and necessary step towards the realization of smart cities; still the availability of data is not sufficient per se to develop an urban mashup.

One issue in this respect comes at conceptual level in relation to data modeling. Maps are the common abstraction used to model city data: streets, squares, rivers, rails, etc., their names and their location. Such models can vary in granularity of information (are one-way streets, traffic lights, traffic islands, etc. explicitly modeled?) and can be linked to other related datasets that are often produced and updated in an independent way (e.g. traffic, local businesses, bus stops, weather, pollution). Of course, no perfect model of a city exists and data are usually described by models that are "good enough" for a given purpose; this implies that mashup designers should deal with this issue. In the Urban LarKC mashup, we "linked" monuments and event locations in Milan to the closest node in the city topology: this light-way integration was made possible by the simple use of geographic coordinates (latitude and longitude) that were present in all the mashed-up datasets.

Another issue deals with the scale of available data. A common requirement in urban mashups is to compute large amounts of data (including real-time generated data) with low response times. This, on the one hand, calls for technological solutions able to process those "big data" and, on the other hand, requires an intelligent approach to divide data in smaller and more manageable chunks. In the Urban LarKC application, we adopted different strategies [18] to select, extract and process smaller portions of the street topology to compute paths.

With regards to the heterogeneity of data formats, a solution lies in the use of light-weight data integration means, like those offered by Semantic Web technologies. For example, in the Urban LarKC application we used RDF as interchange data representation format, thus easing the "mashup" of urban data.

### 10.3.3 Mashup Processing

The core characteristics to describe and analyze an urban mashup lie in the use of smart techniques to process and integrate urban data and services. The challenge here is to make different technologies interplay so that they satisfy one or more smart cities needs.

Most of the mashups we present below are designed as workflows: it means that the processing can be separated in several steps and each of them computes part of the system output. The LarKC platform [14] behave like this. It supports massive distributed reasoning and it aims to remove the scalability barriers of currently existing reasoning systems for the Semantic Web. LarKC offers a pluggable architecture that makes it possible to exploit techniques and heuristics from diverse areas such as databases, machine learning, cognitive science, Semantic Web, and others. Plug-ins can be combined in workflows, allowing to reuse components and to modularize applications.

Apart from common Web and Web Service technologies and traditional data management—as in both relational databases and spatial-specific solutions like Geographic Information Systems (GIS)—a number of different scientific and technical fields can play an important role in urban mashups. One of them is Artificial Intelligence with its multiplicity of topics and applications.

The urban mashups described in this chapter draw on several fields:

- on the Semantic Web, Logics and Linked Data for knowledge representation and reasoning;
- on Natural Language Processing, data mining and machine learning to process unstructured and structured data and derive additional knowledge;
- on Operational Research to address the path-finding specific needs of urban mashups;
- on Human Computation to involve citizens and tourists in the data provision or elaboration.

For example, the Urban LarKC mashup exploits Semantic Web technologies to manage monument data and to "glue" together the different components. Additionally the mashup uses traditional RESTful Web services for event information and Operational Research for path-finding.

### *10.3.4 Mashup Delivery*

The urban mashups can offer their functionalities to their intended audience through a multiplicity of possible channels.

Web-based mashups are usually delivered on the Web, either via Web sites or via REST, Web APIs or SOAP services; the availability and large popularity of map APIs eases and fosters a geographic-based visualization of data, thus paving the way to location-based and location-aware services. The Urban LarKC mashup is delivered to the tourist via a Web interface with a simple and intuitive interaction design.

Furthermore, for the last few years, we have witnessed the growing spread of smart phones with Internet connection and on-board sensors. This flourishing market led to the raise of mobile app stores (like Apple iTunes or Google Play), in which a large supply of mobile-specific apps encapsulate urban mashups functionalities.

A final consideration about the importance of building user-centered applications. Designing applications that consider user preferences is important. Customized and personalized services can represent an added value that influences the application adoption. Realizing user-centered applications influences not only the delivery of the functionalities, but often the modeling itself of the city: different points of view can be adopted to describe the city, thus there is no single or correct way to identify the best one [20], because of cultural bias or pragmatic reasons.

## 10.4 Examples of Urban Mashups

In this section we describe four urban mashups that integrate different kinds of data, adopt a variety of technologies and fulfill different requirements of smart cities scenarios. We use the four dimensions introduced and explained in Sect. 10.3 to illustrate and analyze those exemplary applications.

First we will describe the Traffic LarKC, an application that integrates traffic-related data to forecast the street conditions in the near future and find the most desirable routes between two points of the city. To this end, the application combines techniques from Machine Learning, Operational Research and Semantic Web.

Then we present BOTTARI, a location-aware application that offers to its users descriptions about Seoul restaurants and personalized recommendations about them. The system processes data streams from Twitter extracting opinions about restaurants through a Sentiment Analysis engine and then elaborates recommendations with a combination of inductive and deductive reasoning.

The third presented mashup is UrbanMatch Milano, a Game With a Purpose (GWAP) that aims to annotate photos with the monuments they represent. The application combines data from Open Street Map, Flickr and Wikimedia Commons, and processes them through a Human Computation approach.

Finally, we will introduce Korean Road Sign Management, an application to help the Korean road traffic authority in checking the validity of the road signs' contents placed in Seoul. This mashup exploits Semantic Web techniques to identify the inconsistencies in the road sign representations.

### *10.4.1 Traffic LarKC*

A common problem of people living in cities is route planning, the task of finding paths connecting two or more points, given a set of constraints. For example, people could be interested in planning a route to visit some shops during their opening time, or they have to find a way to move from home to school/workplace in time and taking into account the traffic conditions. The application we will discuss in this section, named Traffic LarKC [19], is a mashup to compute a set of paths between two points taking into consideration several factors, like route length and traffic estimation. The Traffic LarKC won the AI Mashup Challenge 2011.[5]

[5]Cf. http://www.eswc2011.org/.

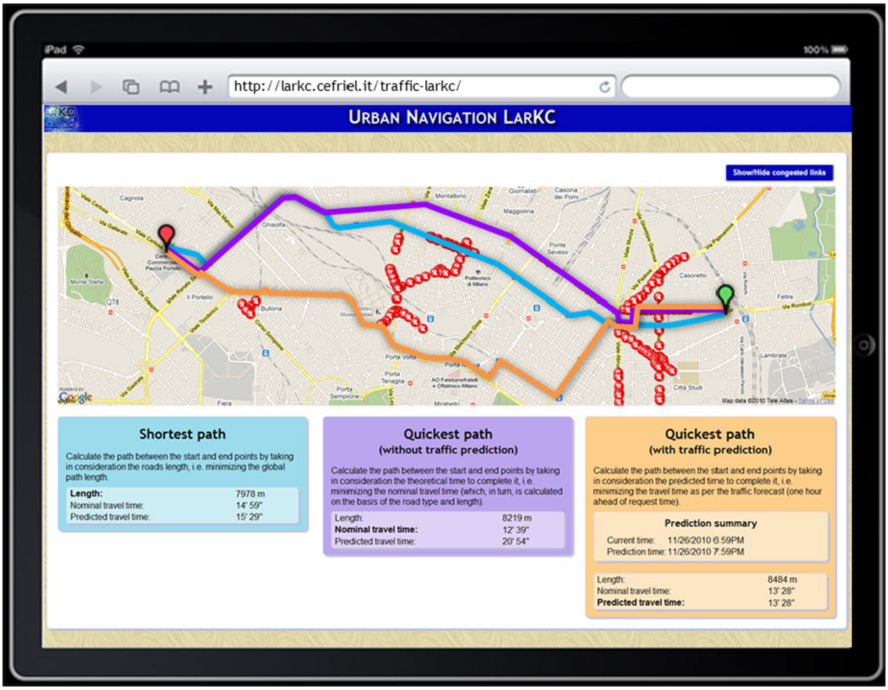



**Fig. 10.1** Screenshot of the Traffic LarKC

#### 10.4.1.1 Mashup Overview

People in (unfamiliar) urban environments ask questions like "which is the quickest way to a modern art exhibition?" or "which are the modern art exhibitions that I can reach in less than 25 minutes if I can get into my car this afternoon at 4 pm?". To find an answer to those questions a combination of pure semantic information retrieval (the available modern art exhibitions), geo-spatial processing (the path to the desired destination) and machine learning (statistical traffic forecasting) is required. Moreover, we must face a number of challenges, ranging from the data size to time-dependency, from the heterogeneity to the quality of data sources, from the different semantic layers of information to the unknown knowledge of unobserved events, etc.

The Traffic LarKC is a system that aims to overcome the existing problems and finds useful answers to users' requests. While using RDF as interchange format to support the integration with semantic processing and formal reasoning, the Traffic LarKC combines state-of-the-art statistical learning, probabilistic reasoning and operational research. The resulting application is able to route users from their current position to a target point of interest within the city of Milan, taking into account the future traffic conditions at the projected time of travel (Fig. 10.1).

The challenges faced by the Traffic LarKC are as follows [9]. The historic traffic database contains more than 1 billion triples and predictions amount to 9 million

**Table 10.2** Traffic LarKC according to the defined dimensions

| Stakeholders | Mashup Data | Mashup Processing | Mashup Delivery |
|---|---|---|---|
| Municipalities, citizens | Maps, traffic data, weather data | Machine Learning, Operational Research and Semantic Web | Web application and SPARQL end-point |

new ones each day. Thus, data size is an issue when real-time predictions are required. The data are also very noisy, e.g. due to broken sensors, and do not obey a closed world assumption due to many unobserved effects, e.g. parking cars or small accidents. Moreover, traffic data are time-dependent and a prediction framework requires heterogeneous data sources, such as a street graph, historic time-series of speed and flow at traffic sensors, weather data, or different calendar events like special holidays. On the query side, the routing should take into account the desired path (the shortest path vs. the fastest one, the best path on an average day vs. the best one at a specified time–date) and it should be coupled with "semantic" layers of the city, such as its points of interest.

Those challenges can be mastered through the LarKC semantic computing platform [14]: it exploits the flexibility of its RDF data representation with a routing-oriented ontology and provides a query interface compliant to the SPARQL standard. The algorithmic methods, however, are not just restricted to formal reasoning, but different pluggable parts employ state-of-the-art statistical learning and efficient approaches from operations research. In the following we will explain how those techniques from different fields can easily and efficiently be integrated with the help of the LarKC platform to obtain a traffic-aware routing system.

The summary of the analysis on the Traffic LarKC analysis is reported in Table 10.2.

#### 10.4.1.2 Mashup Analysis

The stakeholders of the Traffic LarKC are citizens: they can use the application to find the best route to move from a point of the city to another one. The provider of the application could be the public administration: on the one hand they own most of the data required as input (traffic sensors streams, city maps, etc.), on the other hand they can also be final users of the application: the results of the computation of the Traffic LarKC are a potential useful dataset to build traffic analysis tools.

The Traffic LarKC combines information from several sources—maps, traffic sensors recordings, calendar and weather data. The data about street topology and traffic sensors were obtained from the Municipality of Milan, Agenzia Mobilità Ambiente e Territorio (AMAT). They consist of a very detailed topology map with more than 30,000 streets (i.e. portions of roads with a specific flow direction) with 15,000 nodes (i.e. road junctions); each street portion is described with a set of both geometrical attributes (e.g. coordinates, length, number of ways, etc.) and flow-related characteristics (e.g. indicators of flow and congestion, turning prohibitions, etc.). The traffic sensors data give information about 300 sensors with their positioning

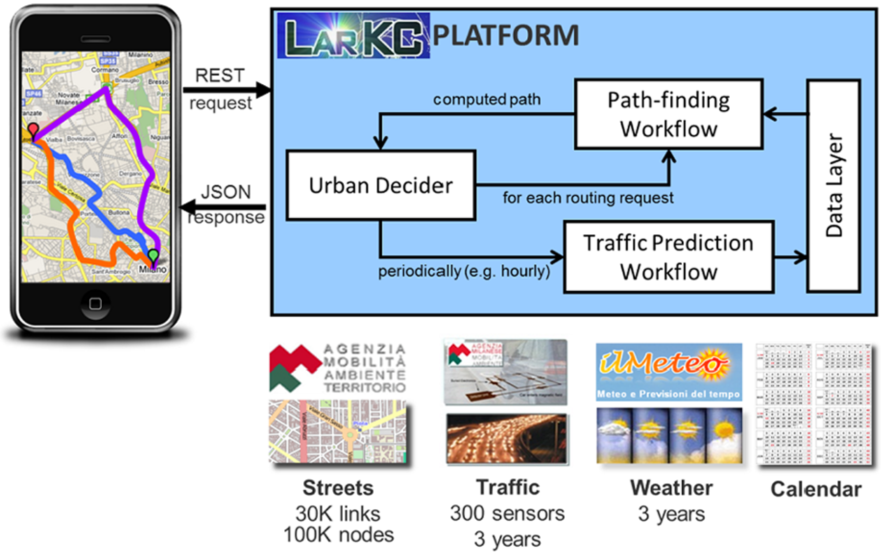



**Fig. 10.2** Traffic LarKC workflows

and sensing capabilities; the 3 years time-series of those sensor data records the traffic as sensed every 5 minutes intervals. As such, the sensors records sum up to more than $10^9$ records in a 250 GB database. Additionally, for the same time-span, that information is complemented with historical weather data from the Italian website ilMeteo.it[6] (CSV data with $10^8$ records) and with calendar information (week days and week-end days, holidays, etc.) from the Milan Municipality and from the Mozilla Calendar project.[7]

For the Traffic LarKC two workflows (depicted in Fig. 10.2) were designed. They run on top of the LarKC platform: the first "on-demand" runtime workflow calculates the most suitable path between the starting point and the destination in the user's request; the second "scheduled" batch-time workflow periodically recomputes the traffic predictions for the next two hours for all streets in Milan. The two workflows read their inputs and write their computation results via the LarKC Data Layer—an extension of the BigOWLIM triple store[8] used as shared storage area—within the platform. Finally a "Decider" plug-in orchestrates the behavior of the two workflows and manages the request/response interaction with the user interface.

The LarKC path-finding workflow encapsulates the Operational Research algorithm to compute the best path between two points on the map of Milan. In order to find the path, it is necessary to define what "best" does mean. In fact the policy

[6]Cf. http://www.ilmeteo.it (Italian).

[7]Cf. http://www.mozilla.org/projects/calendar/holidays.html.

[8]Cf. http://www.ontotext.com/owlim.

defines the specific dimension that should be minimized when computing the "shortest" path; in our scenario, this dimension can assume three values: the path length, the nominal travel time (traversal without traffic) or the estimated travel time (using traffic predictions). Following this modeling, the path computation is expressed in RDF—the considered interchange format—while keeping the actual processing inside a LarKC plug-in that encapsulates the Dijkstra algorithm (to compute the most desirable path).

The traffic prediction LarKC workflow combines different Machine Learning algorithms; the workflow consists of a pipeline with three steps. First, the application uses time-delay Recurrent Neural Networks (RNN) [41] in order to forecast traffic speed and flow at sensor locations for the next four hours in 5 min intervals. For this task the Traffic LarKC considers the sensors traffic observations from the last 24 hours that are available through the platform Data Layer. The second step is the categorization of the predictions into two robust traffic conditions: normal or congested. Last, the Traffic LarKC generalizes the traffic conditions from sensor locations to all streets of the road network and assign estimated travel times based on predicted traffic condition, road length and category. To solve this task it employs a Bayesian formulation of semi-supervised learning [13]. The results are then written back to the Data Layer for further query processing.

Traffic LarKC uses Semantic Web technologies for integration purposes. In order to do it the application exploits the LarKC platform. In fact LarKC is "semantic" in that it uses RDF as data format and light-weight data integration means, but it goes well beyond usual Semantic Web platforms in that it demonstrates its flexibility in encapsulating Neural Network systems for the traffic prediction and Operational Research routing algorithms for path finding.

There are two ways to deliver the result of the Traffic LarKC computation. The first one is a SPARQL end-point: LarKC and applications built on top are exposed to the Web through a SPARQL end-point. This means that it is possible to send LarKC path finding queries, specifying some parameters (the start and the goal nodes, the policy, etc.), receiving the computed path as response. The second way is a "user-friendly" interface: a Web application (shown in Fig. 10.1) that allows users to send their requests without requiring the technical knowledge to formulate the SPARQL queries.

#### 10.4.1.3 Additional Details and Evaluation

The quality of the RNN traffic forecasts was evaluated and the results are displayed in Fig. 10.3. On the left traffic flow time-series for some example sensors are shown. The past 24 hours of known measurements are used to predict the next four hours. A numerical evaluation against other standard regression techniques, namely a feed-forward neural network and linear regression, is presented on the right. The average relative error of the time-delay RNNs is significantly lower than for the competing methods, and also shows a much smaller variance.

An example of the network-wide generalization is shown in Fig. 10.4. Numerical validation is problematic here, as no in-between-the-sensors information was

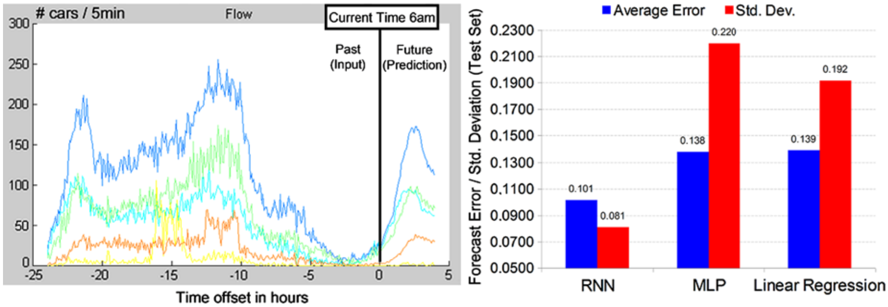



**Fig. 10.3** RNN traffic predictions: time-series with some example sensors (*left*), comparison of the time-delay RNNs vs. feed-forward neural networks and linear regression (*right*)

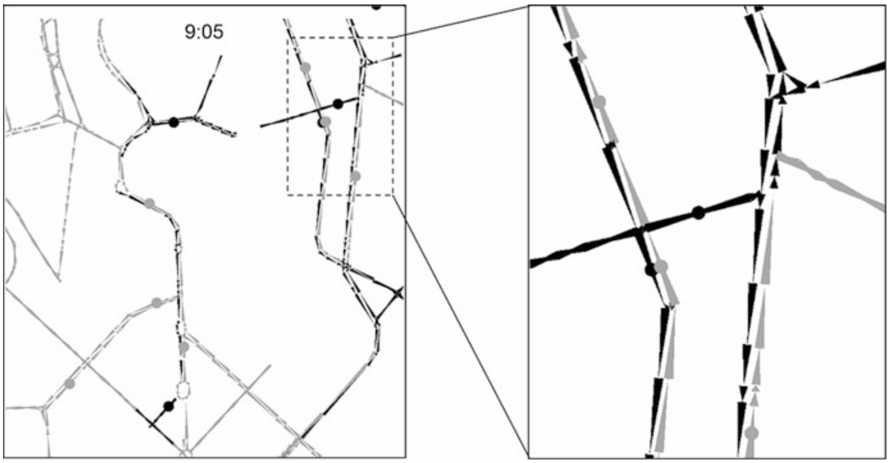



**Fig. 10.4** Results of generalizing traffic condition predictions from sensor locations (*dots*) to all links of the road network via Bayesian Semi-supervised Learning. *Light gray* means normal condition, *dark gray* congested

available. However, the results are qualitatively plausible. They show connected areas of congestion around sensor locations with traffic distortions. Different road directions—modeled with separate links—may show different traffic situations, as common in real situations.

### *10.4.2 BOTTARI*

BOTTARI is a location-aware social media analysis mashup that, starting from Twitter micro-posts, analyzes the "sentiment" of people regarding restaurants in a tourist area of Seoul in Korea and provides details and recommendations about


**Fig. 10.5** Screenshot of BOTTARI Android app with the four recommendation types

those restaurants to a user in mobility. BOTTARI won the first prize of the Semantic Web Challenge 2011.[9]

The peculiarities of this mashup rely on: the combined use of curated datasets (details about restaurants) and user-generated data (comments about restaurants), the exploitation of large-scale streaming data, the employment of an innovative mix of inductive and deductive techniques and the clear business potential of the resulting end-user application.

#### 10.4.2.1 Mashup Overview

BOTTARI [2] is an Android application (for smart phones and tablets) in augmented reality (AR) that directs the users' attention to points of interest in the neighborhood of the user's position, with particular reference to restaurants and dining places. However, BOTTARI does not simply show the available places; indeed it provides personalized recommendations based on the local context and the ratings of the POI derived by micro-posts analysis. It offers different types of recommendations based on the mashup of the different data and techniques.

BOTTARI provides to its users four different types of recommendations (cf. top-left of Fig. 10.5):

- *Interesting* recommendations suggest POIs indicated for foreign visitors in Korea; this feature calls for analysis and retrieval of POIs attributes;

[9]Cf. http://challenge.semanticweb.org/2011/.

- *Popular* recommendations suggest the POIs which show the highest level of reputation on social media; this feature calls for a complete analysis of the social sentiment about POIs;
- *Emerging* recommendations suggest the most popular POIs in a delimited period of time (e.g. last 6 months); this feature calls for the identification of "hypes" and new trends in the social sentiment;
- *For me* recommendations suggest POIs of interest for the current user; this feature calls for personalized recommendations.

Those four types of recommendations provided by BOTTARI require different levels of semantic technologies. BOTTARI is a mashup of those techniques as explained hereafter.

The *Interesting* kind of recommendations requires to suggest the user with a subset of the POIs that matches (1) the user current location and (2) the category of "attractions of interest for foreign visitors". To provide those recommendations, Semantic Information Retrieval techniques are employed. The SOR triple-store[10] containing the mashed-up data provides a geographic extension of SPARQL to query both the "semantic" description of POIs and their physical location.

The *Popular* type of recommendations requires an analysis of the social media. The tweets are processed by a sentiment analysis algorithm that detects if the message talks about a POI and, in case, if it expresses a positive or negative rating on the POI. The approach to compute the "sentiment" is twofold: on the one hand, a pure machine learning approach using SVMs (Supporting Vector Machines) with syllable kernel is used and, on the other hand, an NLP rule-based approach is employed to analyze the Twitter messages in terms of their structure and language.[11] Once the sentiment is elicited, this information is attached as metadata to the message description in the triple store. The popular recommendations are then generated by querying the knowledge base and suggesting the POIs with the highest number of positive ratings; the geographic features of SOR are also used to filter POIs and recommend only those around the user current location.

The opinion of users on POIs can change over time: the *Emerging* kind of recommendations suggests the users with POIs that are "on fashion" in the latest period of time. To this end, Stream Reasoning [17] was adopted to identify trends and changes in the sentiment about the POIs. Because of the sentiment analysis elaboration, the stream of messages annotated with the user sentiment is not in real time, but it is "re-streamed" from its storage. The queries enabled by the C-SPARQL Engine [3, 4] let find the emerging opinion of users about POIs: the engine counts the positive opinions about a POI per each day and their aggregation by week or by month.

Finally, POI recommendations can be personalized: the user can be suggested with POIs that could be interesting for her. To this end, inductive reasoning was

[10]Cf. http://semanticwiki-en.saltlux.com/index.php/SOR.

[11]Those rules are both manually coded and generated by machine learning algorithms with specific reference to the Korean language.

**Table 10.3** BOTTARI according to the defined dimensions

| Stakeholders | Mashup Data | Mashup Processing | Mashup Delivery |
|---|---|---|---|
| Citizens and Tourists, Local businesses | Curated dataset and Social Media content | NLP, Sentiment Analysis, Stream Reasoning, Collaborative Filtering and Data Mining | Mobile app, Web app, SPARQL end-point |

adopted on social media to compute BOTTARI's *For me* recommendations. The SUNS approach (Statistical Unit Node Set) described in [5, 37] was exploited. SUNS is a machine learning approach for exploiting the regularities in large datasets in relational and semantic domains. The approach can be used to detect interesting data patterns and predict unknown but potentially true statements. In BOTTARI SUNS estimates the probability that a user will like a POI, based on the sentiment the same user expressed about other POIs and the opinion that other users expressed about that POI. In this sense, BOTTARI provides a personalized collaborative filtering recommendation engine, to suggest users with the most interesting POIs with respect to their preferences.

#### 10.4.2.2 Mashup Analysis

A summary of the main characteristics of BOTTARI according to the dimensions described in Sect. 10.3 is offered in Table 10.3.

Regarding the *stakeholders*, the final users of BOTTARI are citizens or tourists moving in Insa-dong, equipped with a mobile device (an Android phone in this specific case) and looking for a restaurant. The assumption is that, on the one hand, people wish to have "on-site" recommendations about the environment they are moving in and, on the other hand, micro-blogging platforms like Twitter are a very good source for this location-based wisdom. Among the stakeholders, however, a relevant role is played by local businesses, which are eager to understand the changing opinion of users about their offered services. The "sentiments trends" displayed by BOTTARI (see Fig. 10.6) could give very valuable insights to local businesses about what is on-fashion and favorite by their customers.

The input *mashup data* come from both curated and user-generated Web sources. One main source of information is a curated dataset about the Insa-dong area, and collects information about some 100 POIs, each one described by a few dozen attributes (location, description, place category, price range, reviews, contacts, etc.); this dataset content is quite static and is used as "background" information about the POIs. These data are expressed in RDF, described with regards to an OWL ontology and sum up to more than 20 thousand triples. The other dataset is gathered from social media. The main source consists of tweets collected from Korean users (i.e., all tweets are written in Korean language) between February 2008 and November 2010; those short messages are acquired by means of the Twitter APIs, are further elaborated to identify the tweets talking about POIs in Insa-dong and processed

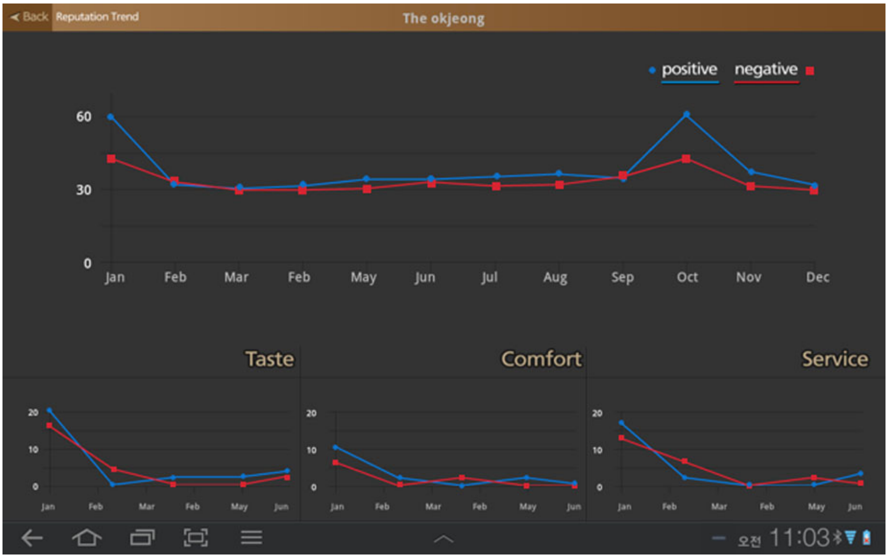


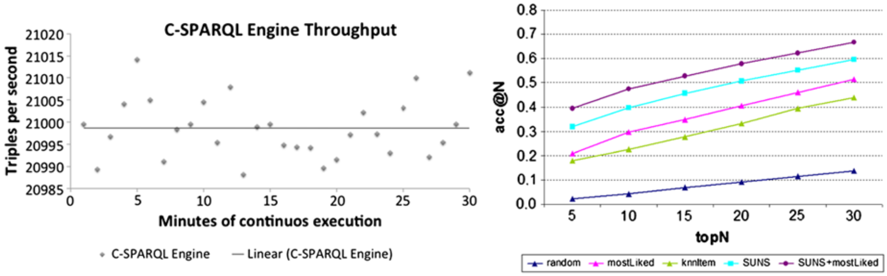

**Fig. 10.6** Sentiment trends over time about a restaurant recommended by BOTTARI

to assess the "sentiment" they express (positive judgment vs. negative rating). The results are expressed in RDF and described with regards to the OWL ontology illustrated in [10]; those triples—which account for almost 1 billion triples—are then stored in a SOR triple-store repository. The output dataset—the user sentiment about restaurants and its evolution over time—is very valuable for the local businesses as explained before.

The technologies involved in the *mashup processing* are various semantic technologies: NLP and sentiment analysis to process the micro-posts, Stream Reasoning (both the Semantic Web-based deducting reasoning and the Machine Learning-based inductive reasoning) to analyze sentiment trends. All those techniques were mashed-up together by using the semantic workflow-based LarKC platform [14] which allows for a seamless integration between different methods for large-scale data processing.

The BOTTARI *mashup* is delivered in different forms. The final users enjoy the mashup under the form of a mobile app for Android; the stakeholders interested in the various trend analysis are given a Web-based application[12] that visualizes BOTTARI "back-end features". Moreover, the output dataset can be queried via traditional Semantic Web technologies, through SPARQL queries to the BOTTARI end-point (which is able to process both SPARQL and C-SPARQL queries).

[12]Cf. http://larkc.cefriel.it/lbsma/bottari/.




(a) (b)

**Fig. 10.7** BOTTARI evaluation: (**a**) execution of a continuous query registered in the C-SPARQL Engine and (**b**) accuracy of recommendations

### 10.4.2.3 Additional Details and Evaluation Results

In the stream reasoning processing the C-SPARQL Engine [4] is one of the main components. In Fig. 10.7(a) we report the results of a long lasting execution of a simple continuous query registered in the C-SPARQL Engine: the query matches all triples in the stream and regenerates them out. In our experimental settings (an Intel Core 2 Duo T7500 at 2.2 GHz, 4 GB of RAM DDR2 at 667 MHz, Hard Disk at 5400 rpm), the C-SPARQL Engine throughput was between 20,987 and 21,015 triples/second; the average throughput was 20,999 triples/second. This result proves that we are able to process the amount of data currently produced in social media streams.

In Fig. 10.7(b), we show the accuracy of recommendations: we compare two baseline algorithms—random guess (random) and item-based k-nearest neighbor (knnItem)—with C-SPARQL's *emerging* recommendations (mostLiked) and SUNS' *for me* recommendations; here, we do not consider the location dimension (i.e., in this evaluation we recommend POIs regardless the current position of the user). As expected, the random is the worst; C-SPARQL is slightly better than the similarity-based method: this might indicate the "bandwagon effect" that exists in many social communities; SUNS significantly outperformed all other methods (with a number of the latent variables greater than 100). The best ranking was produced by the combination of both SUNS and C-SPARQL: these results confirm again the effectiveness of combined approaches of deductive and inductive reasoning [5].

The design of the BOTTARI service from the user point of view, as well as its prototypical implementation and its early evaluation, demonstrate that Semantic Web technologies can be successfully applied to concrete scenarios and can help in adding added-value functionalities. We foresee a potential market exploitation of this kind of location-based social media analysis applications and the Korean company Saltlux[13]—which was one of the initiators and contributors to the BOTTARI prototype—is going to continue the development of BOTTARI to a commercial product for Korean customers.

[13] Cf. http://www.saltlux.com.

### *10.4.3 UrbanMatch Milano*

UrbanMatch Milano is a mashup that integrates location-based data about POIs in a city from open data and linked data sources—like OpenStreetMap[14] and Linked-GeoData[15]—and user-generated contents about those POIs, in the form of photos. The peculiarity of this mashup, with respect to the other ones described in this chapter, consists of the fact that the data processing is not only based on computer-based algorithms (like automatic linking processes) but relies also on "human computers": the mashup users provide valuable contribution to the improvement of the quality of the input datasets.

#### 10.4.3.1 Mashup Overview

The UrbanMatch Milano game [11] is a mobile gaming application that joins data linkage and data quality/trustworthiness assessment in an urban environment. By putting together Linked Data [27] and Human Computation [39], UrbanMatch creates a new interaction paradigm to consume and produce location-specific linked data by involving and engaging the final user. This game can also be seen as an example of value proposition and business model of a new family of linked data applications based on gaming in Smart Cities.

UrbanMatch[16] aims at selecting the most representative photos related to the points of interest (POI) in an urban environment; more specifically, UrbanMatch is oriented to link the monuments and relevant places of the city of Milan with their respective photos as retrieved from social media Web sites and to "rank" those links, so to identify the most characteristic ones and to discard the others, thus improving the quality.

The UrbanMatch game is a photo coupling game. The game mechanics respects the best practice of casual games and Games with a Purpose [23]: it consists of a simple and intuitive interface that presents the player with eight photos of POIs in the vicinity of the player and asks for their coupling (cf. Fig. 10.8).

Through a Human Computation approach, UrbanMatch collects evidence of players decisions to correlate images. Then, the collected data are processed—similarly to what happens in other Games with a Purpose [38]—with majority voting and other statistically relevant algorithms [12]. The elaboration of those pieces of evidence leads to a ranking of the photos and thus makes it possible to select the most representative pictures of the urban POIs.

---

[14]Cf. http://www.openstreetmap.org.

[15]Cf. http://linkedgeodata.org/.

[16]Cf. http://bit.ly/urbanmatch.


**Fig. 10.8** Screenshots of UrbanMatch Milano

**Table 10.4** UrbanMatch Milano according to the defined dimensions

| Stakeholders | Mashup Data | Mashup Processing | Mashup Delivery |
|---|---|---|---|
| Citizens, Tourists, Tourism offices | User-Generated Content (photos and data about POIs) | Human Computation, Semantic Web | Mobile app, Linked Data |

#### 10.4.3.2 Mashup Analysis

A summary of the main characteristics of UrbanMatch Milano according to the dimensions described in Sect. 10.3 is offered in Table 10.4.

Regarding the *stakeholders*, the final users of the game are citizens or tourists moving in the urban environment of Milan equipped with a mobile device (an iPhone in this specific case). The assumption is that, on the one hand, people "on-site" can better distinguish the photos that actually depict the POIs surrounding them and, on the other hand, the gaming flavor of the application can engage people to contribute to the data processing. The recent popularity of location-based services

(LBS) is a sign that this approach can be successful: people are more and more used to "check-in" physical places with their mobile devices and to add small bits of information related to their activities and actions in the physical world.

The output dataset—a high-quality set of photos correctly linked to the respective POIs—can be interesting for a number of target users. Firstly, tourism offices and in general local businesses interested in tourism could benefit from the dataset, which represents a clean, curated and open-licensed set of multimedia files about tourist attractions; the content of this dataset saves the need for an image search on the Web. Additionally, the fact that the POI–photo "links" are released under the form of linked open data—reusing or linking to existing resources of the LOD cloud like LinkedGeoData—makes the UrbanMatch output dataset very interesting for a much larger audience interested in the reuse and (semantic) mashup of urban-related data.

The input *mashup data* come from available Web sources. Points of interest in Milan were collected and chosen among those available from OpenStreetMap, a collaborative project to create a free editable map of the world, whose approach to mapping was inspired by wiki-sites such as Wikipedia. An RDF description of those OpenStreetMap POIs is also available in LinkedGeoData [1], an effort to add a spatial dimension to the Semantic Web that uses the information collected by the OpenStreetMap project, interlinks those data with the LOD cloud and makes the result available as an RDF knowledge base according to the Linked Data principles.

A high number of photos of Milan POIs were collected from Wikimedia Commons[17]—the media collection of Wikipedia—and from Flickr,[18] probably the most popular social media sharing site dedicated to photos. The images were collected either by keyword/concept search (i.e., photos explicitly related to Milan POIs) or via location-based queries (e.g., search by geographical coordinates). Among the collected photos, we considered only those released with an open license, allowing for a free reuse of the image (like the CreativeCommons "Attribution" license).

The technologies involved in the *mashup processing* are Human Computation [39] (under the form of Games with a Purpose [38]) and Semantic Web ones. The former is aimed to involve the user in the loop and to mashup people capabilities with the computer-based data processing, while the latter is employed both to create the initial dataset—POI–photo links derived from the available open data and linked data sources which are then processed by the game players—and to publicly release the output dataset according to the Linked Data best practice [7].

The *mashup* is delivered in different forms. The game players enjoy the mashup under the form of a mobile app for iPhone; the stakeholders interested in the output dataset get the mashup results via Linked Data technologies, since the POI–photo links selected by the game players are re-published on the Web of Data and linked to the pre-existing sources, like LinkedGeoData. Currently, UrbanMatch delivers only the selected data under a simple RDF triple form: `<POI> foaf:depiction <photo>`. All player evidence and the confidence values attached to each POI–photo link might be published as linked data, too.

[17]Cf. http://commons.wikimedia.org/.

[18]Cf. http://www.flickr.com/.

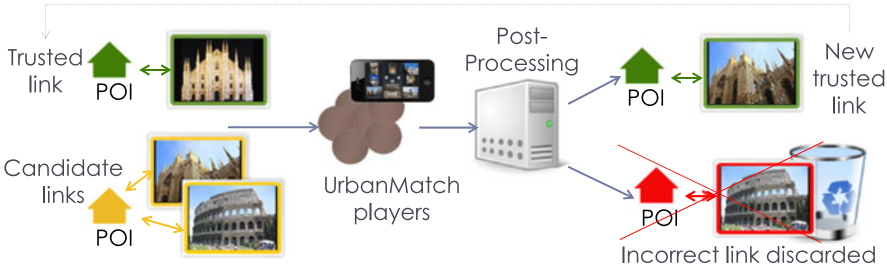



**Fig. 10.9** Computing UrbanMatch links (from [11])

### 10.4.3.3 Additional Details and Evaluation Results

UrbanMatch input data are of two different types: the POI descriptive data and a manually selected set of photos linked to those POI are a "trusted source" in that their value is ensured by design; this dataset is constituted by 196 POI–photo links (expressed in RDF as explained before) whose validity is certain. A second "uncertain source" is on the other hand formed by the automatically selected photos retrieved by the POI-based search on Wikimedia Commons and Flickr; the second dataset is constituted by more than 37,000 candidate links, which should be validated by the UrbanMatch Human Computation approach. Those candidate links are annotated with a *confidence value* that expresses the lack of certainty about their trustworthiness.

As visualized in Fig. 10.8, players are presented with eight photos and have to find matching couples, i.e. they have to mark the photos that depict the same POI. The user geographic position is taken from the mobile device sensors to propose photos about the POIs in the proximity of the player. The links between the POIs and the presented photos are not the same for all the presented eight photos: some links are *certain*, because they come from the trusted source, and some are *uncertain*, because they are taken from the set of candidate links. The game purpose is to modify the confidence value of the candidate links to assess if they are trustable or incorrect.

The mashup human computation is represented in Fig. 10.9, in which trusted POI–photo links are represented in green (the upper left link with Milan's Duomo picture) and the candidate links related to the same POI are marked in yellow (those on the left, with uncertain photos retrieved from Wikimedia Commons and Flickr as being related to Milan's Duomo).

If a player associates a trusted photo with an uncertain photo, the candidate link related to the latter is given a sign of "trust" and its confidence value is increased. Otherwise, if there is no evidence of the association between an uncertain photo and any other one, the lack of coupling actions is considered as a sign of "distrust" and decreases the confidence value of the candidate link. When, after a variable number of played games in which different players were given the same candidate link, its confidence value crosses some thresholds, the link leaves its uncertainty status and becomes either a trusted link (confidence value greater than an upper threshold, see top-right of Fig. 10.9) or an incorrect one (confidence value smaller than a lower threshold, see bottom-right of the figure, in which the photo evidently depicts Rome's Colosseum).

UrbanMatch evaluation is aimed to assess the game purpose, thus a number of metrics were identified to measure the mashup capability to improve the quality of the urban-related data involved in the game. The *completeness* metrics is defined as the capability of the game to assess all the input candidate links, deciding if they are either trustable or incorrect. The completeness is calculated by dividing the number of assessed links (i.e. the links that became either trusted or incorrect after the gameplay) by the total number of input uncertain links. The *accuracy* metrics is defined as the capability of the game to make correct assessments about the input links, minimizing the "false positive" outcomes (i.e., POI–photo links considered trustable but actually incorrect) and "false negative" outcomes (i.e., POI–photo links considered incorrect but actually trustable). To measure the game accuracy, the assessed links were manually checked to identify the false positive/negative items. The accuracy is then calculated by dividing the number of correct assessments (true positive and true negative items) by the total number of input uncertain links.

A preliminary evaluation gave the following results: evidence was collected from 54 unique players who played 781 game levels, in which they tested 2,006 uncertain links. Setting the thresholds on the confidence value to 70 % and 20 % for the upper and lower limits, respectively, the game assessed the correctness/incorrectness of 1,284 uncertain links, getting to an improvement of the global completeness from 1.54 % to 4.98 % with a final accuracy of 99.4 % (four false positive and eight false negative links).

## *10.4.4 Korean Road Sign Management*

Despite the previous mashups, where Semantic Web technologies were involved mainly for the integration of other AI techniques, in the last mashup of this section we present an application where all the data processing is done using the Semantic Web.

### 10.4.4.1 Mashup Overview

A typical building in South Korea is described by the administrative divisions[19] in which it lies rather than by street names.

Figure 10.10 shows a typical road sign post on a major street of southern downtown in Seoul. The road sign post is installed at the side of the road with a massive structural support and a huge plaque hanging over the road; this provides rather excessive details on road guide information.

[19]Cf. http://en.wikipedia.org/wiki/Administrative_divisions_of_South_Korea.

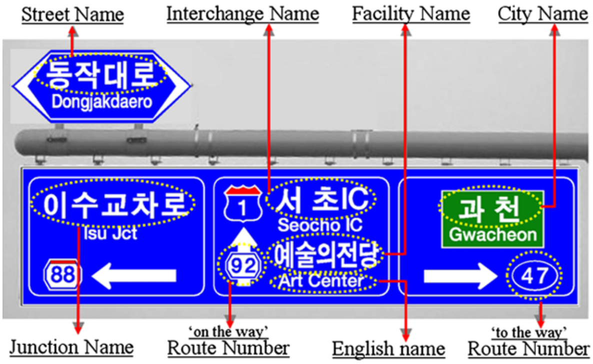



**Fig. 10.10** Typical Korean road sign

If the address is written in Korean, the largest division will be written first, followed by the smaller divisions, and finally the building and the recipient, in accordance with the East Asian addressing system. Divisions could be identified after the name of the nearest point of interest (POIs can be schools, police stations, hospitals, local parks and tourist points, etc.). In addition, it is mandated by the regulation to include English translations for all details specified on the road signs.

The problem is that Korean cities grow and evolve much faster than western cities. POIs may move, new roads may be built, and road signs may be changed accordingly. Effectively managing road signs, in particular validating if a sequence of road signs leads to a given address, is a major problem. For this reason the Korean Road Traffic Authority maintains a database of all Seoul road signs. The directions given on each road sign are formally described together with their actual location. The KRSM mashup presented in this section combines data from several data sources to validate Seoul road signs and to identify the invalid ones [31].

#### 10.4.4.2 Mashup Analysis

A summary of the main characteristics of BOTTARI is offered in Table 10.5.

The *stakeholders* of this mashups are the road traffic authorities: they manage the road signs in the cities, so on the one hand they are the main data providers—they own the datasets with road signs position and their contents, on the other hand they are the final users of the application—they have interest in checking the consistency of the reported directions.

The mashup considers four *datasets*: Open Street Map, Linked Geo Data, Road sign database from the Korea Institute of Construction Technology (KICT) and

**Table 10.5** Korean Road Sign Management according to the defined dimensions

| Stakeholders | Mashup Data | Mashup Processing | Mashup Delivery |
|---|---|---|---|
| Municipalities, Road traffic authorities | Maps (Open Street Map), POIs (OSM, closed dataset) and Road signs (closed dataset) | Semantic Web | Web application and SPARQL end-point |

a Korean POI dataset owned by the Korean company involved in this mashup development—Saltlux.

The OSM dataset contains POIs, roads and their related information within Seoul area. The data are retrieved through Open Street Map API and it is formatted in XML. The mashup considers about 100,000 nodes and 5,000 links.[20] As explained above, LinkedGeoData (LGD) is an RDF dataset derived from the OpenStreetMap. The Seoul area has been extracted using a geographical query: the resulting dataset contains 79,000 triples describing the nodes (each node is annotated with WGS84 coordinates and both Korean and English names). The Korean road sign (KRS) dataset is owned by the Korea Institute of Construction Technology (KICT). In the Seoul area there are 9,514 road signs and those data are contained in an Excel file. For each road sign the dataset contains: the coordinates, the POIs reported in the sign and the directions to reach them. The Saltlux Korean POI (KPOI) dataset contains 67,724 POIs within Seoul area and it is available as a relational database. For each POI an ID, the Korean name and the WGS84 coordinates are available.

The mashup process involves three main steps: conversion, integration and analysis. While LGD offers data in RDF, the others three datasets do not; so an initial conversion phase to extract RDF data is required to let the mashup use RDF as common data format:

- OSM: the RDF data were extracted through XSL transformations;
- KPOI: an RDBMS2RDF tool was employed (D2R);
- KRS: a custom extractor was developed in order to process the data from an Excel file.

After the format conversion there is a second conversion phase, in which the system converts the coordinates of the KRS datasets. In fact this dataset contains latitudes and longitudes expressed with regards to the Transverse Mercator coordinate system,[21] while the other three datasets use the WGS84 coordinate system. For the conversion an external service was used, the Korean Yahoo coordinate converting API.[22] The RDF data were stored in SOR, a triple store with geographical extension built on the top of BigOWLIM.

After the data conversion the integration step uses a mediation ontology to integrate the data from the different sources. The ontology is illustrated in Fig. 10.11: it contains eight classes, 29 properties and four axioms. Roads are modeled as a sequence of nodes and links. Four types of node are modeled: the generic nodes that can identify either a junction between multiple roads or a bend in a road; the road sign (RS) nodes that indicate the presence of a road sign; the Korean POIs (KPOI) that indicate POIs from the Korean Road Traffic Authority database; and the Wikipedia POIs (WPOI) that indicate POIs from Wikipedia (obtained through

[20] The amount of data available in June 2010, when the mashup was realized.

[21] Cf. http://en.wikipedia.org/wiki/Transverse_Mercator_projection.

[22] Cf. http://kr.open.gugi.yahoo.com/service/coordconverter.php.

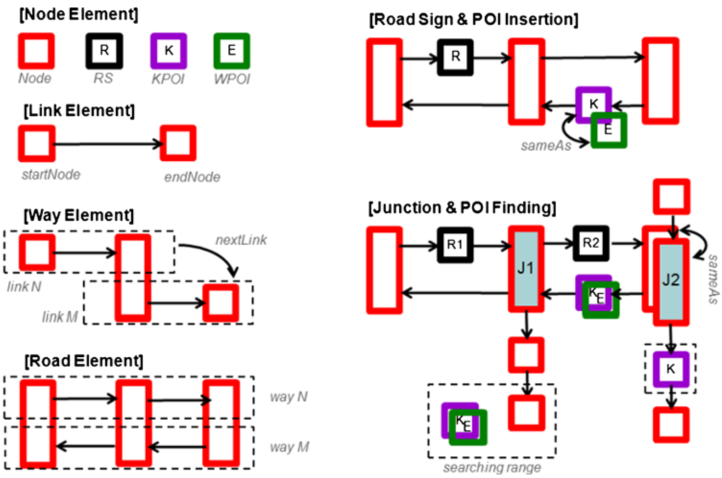



**Fig. 10.11** KRSM mediation ontology

DBpedia). A way is composed of links. A road is composed of ways. Road signs and points of interest are placed along the roads. If KPOIs and WPOIs are recognized to be the same, an `owl:sameAs` link is used to state it. Due to quality issues in the OSM data set, not all the junctions are explicitly stated; where necessary owl:sameAs is also used to state that two nodes are the same node and, thus, that a junction connects among multiple roads. Finally, not all POIs are directly on the roads—some of them may be placed nearby a node in the road.

The last processing step is the analysis of the imported data. The system uses OWL Horst reasoning [36] and SPARQL queries to validate the road signs and identify the ones containing wrong information. Figure 10.12 shows an example of reasoning over the road signs. At the top of the figure the data instances are represented: sign R1 indicates that two POIs (G1 and G2) can be found straight ahead; R2 indicates that POIs G2 and G3 are straight ahead, while G1 can be reached by turning right; R3 indicates that G3 is straight ahead, while G2 can be reached by turning right. In the three boxes (one for each road sign) the reasoning tasks are reported:

- R1 directions: going straight in two nodes it is possible to find R2, which contains further indications for G1 and G2, so both directions are valid;
- R2 directions: direction for G1 is valid because turning right is possible to reach G1, similarly directions for G2 and G3 are valid because going straight is possible to reach R3, which contains indications for them; similarly;
- R3 directions: direction for G2 is valid, going straight is possible to reach G2; direction for G3 is not valid: G3 is reachable only by executing a U-turn.

It is worth noting that also the direction for G3 on R2 is invalid, even if it seemed to be valid. In fact the direction refers to a road sign R3 which is not valid.

The system stores the results of the processing as RDF and offers a SPARQL end-point to query them. Additionally, a simple user interface was developed: it

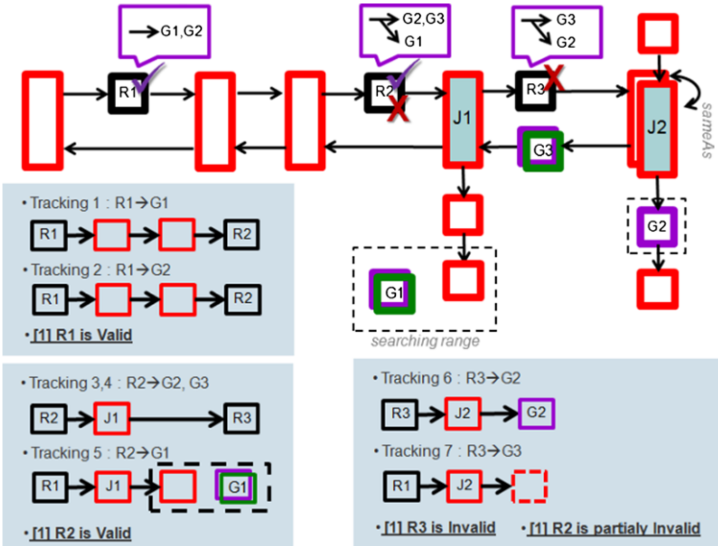



**Fig. 10.12** Examples of reasoning processes


**Fig. 10.13** Example of reasoning processes

is a Web application that presents the validation process using Google Maps as presentation layer. Figure 10.13 shows a screenshot in which there are two road signs, one containing correct information (the one with the star—the green one), while the other contains wrong directions.


**Fig. 10.14** Example of inconsistent data

### 10.4.4.3 Additional Details and Evaluation Results

One of the main problems that emerged while developing the Korean Road Sign Management is the "noise" in the data, with a particular focus on Open Street Map [28]. Some of those data quality issues are strictly related to the urban context and are general enough for mashup developers who works on this domain, so in the remaining of this section we will present them.

The first issue is the most common one: *inconsistent data*. When contributors insert wrong information, the knowledge represented in the map does not reflect the reality.[23] Figure 10.14 reports an example where Junctions 1 and 2 are misplaced; the result is that while in reality there are two roads with a junction where they cross, in the OSM representation there is a road (the one on the left) that splits into other two (the one going to the top and the one going left).

The second issue is related to *duplicated data*: it happens when the same location is assigned to different nodes. Figure 10.15 shows the same junction assigned to two nodes: the result is that the junction on the top can be related to both of them, even if in reality only one way exists.

Another issue is the *data incompleteness*. It happens when two ways cross each other, but they are not connected by any junction. The missing junction can mean two things: in reality the junction is missing—for example there is a bridge—or the junction exists and the representation is wrong. It is possible to find an example of this issue in Fig. 10.16.

Data quality issues are an active topic in the Open Street Map community; a list of initiatives to identify bugs and improve the quality of OSM data is available in the

[23]For the sake of clarity we assume that the Google map representation is correct.

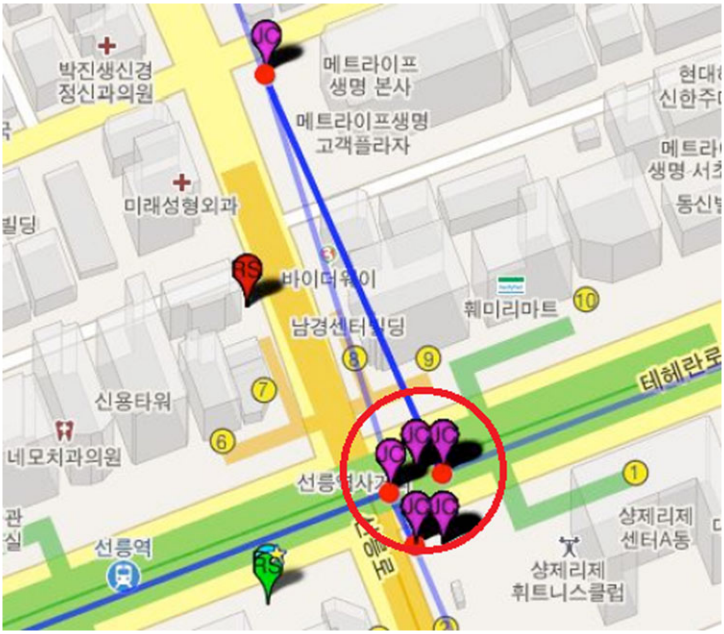



Fig. 10.15 Example of duplicated data


Fig. 10.16 Example of partial data

OSM wiki.[24] In literature the automatic identification of missing junctions in Open Street Map has been investigated by [35]. Nevertheless mashup designers may care about this kind of problems when working with those kinds of datasets.

## 10.5 Discussion, Lessons Learned and Future Prospects

Mashup developers can find in the urban context an exciting playground to develop their applications. In this chapter we presented the urban computing context, explaining how this scenario offers all the required elements to build interesting mashups: huge amounts of distributed and heterogeneous data, existence of stakeholders' needs, open problems to be solved. By providing some mashup examples we illustrated how techniques from different disciplines (Artificial Intelligence, Semantic Web, Natural Language Processing, etc.) can be combined in order to identify, integrate and process the input data to create new data with an added value for the urban users.

To sum up the contribution of this chapter we propose some of the open issues and challenges [16] that urban mashup developers can find when designing an application. We present four main classes of problems: heterogeneity, data-scale, time-dependency, and wrong data.

### *10.5.1 Heterogeneity Issues*

Dealing with heterogeneous data has been a challenge for a long time in many areas in computer science and engineering.

*Representational Heterogeneity* means that data are represented by using different specification languages. Urban Computing-related data can come from different and independent data sources, which can be developed with traditional technologies and modeling methods (e.g., relational DBMS) or expressed with "semantic" formats and languages (e.g., RDF/S, OWL, WSML); for example, geographic data are usually expressed in some geographic standard,[25] events details are published on the Web in a variety of forms, traffic data are stored in databases etc.

The integration and reuse of those data, therefore, need a process of conversion/translation for the data to become useful together. In the mashups we described in the previous section this problem was partially addressed: RDF was adopted as interchange format to link the data retrieved from different input data sources. When possible, existing tools—such as D2R to expose data bases as RDF stores—were employed. In other scenarios ad-hoc solutions were developed—for example when extracting data from CSV and Excel documents. In both cases, the conversion

[24] Cf. http://wiki.openstreetmap.org/wiki/Quality_Assurance.

[25] Cf. http://en.wikipedia.org/wiki/Geographic_Data_Files.

phase usually requires a sensible human effort: domain knowledge, comprehension of source and target schemata, development of the transformation algorithms/rules are tasks that automated agents can hardly cope with.

### 10.5.2 Data-Scale Issues

The advent of Pervasive Computing and Web 2.0 technologies led to a constantly growing amount of data about urban environments, like information coming from multiple sensors (traffic detectors, public transportation, pollution monitors, etc.) as well as from citizens' observation (black points, commercial activities' ratings, events organization, etc.). The result, however, is that the amount of data available to be used and integrated is hardly manageable by state-of-the-art technologies and tools, thus a severe focus on scalability issues must be taken into account. For example, intelligent methods for data sampling or selection should be adopted before employing traditional reasoning techniques, e.g. to select traffic data to employ in predictions.

Although we encounter large-scale data which are not manageable, it does not necessarily mean that we have to deal with all of the data simultaneously. A possible way to address this problem is to develop components that select the data relevant for the processing among the available one. When the data can be treated as streams, stream processing techniques can be used; BOTTARI uses Stream Reasoning [17] to process the Twitter data, identifying the relevant data and querying them.

### 10.5.3 Time-Dependency Issues

Knowledge and data can change over time. For instance, in Urban Computing scenario names of streets, landmarks, etc. change very slowly, whereas the number of cars that go through a traffic detector in five minutes changes very quickly.

Traffic LarKC exploits RDF named graphs [8] to cope with this issue: the data are partitioned along the temporal axis. The time-independent data[26] are stored in one named graph and is used in each computation. The time-dependent data are collected in several RDF named graphs, one for each time interval of validity: in other words we build a timestamped graph containing all the predicted travel times attached to links. The graph timestamp is then used to easily identify and select the relevant graphs to be used in the processing.

[26] It is an assumption for the data that do not change or change very slowly—city topography and calendars.

### 10.5.4 Data Quality Issues

As introduced in Sect. 10.4.4 when developing an urban mashup it is important to consider problems related to the quality of considered datasets. Some examples are

- Noisy data: a part of data is useless or semantically meaningless;
- Inconsistent data: parts of data are in logical contradiction with each another, or are semantically impossible;
- Uncertain data: the semantics of data are partial, incomplete, not clearly defined.

Traffic data are a very good example of such data. Different sensors observing the same road area give apparently inconsistent information. For example, a traffic camera may say that the road is empty whereas an inductive loop traffic detector may tell 100 vehicles went over it. The information of both sides may be coherent if one considers that a traffic camera transmits an image per second with a delay of 15–30 seconds, whereas a traffic detector tells the number of vehicles that went over it in 5 minutes and the information may arrive 5–10 minutes later.

In Korean Road Sign Management the data quality issues influenced the results of the application, so different techniques have been introduced to identify the errors and fix/avoid them [29]. In UrbanMatch the goal is the improvement of the data quality through a game. GWAPs, and in general human computation can be considered a method to improve the quality of the datasets used in mashups.

**Acknowledgements** This research was partially funded by the EU LarKC Project (FP7-215535). We would like to thank the project partners for their collaboration and in particular: Stefano Ceri, Tony Lee, Volker Tresp and Frank van Harmelen.

## References

1. Auer S, Lehmann J, Hellmann S (2009) LinkedGeoData—adding a spatial dimension to the web of data. In: Proceedings of ISWC 2009
2. Balduini M, Celino I, Dell'Aglio D, Della Valle E, Huang Y, Lee T, Kim S-H, Tresp V (2012) BOTTARI: an augmented reality mobile application to deliver personalized and location-based recommendations by continuous analysis of social media streams. J Web Semant 16:33–41. doi:10.1016/j.websem.2012.06.004
3. Barbieri DF, Braga D, Ceri S, Della Valle E, Grossniklaus M (2010) C-SPARQL: a continuous query language for RDF data streams. Int J Semant Comput 4(1):3–25
4. Barbieri DF, Braga D, Ceri S, Della Valle E, Grossniklaus M (2010) Incremental reasoning on streams and rich background knowledge. In: Proceedings of ESWC2010
5. Barbieri DF, Braga D, Ceri S, Della Valle E, Huang Y, Tresp V, Rettinger A, Wermser H (2010) Deductive and inductive stream reasoning for semantic social media analytics. IEEE Intell Syst 25(6):32–41
6. Bassoli A, Brewer J, Martin K, Dourish P, Mainwaring S (2007) Underground aesthetics: rethinking urban computing. IEEE Pervasive Comput 6(3):39–45
7. Bizer C, Heath T, Berners-Lee T (2009) Linked data—the story so far. Int J Semant Web Inf Syst 5(3):1–22 (special issue on Linked Data)
8. Carroll JJ, Bizer C, Hayes PJ, Stickler P (2005) Named graphs, provenance and trust. In: WWW, pp 613–622

9. Celino I, Dell'Aglio D, Della Valle E, Grothmann R, Steinke F, Tresp V (2011) Integrating machine learning in a semantic web platform for traffic forecasting and routing. In: Proceedings of the 3rd international workshop on inductive reasoning and machine learning for the semantic web (IRMLeS 2011), collocated with the 8th extended semantic web conference, ESWC2011, Heraklion
10. Celino I, Dell'Aglio D, Della Valle E, Huang Y, Lee T, Kim S-H, Tresp V (2011) Towards BOTTARI: using stream reasoning to make sense of location-based micro-posts. In: Garcia-Castro R et al (eds) ESWC 2011 workshops. Lecture notes in computer science, vol 7117. Springer, Heidelberg, pp 80–87
11. Celino I, Contessa S, Corubolo M, Dell'Aglio D, Della Valle E, Fumeo S, Krüger T (2012) Linking smart cities datasets with human computation—the case of UrbanMatch. In: Proceedings of 11th international semantic web conference 2012, Boston
12. Chan KT, King I, Yuen M-C (2009) Mathematical modeling of social games. In: Proceedings of IEEE CSE conference 2009, vol 4, pp 1205–1210
13. Chapelle O, Schölkopf B, Zien A (eds) (2006) Semi-supervised learning. MIT Press, Cambridge
14. Cheptsov A, Assel M, Gallizo G, Celino I, Dell'Aglio D, Bradeško L, Witbrock M, Della Valle E (2011) Large knowledge collider. A service-oriented platform for large-scale semantic reasoning. In: Proceedings of international conference on web intelligence, mining and semantics (WIMS 2011)
15. Davis CA, Vieira Monteiro AM (eds) (2007) Advances in geoinformatics. VIII Brazilian symposium on geoinformatics, GEOINFO 2006, Campos do Jordão (SP), November 19–22, 2006. Springer, Berlin, 315 pp
16. Della Valle E, Celino I, Dell'Aglio D, Kim K, Huang Z, Tresp V, Hauptmann W, Huang Y, Grothmann R (2008) Urban computing: a challenging problem for semantic technologies. In: NeFoRS 2008 workshop on new forms of reasoning for the semantic web: scalable, tolerant and dynamic, co-located with ASWC 2008—the 3rd Asian semantic web conference, Bangkok
17. Della Valle E, Ceri S, van Harmelen F, Fensel D (2009) It's a streaming world! Reasoning upon rapidly changing information. IEEE Intell Syst 24(6):83–89
18. Della Valle E, Celino I, Dell'Aglio D (2010) The experience of realizing a semantic web urban computing application. Trans GIS 14(2):163–181
19. Della Valle E, Celino I, Dell'Aglio D, Grothmann R, Steinke F, Tresp V (2011) Semantic traffic-aware routing using the larkc platform. IEEE Internet Comput 15(6):15–23
20. Dourish P, Anderson K, Nafus D (2007) Cultural mobilities: diversity and agency in urban computing. In: Baranauskas MCC, Palanque PA, Abascal J, Barbosa SDJ (eds) INTERACT (2). Lecture notes in computer science, vol 4663. Springer, Berlin, pp 100–113
21. ESRI (1998) ESRI shapefile technical description. http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf
22. Schieck AF, Lopez de Vallejo I, Penn A (2005) Urban space and pervasive systems. In: The seventh international conference on ubiquitous computing, September 2005
23. Fullerton T (2008) Game design workshop, a playcentric approach to creating innovative games. Gama network series. Morgan Kaufmann, San Mateo
24. Giffinger R, Fertner C, Kramar H, Meijers E (2007) City-ranking of European medium-sized cities. Centre of Regional Science, Vienna UT
25. Goodchild M (2007) Citizens as sensors: the world of volunteered geography. GeoJournal 69(4):211–221
26. Greenfield A (2006) No boundaries: the challenge of ubiquitous design. Adobe Design Center
27. Heath T, Bizer C (2011) Linked data: evolving the web into a global data space. Synthesis lectures on the semantic web. Morgan and Claypool, San Francisco
28. Huang Z, Fang J, Park S, Lee T (2011) Noisy semantic data processing in Seoul road sign management system. In: Proceedings of the 10th international semantic web conference (ISWC2011), Bonn
29. Ji Q, Gao Z, Huang Z (2011) Reasoning with noisy semantic data. In: ESWC (2), pp 497–502

30. Kindberg T, Chalmers M, Paulos E (2007) Guest editors' introduction: urban computing. IEEE Pervasive Comput 6(3):18–20
31. Lee TK, Park S, Huang Z, Della Valle E (2010) Toward Seoul road sign management on LarKC platform. In: Polleres A, Chen H (eds) ISWC posters and demos. CEUR workshop proceedings, vol 658. CEUR-WS.org
32. Markowetz A, Chen Y-y, Suel T (2005) Design and implementation of a geographic search engine. In: WebDB
33. O'Neill E, Kostakos V, Kindberg T, Schieck AF, Penn, A, Danae Stanton F, Jones T (2006) Instrumenting the city: developing methods for observing and understanding the digital cityscape. In: Dourish P, Friday A (eds) Ubicomp. Lecture notes in computer science, vol 4206. Springer, Berlin, pp 315–332
34. Open Knowledge Foundation. The open data handbook. Technical report
35. Scheider S, Possin J (2012) Affordance-based individuation of junctions in open street map. J Spat Inf Sci 4(1):31–56
36. ter Horst HJ (2005) Completeness, decidability and complexity of entailment for RDF schema and a semantic extension involving the OWL vocabulary. J Web Semant 3(2–3):79–115
37. Tresp V, Huang Y, Bundschus M, Rettinger A (2009) Materializing and querying learned knowledge. In: Proceedings of IRMLeS 2009
38. von Ahn L (2006) Games with a purpose. Computer 39(6):92–94
39. von Ahn L (2008) Human computation. In: Alonso G, Blakeley JA, Chen ALP (eds) ICDE. IEEE Press, New York, pp 1–2
40. Weiser M (1991) The computer for the twenty-first century. Sci Am 265(3):94–104
41. Zimmermann HG, Neuneier R (2001) Neural network architectures for the modeling of dynamical systems. In: Kolen JF, Kremer S (eds) A field guide to dynamical recurrent networks. IEEE Press, New York, pp 311–350

# Chapter 11
# Travel Mashups

**Amparo E. Cano, Aba-Sah Dadzie, and Fabio Ciravegna**

**Abstract** Web 2.0 has revolutionized the way users interact with information, by adding a vast amount of services, where end users explicitly and implicitly, and as a side effect of their use, generate content that feeds back into optimization of these services. The resulting (integrated) platforms support users in and across different facets of life, including discovery and exploration, travel and tourism. This chapter discusses the creation and use of *Travel Mashups*, defined based on the varied travel information needs of different end users, spanning temporal, social and spatial dimensions. The Web is presented in this chapter as a platform for bridging these dimensions, through the definition and use of composite, web- and mobile-based services. We examine the state of the art in existing mashups in the field, in addition to other relevant applications and the services available to build these, leading to a discussion of the multiple perspectives taken in creating such mashups. Based on this analysis we present a generalized architecture for Travel Mashups, from which we identify areas where opportunities exist to improve on the services currently available to the end user. The chapter concludes with a brief description of a scenario that elicits the information needs of an end user exploring an unfamiliar location, and demonstrates how the *Topica* Travel Mashup leverages social streams to provide a topical profile of Points of Interest that satisfies these user's requirements.

## 11.1 Introduction

The overall aim of Travel Mashups is to provide up-to-date, context- and location-relevant information about a traveler's current information needs, in order to support them in making informed travel and exploration decisions. Context-relevant,

A.E. Cano (✉) · A.-S. Dadzie · F. Ciravegna
The OAK Group, Department of Computer Science, The University of Sheffield, Sheffield, UK
e-mail: ampaeli@gmail.com

A.-S. Dadzie
e-mail: a.dadzie@dcs.shef.ac.uk

F. Ciravegna
e-mail: f.ciravegna@dcs.shef.ac.uk

B. Endres-Niggemeyer (ed.), *Semantic Mashups*,
DOI 10.1007/978-3-642-36403-7_11, © Springer-Verlag Berlin Heidelberg 2013

personalizable, portable Travel Mashups are made possible in the modern technology-rich world largely due to ubiquitous, affordable mobile and wireless network technology. From London to Tibet to the Azores and back again, users have access to mobile and other wireless networks, which they are able to connect to seamlessly from their personal computing devices. With advances in technology and an increase in its affordability, the average end user is able to access a variety of small, portable devices, from feature phones to smartphones, to tablet PCs (Personal Computers) [18], able to connect to communication networks as the traveler roams between different locations, close to their home or across borders.

Nearly ubiquitous use of these devices has resulted in unprecedented participation on web platforms, with attendant masses of user-generated information, created explicitly or as a side effect of interaction with these platforms. By providing easy access to different methods for capturing and sharing information about their environment and current activities, a phenomenon has arisen that converts travelers into new nodes of interaction with and consumption of context-rich information. This creates the potential for all kinds of new services which leverage this user-generated content to provide peer-to-peer web applications, with dynamically updating information.

A long tradition of documenting and sharing activities and experiences in the physical world is increasingly being moved to the infinitely more flexible and wide-reaching on-line world. Focusing specifically on travel, information is shared on-line by end users via travelogues; blogs; wikis (e.g., *Wikitravel*[1]); on-line photograph repositories (e.g., *Flickr*[2] and *Instagram*[3]); on-line video repositories (e.g., *YouTube*[4]); geo-social platforms (e.g., *Foursquare*[5]); social networking sites (e.g., *Facebook*,[6] *Twitter*[7]); on-line review and recommendation sites (e.g., *Tripadvisor*, *Revyu*[8]). While each of these platforms and other relevant *apps* (applications) are independently created and built, a simple way in which they are easily connected is through the orchestration of relevant services that cover a particular travel information need.

We discuss in this chapter the amalgamation of this user-generated content and a variety of services to construct Travel Mashups. Section 11.2 looks at the information needs of end users in the Travel domain. Section 11.3 introduces a categorization of Travel Mashups based on a mashup's target group and the mobility features supported by it. This section also presents an analysis of the features provided by state-of-the-art Travel Mashups. Section 11.4 introduces a generalized three-layer

[1] http://www.wikitravel.org.

[2] http://www.flickr.com.

[3] http://instagr.am.

[4] http://www.youtube.com.

[5] https://foursquare.com.

[6] http://www.facebook.com.

[7] http://twitter.com.

[8] http://revyu.com.

architecture for Travel Mashups, and discusses perspectives and opportunities for future Travel Mashups. The chapter closes with the presentation of a use case, the *Topica* Travel Mashup, which leverages social streams in order to provide a topical profile of Points of Interest in a specified location.

## 11.2 Travel Mashups in a Nutshell

*Travel* is defined in the on-line Free Dictionary[9] as: "*the movement of people or objects (e.g. trains, cars, and other conveyances) from one geographical location to another*". The distance between the departure and destination points, and the crossing of geographical and political boundaries characterize a journey as local, regional, domestic or international. While these all have a number of similarities with respect to the traveler's needs, each journey type is characterized by its peculiar information needs. For instance, formal, national identification (ID) such as a passport is normally not required for domestic travel, while such ID and sometimes also visas are needed to cross borders.

The factor that characterizes all travelers is that they often possess inaccurate, misinformed, incomplete, outdated, little or no information about the places they are visiting [9] and/or information on how to get there. This is in large part because the travelers are immersed in an environment that is in many cases new for them. Even for the (seasoned) traveler returning to a previously visited location, an element of uncertainty, due to (potentially) outdated knowledge, exists. This highlights the need for reliable, current, easy to interpret information. We discuss in this chapter the potential for Travel Mashups to bridge this information gap, by identifying travel information needs elicited by different travel scenarios.

### *11.2.1 Information Needs in Travel Mashups*

Travel information needs vary according to the reasons (e.g. business, tourism, volunteering) and motivations (including, e.g., pleasure, relaxation, and discovery) for travel. The relevance of travelers' information needs is transient, and commonly falls within a fixed temporal span, which covers the different stages of travel—before, during and after a trip [9]. The current context also plays a significant, albeit less overt, role in defining the traveler's needs. Identifying accurately travel information needs, as well as the time span of relevance, are the first steps toward generating a successful travel mashup.

Travel information needs vary across the three key stages of travel:

- *planning for a trip*—when travelers require information about their destination(s), the routes they can take to get there, and other requirements for successfully carrying out their travel plans;

[9] http://www.thefreedictionary.com/travel.

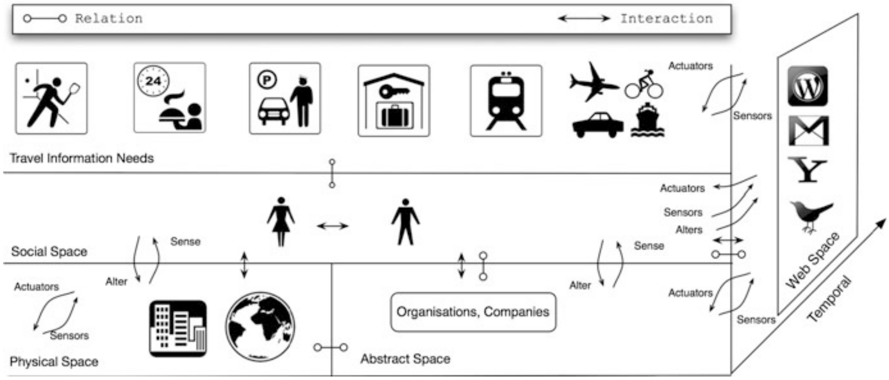



**Fig. 11.1** Travel information needs lie across different dimensions, merging social, spatial, and temporal features. Communication across these layers is achieved through the use of physical and virtual sensors which allows interaction between the traveler's physical and Web space

- *the actual journey*—when travelers require current information about POIs in their typically dynamically changing location;
- *post travel*—when travelers relive and share their experiences with both close members of their physical and on-line social networks, and other more distant individuals and groups.

These stages all share common features which lie across physical, abstract, social, and temporal dimensions (see Fig. 11.1), which may be bridged in Web space (or cyberspace) using sensors and actuators [29]:

1. *Physical Space*: refers to the environment in which a user (the traveler) is immersed; including all the physical entities they interact with. For example, (i) their geographical location—departure, current, final destination, as well as all other intermediate locations; (ii) services such as transport, accommodation, nearby restaurants and tourist attractions.
2. *Abstract Space*: refers to non-tangible entities e.g. (state or charitable) organizations (e.g. *Greenpeace*[10]) or commercial entities (e.g. travel agencies).
3. *Social Space*: refers to other users to which a traveler holds or can potentially hold a (informal) relation with, e.g. family, friends, or people with similar interests they have not met yet (other current, past or potential travelers, in this case).
4. *Temporal Space*: refers to the time span in which travel information needs are relevant to a user. This includes, e.g.,: (i) a specified extent of time prior to the start of the journey; (ii) the extent of time during which the journey takes place; and (iii) a specified extent of time after the journey has been completed (when

[10]Greenpeace, http://greenpeace.org.

the user has reached the final destination point—in the case of a round trip, this is the departure point).
5. *Web Space*: refers to the on-line world that bridges dimensions 1–4 above.

A *Travel Mashup* may then be defined as the combination of a set of web-based sensors and actuators that, based on a user's travel information needs, which may be weighted also by their personal preferences, retrieve relevant information from multiple sources and merge them such that they are presented to the end user as directly usable information. In this type of mashup users are able to interact with and immerse themselves more fully in their environment (i.e. physical space), by combining information coming from different types of sensors, both physical (e.g. GPS, mobile network hubs) and web-based (e.g. on-line social media streams).

We include here also *virtual sensors*, which allow the abstraction of data collection away from a fixed set of physical objects. Virtual sensor values are computed based on indirect or abstract measurements derived from multiple, distributed, often heterogeneous data streams [12, 19, 29]. A virtual sensor may define a number of valid sources of information, allowing it to poll for and retrieve information from different sources and at varying levels of granularity. The redundancy is useful for verification and validation of the information obtained from each source, and for providing backup sources of information.

A *web-based sensor* can be considered as an extension of the concept of virtual sensors, in which the measuring computation involves data streams generated from web resources [11]. Take, for example, the monitoring of flight prices to a particular destination, in which the aggregation of information coming from multiple on-line services triggers an event when the price falls below a pre-specified threshold. A *web-based actuator* may be regarded as a reactive computation that produces a response to a specified event. For example, if the price of a monitored flight drops to a desired price, an actuator service could automatically trigger a service that books this flight.

Another example of a virtual sensor, which makes use of a combination of multiple physical and on-line sensors to obtain information for one or more users, is alerting an individual to nearby "friends", by

- obtaining location information from the user's mobile device's GPS and/or connection to mobile or other wireless connections [*physical space*];
- polling their on-line social network to check for the status of friends in their social network [*social space, web space*];
- for those friends with status or calendar information "available" or "free time", respectively, checking their location information to determine who is nearby [*social space, temporal space*];
- polling for nearby locations that meet this group's preferences for, e.g., activities or events, in order to recommend suitable locations to meet [*physical space, abstract space*].

Travel Mashups leverage web-based sensors and actuators to merge the spatial, social and temporal dimensions (described above), in order to provide services that cover a specific travel information need. This, combined with the smartphone revolution, and the advent of the "Web of Things" [32], introduces a new platform able to meet the travel information requirements of end users in a dynamically changing world, by enabling ubiquitous Travel Mashups. We carry out in Sect. 11.3 a categorization of Travel Mashups and an analysis of the services they employ.

## 11.3 Mashups in the Travel Domain

Travel Mashups combine data, presentation and functionality from various external sources to create a service whose aim is to fulfill a specified travel information need. Section 11.3.1 presents a characterization of current Travel Mashups.

### *11.3.1 Characterization and Purposes of Existing Travel Mashups*

#### 11.3.1.1 Characterization on the Basis of the Target User Group

Mashups can be characterized according to the user group they target [6, 22]. How and what information is presented to the end user is also influenced by whether the target is the general public—an individual or a group traveling together—in which case *consumer mashups* apply. Such mashups aim to assist the traveling public to make informed travel decisions, as consumers of commodities and services. Alternatively, from an enterprise perspective, the aim is to assist Travel Mashup developers to integrate their services with existing composite services—in this case *Enterprise Mashups* (i.e., business or commercial) apply.

On this basis we categorize Travel Mashups as:

1. *Consumer Travel Mashups*: This type of mashup is designed to provide a final service to an ordinary user who consumes this service at a personal level. This category can be further subdivided into:
   (i) *Collaborative Travel Mashups*, which target communities as their final users. Such mashups serve both as an information provider and as an information consumer, as these mashups leverage the information generated as a side effect of their use. Examples of these services are *Recommendation Travel Mashups*, which enrich their recommendation models based on users' preferences. One example is Schemer,[11] which, based on activity and interest-based schemas generated by a location-based community, recommends locally available services. In this case the mashup services are dynamically filtered by location. Schemer therefore also provides location-based mobility (see Sect. 11.3.1.2).

[11] Schemer: https://www.schemer.com.

(ii) *Personalized Travel Mashups*, which, by targeting individual users, provide tailored services to meet each user's travel information need based on the users' profile. Personalization is closely related to *Collaborative Travel Mashups*, since (multiple) user profile information is used to feed their service models.

The modern technology user is already largely familiar with the personalization of services in their physical and on-line activity [17, 21], in exchange for sharing personal information and other contextual information (e.g., location as captured by their personal devices) across systems and services. Personalization depends on user identification, which is commonly simplified by methods that allow the use of unique, persistent IDs through services such as OpenID [25], therefore improving the user's experience, while reducing their load, by removing the need to authorize and personalize each service accessed. This allows users' on-line activities, e.g. web browsing, on-line shopping, review and recommendation of services, social web media use [15, 20], to serve as information sources for user modeling. Further, the emergence of compelling social web platforms, such as Facebook and Twitter, have encouraged end users to pro-actively participate in information creation and enrichment; as a result, shaping their on-line personae and influencing their perception about how they are viewed by others (see, a.o., [26]).

Explicitly created profile information, such as a user's home and workplace (locations), age and occupation (characteristics), is fed into users' static profiles, while frequently changing information (e.g. their current geolocation) contributes to users' dynamic profiles. The overall user model is then enriched with historical information obtained by capturing users' activities in both the physical and the on-line worlds. This, along with current contextual information collected from the user's personal sensors (e.g., GPS, on-line social streams), provides support for personalization and recommendation of services to the user [1, 11]. In a Travel Mashup, location is a key contextual attribute for distilling information relevant to a user; a traveler with a preference for visiting museums and galleries would be recommended the Prado when in Madrid and the Metropolitan Museum when in New York.

2. *Enterprise Mashups*: Also referred to as *Business Mashups*, these concentrate on the delivery of a service at the level of software. In this case the final users are developers who make use of this service either as a building block for constructing more complex services or as an assembler service which enables developers to combine other applications [6].

#### 11.3.1.2 Characterization on the Basis of the Travel Mashup Mobility Features

The construction and spread of communication infrastructure is growing rapidly, as the technology backing it also becomes increasingly advanced, resulting in communication bandwidths which allow the streaming of real-time, rapidly changing information [13]. Mobility has become a central aspect of the digital society, as a result of increased affordability of smartphones, other small devices (e.g., Personal Digital Assistants—PDAs) and tablet computers. This has led to the introduction of location-aware services in Travel Mashups. Mashups in this area include scenarios for tracking travel paths, navigation and social networking [16, 23, 31].

Travel Mashups can then be categorized on the basis of their capability to support mobility. The *mobility* functional range can be subdivided into *user* and *location* mobility.

1. *User-Based Mobility*: Acting as nodes of interaction and consumption, users mirror the desire to share information and experiences in the real (physical) world on-line, through blogs, fora, and social awareness streams (e.g. via *Facebook*, *Twitter*) [5, 13, 14]. Users produce data streams through their mobile devices, providing information about their physical context (e.g. location) [21], as well as their digital environment (e.g. adding new friends to an on-line social network, tweeting about a tourist attraction or eatery just visited). Ubiquitous, personal, near permanently connected devices have led to significantly increased engagement with mobile and on-line services. End users now produce data streams through their mobile devices on a continuous basis, as they move from one location to another and interact with both the physical and the on-line worlds.

   One example of user-based mobility services is *Google Latitude*,[12] which tracks users location through out the day, based on this information it provides statistics on where the user spends their time, at the level of granularity of day, month or year. Google Latitude also provides information about a user's friends provided that their profiles are set as publicly available.
2. *Location Mobility*:

   - *Location-Based Mobility*: Geographical regions may be profiled not only by static information but also by real-time events which occur in a specified location. Such information exposes dynamic features that provide a spatio-temporal characterization of a physical region. Location-based mobility refers to the ability of a Travel Mashup to cope with these dynamic features. One example is the *'Outbreaks Near Me' Mashup*,[13] which provides international travelers with geo-location-based, real-time surveillance on a broad range of emerging infectious diseases. This mashup combines geo-location-filtered data from a variety of alert services including the *World Health Organization*,[14] *ProMED*,[15] *GeoSentinel*.[16]
   - *Location-Aware Mobility*: By location mobility we refer to the capability of a *Travel Mashup* to cope with changes of the services used in the mashup

[12] https://www.google.com/latitude/b/0.

[13] http://www.healthmap.org/en.

[14] http://www.who.int.

[15] http://www.promedmail.org.

[16] http://geosentinel.org.

according to changes in location. One example of a location-aware Travel Mashup is the *Pocket VillageTravel Mashup*,[17] which combines location-aware services with geospatial data to provide travel-related activities and advice on tours. Based on a user's input location this service collects data from services like *Rezgo*,[18] *TourCMS*,[19] and others, which in return provide different services featured by this input geo-location.

While the benefits of mobility in Travel Mashups can be clearly seen, the use of mobile devices for such *apps* is beset by a number of challenges, when compared to the use of laptops and desktops. These include, as already mentioned, higher cost of network access, especially when roaming. Considering the underlying hardware, the mobile device user must contend with small screen size, lower computing power, disk space, and memory. In addition to these, any design for mobile *apps* must also consider, in normal usage of mobile devices, a greater number of external distractions and constantly changing context—with an associated increase in cognitive load [2]. The result is that end users often abandon digital media for print (see [9]); doing so, however, means that users also lose the additional context and recency of information available via mobile, context-aware Travel Mashups and other related *apps*. Future Travel Mashups should benefit from the adoption of sensors delivered through, for example, wearable devices whose aim is to facilitate the integration of user-specific needs into the mashup, e.g., dietary restrictions due to health conditions matched to restaurants available in a location.

Section 11.3.2 provides a description of existing Travel Mashups by analyzing existing services available in distributed mashup repositories.

### *11.3.2 Categorical Distribution of Existing Travel Mashups*

In order to characterize the trends of the functionalities offered by existing Travel Mashups we present an analysis of the types of composite services used in Travel Mashups extracted from two main mashup repositories: (i) *Programmable Web*;[20] and (ii) *Yahoo Pipes*.[21]

As of February 2012, there were 496 and 598 mashups tagged as "Travel" on the *Programmable Web*[22] and *Yahoo Pipes*[23] websites, respectively. We found that in Programmable Web Travel Mashups used services from a pool of 95 unique composite services, while Yahoo Pipes used 96. Based on an analysis of each, we labeled

---

[17] http://www.pocketvillage.com.

[18] http://www.rezgo.com.

[19] http://www.tourcms.com.

[20] http://www.programmableweb.com.

[21] http://yahoopipes.com.

[22] http://www.programmableweb.com.

[23] http://pipes.yahoo.com.

**Table 11.1** Categories and subcategories are presented in descending ranked order, based on the number of composite service belonging to each

| Category | Subcategories | Examples |
|---|---|---|
| Search engines | General | search.yahoo |
| | Social streams | search.twitter |
| | Blog | api.technorati |
| | Location-based | yahoo-local-search |
| | Personalized | yahoo-my-web-search |
| Spatial | Maps | google-maps |
| | Geocoordinates | geocoder |
| | Places | google-places |
| | Traffic | yahoo-traffic |
| | Distance | google-distance-matrix |
| Media | Pictures | flickr |
| | Video | youtube |
| Feeds | General | feedburner |
| | News | rss.cnn |
| | Travel | feeds.tourcms |
| News | General | news.search.yahoo |
| Travel | Travel guides | lonelyplanet |
| | Booking services | orbitz |
| | Meta search | kayak |
| | Gov. travel agencies | travel.state.gov |
| | Tech. platforms | travolution |
| | Deals and offers | travelzoo.com |
| | Forums | thorntree.lonelyplanet |
| | Flight scanners | travenjoy |
| Reference | Question & answers | answers.yahoo |
| | Collaborative encyclopaedia | wikipedia |
| | By subject of interest | squidoo |
| | Social bookmarking | del.icio.us |
| Social | Social streams | twitter |
| | Messenger | msn |
| | Dating websites | speedate |
| Blogs | General | blogspot |
| Tools | Mashup service | pipes.yahoo |
| | Charts | google-charts |
| | Translation | google-translate |
| | Authentication | google-client-authentication |
| | Widgets | yahoo-widgets |
| | Visualization | fusion-tables |
| | Topic tracking | netvibes |

**Table 11.1** (*Continued*)

| Category | Subcategories | Examples |
|---|---|---|
| Shopping | General | amazon-ecommerce |
| | Advertising | google-adsense |
| | Location-based | yelp |
| | Online payments | stripe |
| Weather | General | weather-channel |
| Telephony | Voice and text messaging | twilio |
| | Mobile-network-based location | orange-location |
| Events | General | upcoming |
| Storage | General | amazon-s3 |
| Time | General | world-time-engine |

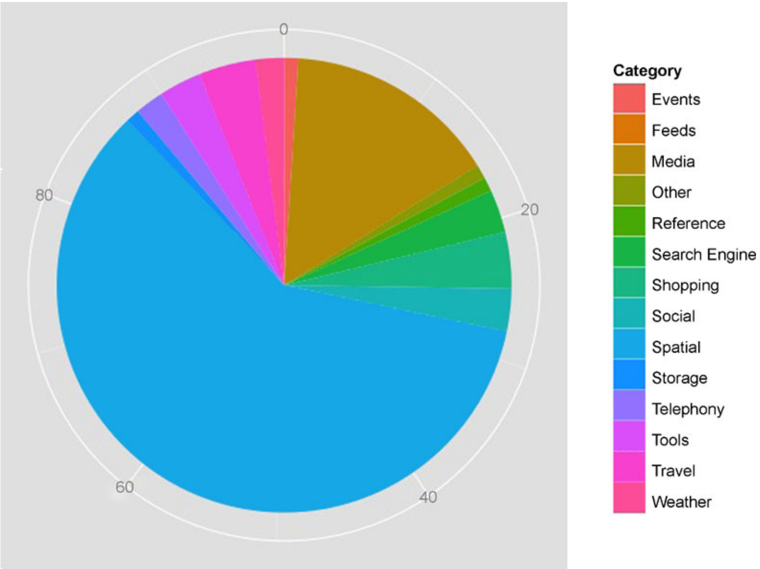



**Fig. 11.2** Categorical distribution of Travel Mashups

these according to the type of service they provide. The service categories most frequently used in these Travel Mashups are presented in Table 11.1 and Fig. 11.2.

These categories give an insight into current trends in the development of Travel Mashups (see Fig. 11.2). As expected the majority of existing Travel Mashups involve the use of services that provide spatial information (e.g. geocoordinate information and maps), followed by the use of media-related services (e.g. Flickr and YouTube). The use of advertising and shopping services is also quite common, showing an increasing number of Travel Mashups providing access to on-line payment services.

Travel Services are also widely used as composite services for retrieving, e.g., travel guides and booking services. Search engines, along with reference services, are also a very common composite service in Travel Mashups. Finally, the use of the Social Web shows the emerging appearance of social dating services in Travel Mashups.

From the point of view of the developer there is wide adoption of mashup service platforms (e.g. yahoo.pipes) for service composition. Particularly, rendering services (e.g. *Google Fusion Tables*[24]), which provide visualizations that adapt to the type of data provided in a Travel Mashup, are widely adopted.

Mobility requirements in Travel Mashups generate the need to store and access information ubiquitously. In travel scenarios, where users are immersed in unknown environments, the availability of their personal data at any time, independent of their current location, is essential. Some of the Travel Mashups analyzed include cloud storage as a composite service to tackle this issue.

There appears also to be an emerging trend toward the use of streaming services for collecting social as well as event-related information. Travel Mashups also make use of services that provide volatile information regarding locations, i.e., information that is relevant to locations only for a particular period of time (e.g. news, deals, and offers).

Ubiquitous access to information as well as dynamically evolving needs are among the key emergent characteristics of travel information needs in today's Digital Society. Taking into account the analysis presented here, Sect. 11.4 presents a generalization of a Travel Mashup architecture.

## 11.4 Travel Mashups—Design & Architecture

Although Travel Mashup architecture varies according to the type of information needs covered, a number of core architectural features exist that are common to all such mashups. Figure 11.3 depicts the fundamental structure of a Travel Mashup which consists of three layers: (i) the Information Sources Layer; (ii) the Travel Mashup Generation Layer; and (iii) the Presentation Layer, which build on each other.

### *11.4.1 Information Sources Layer*

The lowest layer, *Information Sources*, represents the Web services and resources that are the foundation of a mashup. These sources may be internal or external, and can include real-time information by means of streaming services and static information. These sources can be queried a priori (in the case of static or rarely

[24] http://www.google.com/fusiontables/public/tour/index.html.

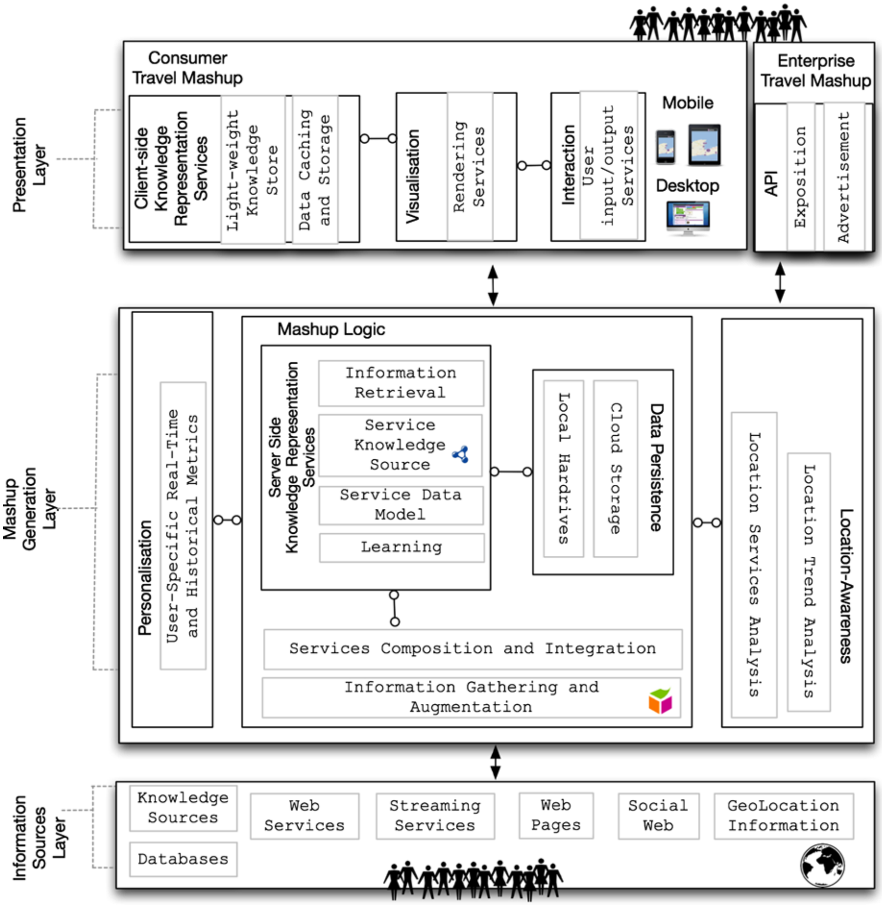



**Fig. 11.3** Generic three-layer Travel Mashup architecture

changing information), or on the fly. A common method for retrieving information from external sources is the use of REST services, which address web services as resources.

## *11.4.2 Travel Mashup Generation Layer*

This layer involves the orchestration and adaptation of services to a particular travel information need within a location-based setting. This layer consists of the following stages:

1. *Personalization*: This stage captures the characterization of a user. The user profile representation includes the analysis and representation of user-specific

features which can be based on historical (e.g. a user's address and language) or real-time information (e.g. a user's current location).

2. *Location Awareness*: While a regional space can be described statically based on a name and a set of geocoordinates, a dynamic representation of a location can be achieved through location profiling. This stage involves the process of profiling a regional area by means of internal (e.g. mashup's location profiling services) and external location services (e.g. external location trend services). This task involves collecting features that characterize a physical space. Processes in this stage include, e.g., location trend and topical analysis.
3. *Mashup Logic*: This stage consists of all the processes involved in the logic of the mashup. It can be subdivided into three main interdependent sections:
   a. *Information Gathering and Augmentation*: When data are retrieved from the Information Sources layer, further related information can be extracted from external knowledge sources like *Linked Data*.[25] The information augmentation stage communicates with the Knowledge Representation services which keep a local, up-to-date knowledge repository containing the information handled in the mashup.
   b. *Services Composition and Integration*: According to the type of services that a Travel Mashup provides, existing services will be composed and integrated. Service orchestration can be carried out statically where the service interfaces and available data types are known a priori; or dynamically where the available data are matched against the available services' interfaces. The input and output data generated by these services will be handled by the knowledge representation services.
   c. *Server-side Knowledge Representation Services*: This stage handles services related to data modeling, retrieval and storage of information, providing an interface to the data persistence stage and to the presentation layer. This stage also involves the execution of learning services which take information from the Travel Mashup's knowledge sources and carry out inferences over these data.

      This stage also handles the design of the Travel Mashup data model, which can be exposed as Linked Data. (Standard) schemas for exposing structured data from the Travel domain can be found at: (i) schema.org,[26] (ii) OpenTravel[27] community, and (iii) W3C's Points of Interest Core.[28]
   d. *Data Persistence*: This stage consists of services for persisting data either locally or by means of shared remote infrastructure (i.e., cloud storage). This stage is accessed through services provided in the Server-side Knowledge Representation Stage.

[25] W3C standards—Linked Data: http://www.w3.org/standards/semanticweb/data.

[26] http://schema.org/LocalBusiness.

[27] http://www.opentravel.org/Specifications.

[28] http://www.w3.org/2010/POI/documents/Core/latest.

In order to account for personalized travel information needs, the Personalization stage can be merged into the Mashup Logic. In the same way, if the Travel Mashup must account for changes in the travel information needs according to changes in a location feature before the Location-Awareness Stage can be plugged into the Mashup Logic. The majority of current Travel Mashups manage user and location profiling based on static attributes of their users and their locations. However, the recent increase in accessibility of Social and Location-based streaming services, which provide real-time changing features, encourages programmers to develop mashups that account for these dynamic features. This integration is achieved by merging information coming from the personalization and location-awareness stages into the Mashup Logic.

### *11.4.3 Presentation Layer*

The presentation layer is the interface between end users and the services provided by the mashup. The stages involved in this layer change according to the target group of users—consumer or enterprise (see Sect. 11.3.1)—to which the Travel Mashup is addressed.

#### 11.4.3.1 Consumer Travel Mashups

The presentation layer for Travel Mashups targeting private end users involves the use of a device (e.g., mobile or desktop), in which an application serves as the interface to the Travel Mashup services. The stages involved in the realization of these interfaces include: (i) Client-side Knowledge Representation Services; (ii) Visualization; and (iii) Interaction.

1. *Client-side Knowledge Representation Services*: This stage exposes services that act as a proxy to the Mashup Generation Layer. These services cope with the business logic related to client-side data management as well as the business logic required for retrieving information from the server-side services exposed by the Mashup Generation Layer.

   In rich, interactive Travel Mashups, users expect an application to respond quickly to interaction and transitions, especially where dynamic visualization is used. For this reason, some Travel Mashups employ client-side data management and caching, preload data before running the application. This also has the advantage of supporting off-line navigation services—often an issue during travel, when the user may not have access to an Internet connection.

   For Web-based applications client-side data management and caching is commonly achieved by means of HTML5 webstorage.[29] This API defines key-value pair data storage in web clients in which the expiration of the data is handled through user sessions or scripting.

[29] http://dev.w3.org/html5/webstorage.

For knowledge-based Travel Mashups, which deliver client-side, light-weight learning and inferencing based on information extracted from user interaction with the visualization, the use of light-weight knowledge repositories is also required. Although this particular setup is still an area of research, the use of HTML5 webstorage and RDFa-based triple stores such as *rdfquery*[30] is currently being implemented [10].

2. *Visualization Stage*, which involves the use of external or internal rendering services. Rendering services are a subset of the services involved in the application interface presented to the end user. The data passed to these (rendering) services are usually extracted through the services contained in the Client-side Knowledge Representation Services (see point 1). In the case of Web-based visualization, CSS-based designs are employed for visualizing information according to their data types. For example, *CloudMade Maps*[31] provide services that render location information using customized cartography styles. Another example is Google Fusion Tables, which provides services for visualizing data on maps, timelines, and charts.
3. *Interaction Stage*: Interaction in Travel Mashups acts as trigger of input–output services from the Mashup Generation Layer. The interaction stage is tightly coupled to the rendering services in the visualization stage, and with the use of client-side stored data. The services triggered by user interaction activities are generally performed through the use of *Asynchronous JavaScript over XML* (AJAX). The use of AJAX helps to reduce response time, and hence, improve interactivity. Touch and gesture interaction are also widely used in Travel Mashup applications. For HTML5-based Travel Mashup applications, the W3C specification for abstracting touch and finger gestures in HTML5, the *Touch Events API*[32] is recommended.

#### 11.4.3.2 Enterprise Travel Mashups

In contrast to the Consumer Travel Mashups, the Enterprise Travel Mashups consist of software services which not necessarily involve a visualization stage. Enterprise Travel Mashups are ready to use pieces of software which are exposed as services, or as a Travel Mashup enabler service. In the latter case the service may be a building block to other services. Providers of enterprise mashups often expose and advertise their service APIs using existing mashup repositories such as ProgrammableWeb and YahooPipes.

### *11.4.4 Toward User and Location-Aware Travel Mashups*

Despite the fact that wearable sensors are still not widely adopted, the design for future Travel Mashups suggests the continuous integration of user and location

[30] http://code.google.com/p/rdfquery.

[31] http://cloudmade.com.

[32] http://dvcs.w3.org/hg/webevents/raw-file/tip/touchevents.html.

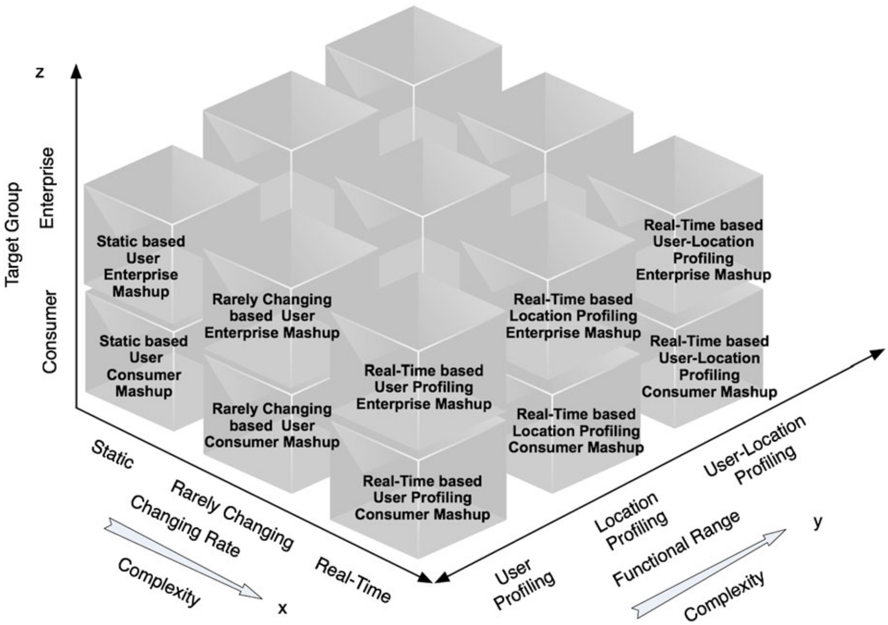



**Fig. 11.4** Classification of Travel Mashups based on their functional range and changing rate of information

dynamic features into mashup logic. On the basis of this integration, a Travel Mashup can be classified according to its functional range and the changing rate of information that the mashup's logic can support (see Fig. 11.4).

The majority of current Travel Mashups integrate static and rarely changing data on user and location profiles into their mashup logic stage. In Fig. 11.4 these mashups correspond to the

$$\text{cubes in coordinates } (x, y, z)$$
$$\text{where } x = \{\text{static, rarely changing}\},$$
$$y = \{\text{user profiling, location profiling}\},$$
$$\text{for } z = \{\text{Consumer, Enterprise}\}.$$

The integration of streaming data, which reshapes user and location profiles in real-time and feeds into a Travel Mashup logic, is still an area of research; particularly for knowledge-based Travel Mashups in which this integration involves the use of on-line learning algorithms [7, 24].

The complexity for delivering intelligent, real-time Travel Mashups increases as location and user awareness features are added to the mashup business logic. The goal for future Travel Mashups is to cope with advancing technology and new algorithms capable of supporting the merging of these features, in order to deliver services to meet increasingly complex and wide-ranging travel information needs.

This is exemplified in Fig. 11.4 by the coordinates $(x, y, z)$ where $x = \{\text{Real-Time}\}$, $y = \{\text{User–Location Profiling}\}$, and $z = \{\text{Consumer, Enterprise}\}$.

Section 11.5 presents the *Topica* Travel Mashup, as an example of a knowledge-based Travel Mashup. This consumer mashup uses on-line social activity streams as an information source for building a topic-based profile of locations.

## 11.5 *Topica* Mashup—Visualizing Emerging Semantics of POIs

The rise of ubiquitously available social media services has contributed to altering how people engage with their social context and environment. Particularly, social activity streams, which are a collection of semi-public, natural-language messages produced by different users and characterized by their brevity, have revealed new patterns of communication characterized by high social connectivity and their ability to communicate trends [3, 28].

The development of Social Web platforms that enable real-time posting of information suggests the use of humans as sensors. Citizen sensor networks is an emerging area of research in social computing, which regards the Social Web as a network of interconnected, participatory citizens who actively observe, report, collect, analyze and disseminate information about events and activities via text, audio and/or video messages in almost real-time [27].

The use of citizen sensing for Travel applications has resulted in a new generation of applications which demand real-time processing of very large amounts of data. One of the characteristics that make citizen sensing extremely useful for situational awareness in Travel Mashups is the different dimensions reported about a particular event occurring at a specified geographical location. Multiple perspectives of a single event can be derived by taking into account the information contained in the spatio-temporal metadata captured by the device from which a piece of content is posted.

In this section, we present the use of social stream aggregations as a data source that can convey meaningful, collective information for modeling dynamic characteristics of a POI. Although a POI is typically represented as a set of static data (e.g. name, address, geocoordinates), there are many latent (or hidden) features which can describe volatile and temporal aspects of it. For example, an Italian restaurant may be well known for good, hand-made ravioli, or for the (transient) two-for-one offer available during the current month.

We present *Topica*,[33] a mashup application based on social awareness streams that enhances collective information about a POI by leveraging structured data extracted from the Linked Data (LD) cloud. The *Topica* application provides the following contributions to the state of the art:

1. **Exposing Latent Features of POIs as Linked Data.** In order to achieve this we exploit the information available from a number of social media and web-based services:

[33] pulsar.dcs.shef.ac.uk/topica.

- *Facebook Locations*—the Facebook representation of a POI– and *Facebook Pages*[34] to retrieve information contributed by the ordinary end user and more official sources about events and activities related to a Facebook Location (a POI);
- *OpenCalais*,[35] *Zemanta*,[36] and *DBpedia Spotlight*,[37] to disambiguate and enrich entities extracted from social awareness streams, and enable further retrieval of related information from the LD cloud and other Semantic Web (SW) services [28].

2. **Mashup Visualization of POIs.** Each POI is presented with a set of topics, tags and messages, in addition to an address and a title. The visualization is created using the Prism framework,[38] which acts as a filtering interface. The framework enables synchronous filtering of POIs based on their facets (or attributes).

## *11.5.1 Application Scenario*

To illustrate our work consider the following scenario:

> *Alice, who grew up in Rethymnon, in Crete, has moved back to take up a new job in the city, after having been away for almost 10 years to study and then work in a different city. Alice wishes to revisit old haunts as well as explore what is new in Rethymnon. She starts with the city center and its immediate surroundings, to find out if the venues she often frequented in her late teens match her (evolved) interests. Recognizing also that some of the places she remembers may no longer exist, Alice decides to make use of an app suggested by a colleague for discovering venues and events in her (physical) neighborhood. This* `app`*, which she is able to access from a (desktop) web browser and also her new smartphone, allows her to browse current topical information about nearby POIs, based on (collective) information contributed by visitors to these POIs and also from more official sources (such as event organizers and venue managers).*

Although Alice is not, strictly, a traveler, her information needs in this scenario match those of a traveler seeking information about a selected location during the *the actual journey* stage of travel. We demonstrate the approach taken in *Topica* to realize this scenario, i.e., to support topical recommendation of POIs based on POIs' latent features. In order to do so, *Topica* analyzes publicly available social stream information—in particular Facebook—filtered by location—in this case the city of Rethymnon on the island of Crete, in Greece.

*Topica* models space based on the topical information buzzing in social media streams. From the city center and its immediate surrounds, location-based filtering

[34] http://www.facebook.com/about/pages.

[35] http://www.opencalais.com.

[36] http://www.zemanta.com.

[37] http://dbpedia.org/spotlight.

[38] http://evhart.online.fr/prism.

is performed to derive topical information. This information is aggregated to profile regional areas enabling *Topica*'s users to expand a topical search from a focus to include increasingly wider radii of regional interest.

From an SW point of view, *Topica* identifies the topics that model a POI by extracting DBpedia categories from (key) *entities* (e.g. `Location`, `Organisation`, `People`, `Places`) and *keywords* obtained from the POIs' related social awareness streams—in this case, an aggregation of Facebook comments regarding a POI. From Alice's point of view, *Topica* allows her to retrieve topical information of interest to her from the collective information obtained by aggregating comments contributed by users in the locality. She is able to select from these topics that match her interests—she chooses now to focus on music (*Jazz, World Music*) and food (*Italian*). *Topica* is then able to retrieve POIs by topic as well as the related current, valid concepts and categories that feature a POI.

Alice decides to try out one of the Italian restaurants suggested by *Topica* with a couple of friends. During the meal she posts a comment on Facebook Places about the restaurant, and comments on her main meal (ravioli with a creamy lobster sauce) and the live Jazz band playing. This information is consumed by *Topica*, which then reprofiles the regional area (in this case the POI) accordingly (to include, in addition to Italian_Cuisine—which prompted Alice to select this restaurant, also Sea_Food and Jazz_Music).

## *11.5.2 The Topica Application*

*Topica* facilitates the retrieval of POIs by characterizing them with latent topical features extracted from the collective perception of a POI over a period of time. As demonstrated in Sect. 11.5.1, *Topica* exploits social awareness streams to enrich POIs with structured data from the LD Cloud and provide a filter-based visualization of POIs.

### 11.5.2.1 Enriching POIs with Linked Data

The approach taken in *Topica*, illustrated in Fig. 11.5, encompasses the following steps:

(1) For a given geographically bounded area (i.e. a collection of geocoordinates contained within a delimited area) Facebook locations are retrieved.
(2) Using the Facebook location properties (e.g. `Name`, `Address`, `Description`) we align the location (POI) with a Facebook page.
(3) For each Facebook page, comments are extracted. These comments are enriched by querying the following services: OpenCalais, Zemanta, and DBpedia Spotlight. From these services, keywords, entities, and related pages are extracted.

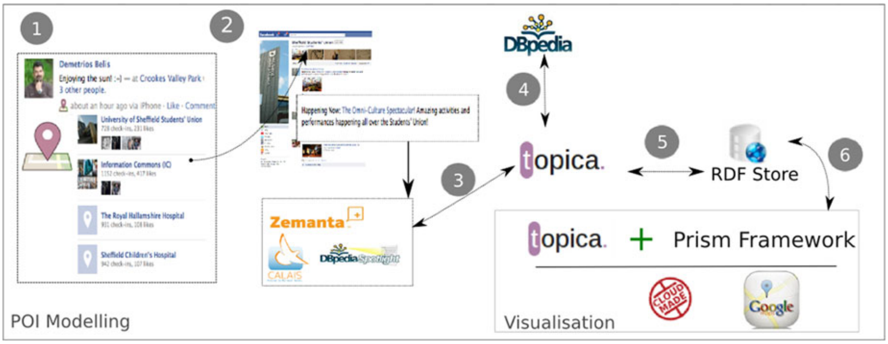



**Fig. 11.5** *Topica* architectural design

(4) The data collected (in points 1, 2, and 3) are used by *Topica* to generate a list of potential DBpedia resources. In order to model the topics of a comment, *Topica* uses this resource list to query DBpedia. For each resource DBpedia's categories and broader categories are extracted. For example, for the resource Ravioli, these categories include: Pasta and Italian_Cuisine; and broader categories for Italian_Cuisine includes: Mediterranean_Cuisine and Italian_Culture. The set of categories collected from the comments of a page are weighted following a tf-idf (term frequency–inverse document frequency) function.

(5) We encode POIs in a structured format—RDF—using shared vocabularies including: `SIOC`,[39] `SKOS`,[40] and `CURIO`.[41] Our ontology, `LinkedPOI`, takes advantage of `sioc:Container` and `curio:LocalisedItem` to model POIs as elements of a bounded area (`linkedPOI:Patch`). This allows SPARQL querying[42] for concepts featuring a bounded area.

#### 11.5.2.2 Map-Based Visualization

*Topica* makes use of the Prism Framework to enable synchronized semantic filtering of POIs. Prism is a JavaScript Semantic framework that allows the creation of synchronized semantic filters using a seed query. Prism relies on *OpenStreetMap*[43] and *Google Maps*.[44] It computes a SPARQL query against a set of filter parameters in order to select a subset of the objects returned by the seed query. *Topica* uses the following Prism filters:

[39] SIOC—Semantically-Interlinked Online Communities: http://sioc-project.org/ontology.

[40] SKOS—Simple Knowledge Organization System: http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20050510/.

[41] CURIO—Collaborative User Resource Interaction Ontology: http://purl.org/net/curio/ns.

[42] SPARQL Endpoint at http://nebula.dcs.shef.ac.uk/topica/sparql.

[43] http://www.openstreetmap.org.

[44] http://code.google.com/apis/maps/index.html.

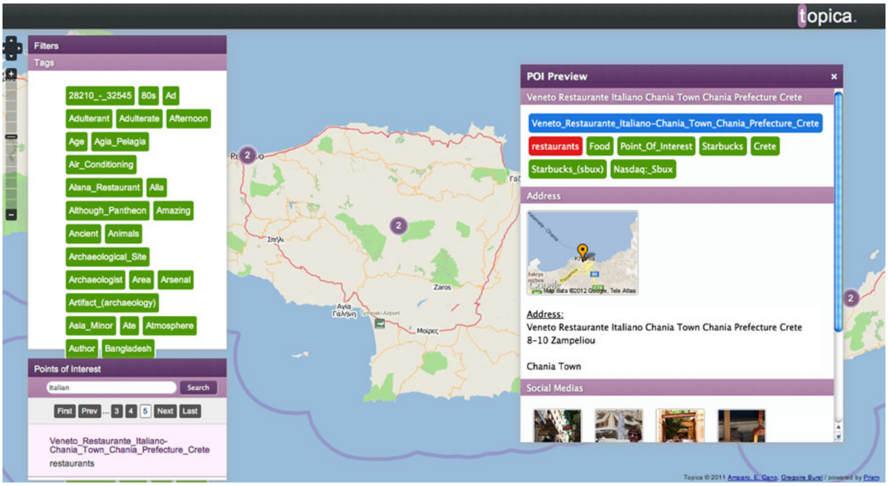



**Fig. 11.6** Overview of the Prism-rendered *Topica* user interface, showing sub-windows with additional detail for the POIs selected

(1) The *Location lens* is used to filter POIs according to their location, using geocoordinate information. This activity is supported through the use of a map widget.
(2) The *Tag lens* enables the selection of POIs according to their associated tags. This involves the generation of a geo-tag filter to allow the retrieval of POIs labeled by a given tag.
(3) The *Search lens* provides a text filter that operates on the POIs' messages. This filter extracts POIs based on comments in emerging social streams that contain the search text.

By employing the Prism framework *Topica* is able to support the user to retrieve information stored in the LD cloud using intuitive visualization. The visual representation provides a consistent and synchronized view of POIs according to the filter parameters provided by the user. When the user is satisfied with the values returned by the filters, they can access a description of a particular POI by clicking on a map object from the POI list. The pop-up visualization revealed provides a description of the POI. The description includes the name and address of the POI, its tags and topics, and the different social messages associated with the POI (Fig. 11.6).

### *11.5.3 Topica Mashup—Summary of Key Features*

*Topica* improves on existing POI retrieval *apps* in that *Topica*: (1) does not require end users to explicitly contribute information about events—the mashup extracts these from users' social interaction data; (2) as in [8], *Topica* enriches POIs, but goes beyond the work of Braun et al. by adding DBpedia resources extracted from comments related to POIs. By using SW services, such as OpenCalais and Zemanta, we provide even richer annotation and widen recall during information retrieval.

*Topica* was developed to support especially those end users who may have little to no knowledge about where to find information about nearby physical entities, but who are able to trigger search based on their interests. *Topica* caters to the modern user's expectations of ubiquitous technology, by exploiting the collective knowledge of crowds to satisfy overlapping information requirements. We plan to improve information retrieval by employing syntactical analysis of comments in order to filter out spam, and to obtain entities that are more relevant to POIs. *Topica* currently uses fixed snapshots in time for each of its location datasets (December 2010–January 2011)—the restricted dataset was to allow a focus on the design and the development of the backing technology. We aim, however, to move to dynamic batch update, to ensure that the fully working tool allows the end user to retrieve the most recent information about POIs, in addition to still valid historical information. Future work also includes the modeling of relevance decay functions for the latent features of POIs.

## 11.6 Related Work—Applications

Searching for information about POIs and events in a user's environment is an oft-performed activity. Support for meeting the traveler's (current and projected) information needs is typically available to them prior to embarking on a journey, in physical locations such as their travel agent, on a variety of web sites publishing information based on end-user experiences of places and services, in addition to vendor-supplied information (e.g. by tourism and culture ministries and other relevant state organizations, travel agencies, airlines, transfer services, hotels and other relevant hospitality services). However, as the traveler moves away from their home environment to less familiar or completely unknown locations their ability to access information becomes increasingly limited. This is due both to restricted and/or significantly more costly access to computing resources while on the move and little to no knowledge of alternative *bricks and mortar* services where they might find the information sought. Travel Mashups help to fill this gap by providing a one-stop shop, based on the current or other specified location, and in some cases, personal information about the end user, to provide tailored services to satisfy the user's information needs—through the planning stage and the actual journey.

Computing-based applications to support traveling increasingly make use of on-line resources that encourage end user (traveler) contributions by making use of SW and Web 2.0 technology [13, 14]. Well-known examples include *TripIt*,[45] *DBpedia Mobile*,[46] *Revyu.com*,[47] *MetaCarta*[48] and *CouchSurfing*.[49] Other on-line services

[45] http://www.tripit.com.

[46] http://mes-semantics.com/DBpediaMobile.

[47] http://revyu.com.

[48] http://www.metacarta.com.

[49] http://www.couchsurfing.org.

that make no (explicit[50]) use of SW services still remain popular with travelers—including *TripAdvisor*[51] and *Journeywoman*.[52] A notable advantage in the use of such resources, especially those that take advantage of SW technology to enrich user-contributed information, is the varied and rich contextual information contributed from multiple perspectives, by end users with actual experience of the locations and services they describe.

Hao et al. [14] demonstrate the use of information extracted from travelogues to obtain information specifically geared at the traveler, by other travelers. *DBpedia Mobile* [4] is a location-aware SW client, which, based on the user's GPS or IP (Internet Protocol) information, renders a map with nearby locations extracted from the DBpedia dataset. Icons are used to provide information about different entity types in the overview, and the application queries the LD cloud for additional information on the user's focus. Other applications, such as *Stevie* [8], allow users to share and browse temporal information about POIs—events—on a map visualization, based on the location broadcast by end users' GPS. Stevie annotates the information shared using an ontology, and by linking to corresponding entities in DBpedia.

*Automated Murmurs* [13] makes use of the contributions by end users from mobile, location-aware devices to provide context-sensitive information about POIs. While developed for mobile use, it also provides a (desktop) web browser version. Automated Murmurs encourages the attachment of media—video, audio and images—to the messages exchanged, as a way of providing additional information to the (default) text content. Information is exchanged via GPRS (General Packet Radio Service), or where not available, SMS (Short Messaging Service) and MMS (Multimedia Messaging Service). The service prioritizes push services to users in common social networks, in addition to relevance based on location and context. One added benefit in the application is support also for indoor use, predominantly via *Ubisense*,[53] in addition to GPS, to obtaining geo-location information. Automated Murmurs supports the submission of reviews about POIs and relevant services, and mapping and storing routes through its map interface. Finally, it also promotes the creation of new social contacts between co-located users.

MetaCarta, while not specifically targeted at traveling, may be used to obtain news and breaking events, as well as location-specific information about POIs and local events in a traveler's destination. MetaCarta uses location-specific information collected from both traditional and on-line news media to provide "geographic intelligence solutions". By linking this to a geographical knowledge base and custom gazetteer, MetaCarta is able to provide in-depth context- and location-aware information, with options to personalize the information presented to an end user.

[50]We differentiate "explicit use of SW technology and services" by excluding web sites and services that simply provide, e.g., links to a Twitter account or feed, flickr tags and photos, or include a Facebook like button.

[51]http://www.tripadvisor.com.

[52]http://www.journeywoman.com.

[53]http://www.ubisense.net.

Tintarev et al. [30] demonstrate the added benefit in personalizing recommendations of popular POIs for tourists. Hornecker et al. [17], like [30], recognize the benefit in using personal information to guide the exploration of new areas. To reduce information load and to allow serendipitous discovery, however, [17] only alert the user to POIs nearby that match their preferences or that are similar to previously visited POIs. Sheth and Thomas [28] recognize the specific challenges in retrieving semantically enriched information in dynamically evolving situations, such as commonly occurs in social media.

One limitation in mobile device-based travel and tourist applications is that they are often tied to a specific location or type of institution (e.g., museums), often due to limited resources in small devices and the cost of network access [9, 13], especially when roaming outside a user's home location ([21] provides some examples).

## 11.7 Conclusion

This chapter has examined state of the art perspectives in the design and development of Travel Mashups. We have explored the varying travel information needs of the modern technology user, and mapped these to the social, abstract, physical, and temporal dimensions which they span in both the physical and the on-line worlds. By examining also how mobility in the digital society has re-characterized travel information needs we have identified where these changes have introduced new perspectives for the categorization of Travel Mashups. Based on this analysis we have provided further categorization based on the mobility features offered by *Travel Mashups*, and based also on the target group to which the mashup services are offered. We have introduced, in this chapter, a generic three-layer architecture for Travel Mashups, which takes into account an information sources layer, a generation layer and presentation layers. These layers cater for knowledge-based orchestration of services and the capability for exposing Travel Mashup data as Linked Data, in order to support structured capture and reuse of the collective knowledge that feeds into Travel Mashups.

This chapter provides a description of the *Topica Travel Mashup*, to illustrate how the design and architecture ideas presented are met in the state of the art *Travel Mashup*. *Topica* uses social stream data as an information source, to provide knowledge-based, business logic that meets the travel information needs of different end users. *Topica* associates the retrieval of POIs with topical information, based on the end user's expressed and implicit interests, which characterize POIs in the user's physical location.

The chapter concludes with a review of existing Travel Mashups and other closely related web- and mobile-based *apps*. As future perspectives of travel information needs include the demands of real-time location and user awareness, future Travel Mashups will need to cope well with real-time changes in user and location features as well as the dynamic integration of composite services able to adapt on the fly to these changes. We envisage that research in the field will continue to result in improvements to existing services, as well as the development of new SW-enabled services to meet the expectations of the modern, ubiquitous technology user.

## References

1. Abel F, Gao Q, Houben G, Tao K (2011) Semantic enrichment of twitter posts for user profile construction on the social web. In: Proceedings of the extended semantic web conference 2011
2. Adipat B, Zhang D (2005) Interface design for mobile applications. In: A conference on a human scale, 11th Americas conference on information systems (AMCIS 2005), p 494
3. Asur S, Huberman BA (2010) Predicting the future with social media. In: Proceedings of the international conference on web intelligence and intelligent agent technology, pp 492–499
4. Becker C, Bizer C (2009) Exploring the geospatial semantic web with DBpedia mobile. J Web Semant 7(4):278–286
5. Becker H, Iter D, Naaman M, Gravano L (2012) Identifying content for planned events across social media sites. In: Proceedings of the 5th ACM international conference on web search and data mining, WSDM'12. ACM, New York, pp 533–542
6. Bitzer S, Ramroth S, Schumann M (2009) Mashups as an architecture for knowledge management systems. In: Hawaii international conference on system sciences, pp 1–10
7. Blum A (1998) On-line algorithms in machine learning. In: Developments from a June 1996 seminar on online algorithms: the state of the art. Springer, London, pp 306–325
8. Braun M, Schmeiß D, Scherp A, Staab S (2010) Stevie—collaborative creation and exchange of events and pois on a mobile phone. In: Proceedings of the 2nd ACM international workshop on events in multimedia, EiMM'10, pp 35–40
9. Brown B, Chalmers M (2003) Tourism and mobile technology. In: Proceedings of the 8th European conference on computer supported cooperative work, ECSCW'03. Kluwer Academic, Norwell, pp 335–354
10. Cano A, Dadzie AS, Hartmann M (2011) Who is who—a linked data visualisation tool for mobile environments. In: The semantic web: research and applications. Lecture notes in computer science, vol 6644. Springer, Berlin, pp 451–455
11. Cano AE, Dadzie AS, Uren V, Ciravegna F (2012) Sensing presence (PreSense) ontology: user modelling in the semantic sensor web. In: Garciá-Castro R, Fensel D, Antoniou G (eds) The semantic web: ESWC 2011 workshops, revised selected papers. Springer, Berlin, pp 253–268
12. Compton M, Henson C, Neuhaus H, Lefort L, Sheth A (2009) A survey of the semantic specification of sensors. In: Proceedings of the 2nd international workshop on semantic sensor networks at ISWC 2009, SSN09, pp 17–32
13. Freyne J, Brennan A, Smyth B, Byrne D, Smeaton A, Jones G (2009) Automated murmurs: the social mobile tourist application. In: Proceedings of the international conference on computational science and engineering, CSE'09, pp 1021–1026
14. Hao Q, Cai R, Wang C, Xiao R, Yang JM, Pang Y, Zhang L (2010) Equip tourists with knowledge mined from travelogues. In: Proceedings of the 19th international conference on world wide web, WWW'10. ACM, New York, pp 401–410
15. Hogg T (2010) Inferring preference correlations from social networks. Electron Commer Res Appl (special issue: Social networks and Web 2.0) 9(1):29–37
16. Hong D, Chiu D, Shen V, Cheung S, Kafeza E (2007) Ubiquitous enterprise service adaptations based on contextual user behavior. Inf Syst Front 9:343–358
17. Hornecker E, Swindells S, Dunlop M (2011) A mobile guide for serendipitous exploration of cities. In: Proceedings of the 13th international conference on human computer interaction with mobile devices and services, MobileHCI'11. ACM, New York, pp 557–562
18. Jiang P (2009) Consumer adoption of mobile internet services: an exploratory study. J Promot Manag 15(3):418–454

19. Kabadayi S, Pridgen A, Julien C (2006) Virtual sensors: abstracting data from physical sensors. In: Proceedings of the international symposium on a world of wireless, mobile and multimedia networks, WoWMoM 2006, pp 587–592
20. Kamis A, Stern T, Ladik D (2010) A flow-based model of web site intentions when users customize products in business-to-consumer electronic commerce. Inf Syst Front 12:157–168
21. Kenteris M, Gavalas D, Economou D (2009) An innovative mobile electronic tourist guide application. Pers Ubiquitous Comput 13(2):103–118
22. Kulathuramaiyer N (2007) Mashups: emerging application development paradigm for a digital journal. Computer 13(4):531–542
23. Küpper A (2005) Location-based services: fundamentals and operation. Wiley, New York
24. Li C, Datta A, Sun A (2011) Semantic tag recommendation using concept model. In: Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval, SIGIR'11. ACM, New York, pp 1159–1160
25. Recordon D, Reed D (2006) OpenID 2.0: a platform for user-centric identity management. In: Proceedings of the 2nd ACM workshop on digital identity management, DIM'06. ACM, New York, pp 11–16
26. Rowe M (2010) The credibility of digital identity information on the social web: a user study. In: Proceedings of the 4th workshop on information credibility, WICOW'10, pp 35–42
27. Sheth A (2009) Citizen sensing, social signals, and enriching human experience. IEEE Internet Comput 13(4):87–92
28. Sheth A, Thomas C, Mehra P (2010) Continuous semantics to analyze real-time data. IEEE Internet Comput 14:84–89
29. Sheth AP, Henson CA, Sahoo SS (2008) Semantic sensor web. IEEE Internet Comput 12(4):78–83
30. Tintarev N, Flores A, Amatriain X (2010) Off the beaten track: a mobile field study exploring the long tail of tourist recommendations. In: Proceedings of the 12th international conference on human computer interaction with mobile devices and services, MobileHCI'10, pp 209–218
31. Yamada T, Kaneko M, Katou K (2005) A mobile communication simulation system for urban space with user behavior scenarios. In: Yang LT et al (eds) High performance computing and communications. Springer, Berlin, pp 979–990
32. Zeng D, Guo S, Cheng Z (2011) The web of things: a survey (invited paper). J Commun 6(6):424–438

# Glossary and Subject Index

**Brigitte Endres-Niggemeyer**

## Glossary and Subject Index Use

The glossary lists and explains technical terms used in the book. It works as a subject index, but it comes up with short descriptions of the concepts instead of merely listing them. This has its reasons in the behavior of mashups. As they spread out over many areas, they may draw readers into surroundings where they risk to be confronted with new concepts, so that sometimes some assistance may be welcome. Almost all glossary entries link to the website or another resource (most often a scientific paper) from where the explanation was taken and where more information is available. Often Wikipedia helps with some knowledge first aid. From the Wikipedia pages, links lead to more comprehensive sources and deeper knowledge. Linking straight to the original websites often turned out to be disappointing because they did not offer any suitable material for the short glossary items.

As usual, glossary terms appear in alphabetical order. Their entries end with a set of mnemonic keys that point to the chapters where the term occurs. The keys immediately tell how popular a term is and where it occurs. The keys and their references:

- **[dbpedia]** DBpedia Mashups
- **[eco]** The Mashup Ecosystem
- **[emergency]** Mashups for the Emergency Management Domain
- **[math]** Mashups Using Mathematical Knowledge
- **[search]** Mashups for Web Search Engines
- **[sensors]** Mashups for the Web of Things
- **[simil]** Similarity Mashups for Recommendation
- **[speech]** Speech Mashups
- **[standards]** Mashups Live on Standards
- **[travel]** Travel Mashups
- **[urban]** Urban Mashups

All web links were checked on November 30, 2012.

B. Endres-Niggemeyer (ed.), *Semantic Mashups*,
DOI 10.1007/978-3-642-36403-7, © Springer-Verlag Berlin Heidelberg 2013

# A

**A* (A-star) graph search:** A* is a computer algorithm for pathfinding and graph traversal. Noted for its performance and accuracy, it enjoys widespread use. http://en.wikipedia.org/wiki/A*_search_algorithm **[dbpedia]**

**Acoustic model:** Large sampling library of speech recordings, capturing all possible sound variations. **[speech]**

**Actuator:** Actuators are executing devices of an agent, as opposed to sensors that perceive external stimuli. **[eco] [sensors] [urban] [travel]**

**AIFF:** The Audio Interchange File Format (AIFF) is an audio file format standard used for storing sound data on electronic audio devices. The audio data in a standard AIFF file are uncompressed pulse-code modulation (PCM). http://en.wikipedia.org/wiki/Audio_Interchange_File_Format **[standards]**

**AJAX:** AJAX (Asynchronous JavaScript and XML) is a group of interrelated web development techniques used on the client-side to create asynchronous web applications. **[eco] [search] [sensors] [standards] [travel]**

**Alert Service (SAS):** Web interface for publishing and subscribing to sensor alerts. http://www.ogcnetwork.net/SAS **[sensors] [travel]**

**Amazon Elastic Compute Cloud (EC2):** Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. Developers pay only for what they use. http://aws.amazon.com/ec2/ **[dbpedia] [speech]**

**Ambient intelligence:** Ambient intelligence (AmI) is intelligence which has been integrated into the environment. AmI refers to electronic environments that are sensitive and responsive to the presence of people. http://en.wikipedia.org/wiki/Ambient_intelligence **[eco]**

**Analog-to-Digital (A/D) converter:** Device that converts the input continuous physical quantity to a digital number that represents the quantity's amplitude. http://en.wikipedia.org/wiki/Analog-to-digital_converter **[speech]**

**Annotation:** Additional information associated with a web resource, often containing metadata. **[dbpedia] [eco] [emergency] [math] [speech] [travel]**

**Anomalous State of Knowledge (ASK):** Situation of knowledge gap in information seeking—a user is unable to specify precisely what is needed to resolve the missing knowledge anomaly. **[search]**

**ANSI:** American National Standards Institute. http://www.ansi.org **[standards]**

**Application programming interface (API):** Specification intended to be used as an interface by software components to communicate with each other. http://en.wikipedia.org/wiki/Application_programming_interface **[dbpedia] [eco] [emergency] [math] [search] [sensors] [simil] [speech] [standards] [urban] [travel]**

**ARML:** XML-based Augmented Reality Markup Language. http://www.opengeospatial.org/projects/groups/arml2.0swg **[standards]**

**Atom:** XML-based Syndication Format for web feeds. http://tools.ietf.org/html/rfc4287 **[simil] [standards]**

**Audio capturing:** Obtaining an external sound signal for a sound file, normally through a microphone. http://docs.oracle.com/javase/tutorial/sound/capturing.html **[speech]**

**Audio play-back:** Playing audio from a sound file. **[speech]**

**Augmented reality (AR):** Live view of a physical, real-world environment augmented by computer-generated sensory input such as sound, video, graphics or GPS data. http://en.wikipedia.org/wiki/Augmented_reality **[eco] [standards] [urban]**

**Authentication:** Identifying a computer user, in common practice at least with ID and password. Different from authorization permitting the access to specific resources. **[math] [search] [speech] [standards] [travel]**

**Automatic speech recognition (ASR):** Conversion of speech to text. **[speech]**

**Autosuggest functionality:** Autosuggest, or autocomplete, shows a list of values in a drop down list that is filtered by the user input in a text field. http://www.oracle.com/technetwork/developer-tools/adf/learnmore/004-auto-suggest-169120.pdf **[search]**

**AVS:** Chinese Audio Video Standard. **[standards]**

## B

**Bayesian classifier:** A Bayesian classifier tries to predict the values of features for members of a class from a (natural) class they belong to. Examples are grouped in classes because they have common values for the features. If an agent knows the class, it can predict the values of the other features. If it does not know the class, Bayes' rule can be used to predict the class given (some of) the feature values. The simplest case is the naive Bayesian classifier, which makes the independence assumption that the input features are conditionally independent of each other given the classification. http://artint.info/html/ArtInt_181.html **[simil]**

**Bayesian networks:** Bayesian networks are directed acyclic graphs whose nodes represent random variables. Edges represent conditional dependencies stated in probability functions. http://en.wikipedia.org/wiki/Bayesian_network **[sensors]**

**BDGP:** Berkeley Drosophila Genome Project. **[eco]**

**BNF:** Backus–Naur Form—formal notation for languages. http://otal.umd.edu/drweb/c++tutorial/lessons/BNF.HTM **[speech]**

## C

**C-SPARQL:** SPARQL extension for continuous queries. http://wiki.larkc.eu/c-sparql **[urban]**

**Calibration:** Calibration sets a measurement device to a standard given by a certified master device. **[sensors]**

**CAS:** A computer algebra system (CAS) is a typically interactive software system for computer algebra. Most CAS also include calculus and many other areas of mathematics. http://en.wikipedia.org/wiki/Computer_algebra_system **[math]**

**Cascading Style Sheets (CSS):** Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document, including plain XML, SVG and XUL. http://en.wikipedia.org/wiki/Cascading_Style_Sheets **[standards]**

**CDC:** Center for Disease Control and Prevention. http://www.cdc.gov **[emergency]**

**CDMI:** Cloud Management Interface: Functional interface that applications will use to create, retrieve, update and delete data elements from the Cloud. http://www.snia.org/cdmi **[standards]**

**Chunking:** Chunking refers to the data transfer encoding introduced in HTTP 1.1 to deliver data in 'chunks'. This encoding allows HTTP messages to be delivered without specifying the total length of the message which is typically required by the entity header field

Content-length. In chunking mode, the sender specifies the length of each chunk just before the chunk itself and communicates the end of the message by sending a final chunk of length zero to the receiver. http://en.wikipedia.org/wiki/Chunked_transfer_encoding **[speech]**

**Citizen sensor network:** Social Web as a network of interconnected, participatory citizens who actively observe, report, collect, analyze and disseminate information about events and activities via text, audio and/or video messages in almost real time. **[emergency] [travel]**

**Closed world assumption:** The closed world assumption is the presumption that what is not currently known to be true is false. The opposite of the closed world assumption is the open world assumption, stating that lack of knowledge does not imply falsity. http://en.wikipedia.org/wiki/Closed_world_assumption **[urban]**

**Clustering:** Cluster analysis or clustering is assigning a set of objects into groups (called clusters) so that the objects in the same cluster are more similar to each other than to those in other clusters. http://en.wikipedia.org/wiki/Cluster_analysis **[emergency] [search] [sensors]**

**CMYK:** Color space for print. CMYK refers to the four inks used in some color printing: cyan, magenta, yellow, and key (black). http://en.wikipedia.org/wiki/CMYK_color_model **[standards]**

**Codec:** A codec is a device or computer program capable of encoding or decoding a digital data stream or signal. In practice, "codec" is sometimes used to refer to coding or compression formats. http://en.wikipedia.org/wiki/Codec **[speech]**

**Collaborative filtering (CF):** Collaborative filtering (CF) is a technique used by some recommender systems. They filter for information or patterns with techniques involving collaboration among multiple agents, viewpoints, data sources, etc. This may include automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). http://en.wikipedia.org/wiki/Collaborative_filtering **[simil] [urban]**

**Collaborative software:** Collaborative software or groupware is computer software designed to help people involved in a common task to achieve goals. They normally fulfill different roles, e.g. as initiators, stakeholders and administrators. Travel organizing is a scenario where different players collaborate. http://en.wikipedia.org/wiki/Collaborative_software, http://mashart.org/composableweb2009/paper6.pdf **[eco] [travel] [urban]**

**Comma-Separated Values (CSV):** A Comma-Separated Values (CSV) file stores tabular data (numbers and text) in plain-text form. A CSV file consists of any number of records, separated by line breaks of some kind; each record consists of fields, separated by some other character or string, most commonly a literal comma or tab. Usually, all records have an identical sequence of fields. http://en.wikipedia.org/wiki/Comma-separated_values **[eco] [sensors] [standards] [urban]**

**Complex event processing (CEP):** Event processing that combines data from multiple sources to infer events or patterns that suggest more complicated circumstances. The goal is to identify meaningful events (such as opportunities or threats) and respond to them as quickly as possible. http://en.wikipedia.org/wiki/Complex_event_processing **[sensors]**

**Compression:** Data compression, source coding, or bit-rate reduction involves encoding information using fewer bits than the original representation. Lossless compression reduces bits by identifying and eliminating statistical redundancy. Lossy compression reduces bits by identifying marginally important information and removing it. http://en.wikipedia.org/wiki/Data_compression **[speech] [standards]**

**Concatenative speech synthesis:** Concatenative speech synthesis is based on the concatenation (or stringing together) of segments of recorded speech from a big speech database. http://en.wikipedia.org/wiki/Speech_synthesis **[speech]**

**Consumer mashup:** Consumer mashups are made for/created by end-users. They combine data from multiple public sources in the browser and organize them through a simple browser user interface. They are seen as opposed to data mashups and business mashups. http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid) **[eco] [travel]**

**Content Dictionary (CD):** A Content Dictionary (CD) is the declaration of a collection of symbols, their names, descriptions, and rules. An example of a CD is 'setname1' which indicates the semantic of such symbols as N, the set of natural numbers. http://www.openmath.org/cd/index.html **[math]**

**Context awareness:** Behavior of applications that adapt to context conditions during job execution. **[eco]**

**Corpus:** A corpus is a set of empirical data, often text or speech. The standard use is statistical analysis and hypothesis testing, checking occurrences or validating linguistic rules on a specific universe. http://en.wikipedia.org/wiki/Text_corpus **[dbpedia] [search] [speech]**

**Cosine Similarity measure:** Cosine Similarity is often used when comparing two documents against each other. It measures the angle between the two vectors. If the value is zero the angle between the two vectors is 90 degrees and they share no terms. If the value is 1 the two vectors are the same except for magnitude. http://inside.mines.edu/~ckarlsso/mining_portfolio/similarity.html#cosine **[simil]**

**Crawler:** A web crawler is a computer program that browses the World Wide Web in a methodical, automated manner. http://en.wikipedia.org/wiki/Web_crawler **[dbpedia] [simil]**

**Creative Commons:** Creative Commons is a non-profit organization that enables the sharing and use of creativity and knowledge through free legal tools, the Creative Commons licenses. http://creativecommons.org/about **[dbpedia]**

**Crowd sourcing:** Crowd sourcing is a distributed problem-solving and production model. Problems are broadcast to an unknown group of solvers in the form of an open call for solutions. Users submit solutions. The contributor of the solution is, in some cases, compensated either monetarily, with prizes, with recognition or merely with his own intellectual satisfaction. http://en.wikipedia.org/wiki/Crowdsourcing **[dbpedia] [emergency] [search]**

**CURIO:** Collaborative User Resource Interaction Ontology. http://www.weknowit.eu/content/curio_collaborative_user_resource_interaction_ontology **[travel]**

# D

**Data streaming:** Data streaming is about monitoring huge and rapidly changing streams of data. Relevant applications include analyzing network traffic, online auctions, transaction logs, telephone call records, automated bank machine operations, and atmospheric and astronomical events. The data streaming model differs from computation over traditional stored datasets since algorithms must process their input by making one or a small number of passes over it, using only a limited amount of working memory. http://twiki.di.uniroma1.it/pub/Ing_algo/WebHome/DFchapter08.pdf **[sensors] [speech]**

**Datalog:** Datalog is a declarative logic programming language that syntactically is a subset of Prolog. It is often used as a query language for deductive databases. In recent years, Datalog has found new application in data integration, information extraction, networking, program analysis, security, and cloud computing. http://en.wikipedia.org/wiki/Datalog **[sensors]**

**DBMS:** Database management system. **[sensors] [urban]**

**DBpedia Ontology:** The DBpedia Ontology is a shallow, cross-domain ontology, which has been manually created based on the most commonly used infoboxes within Wikipedia. The ontology currently covers 359 classes which form a subsumption hierarchy and are described by 1775 different properties. http://wiki.dbpedia.org/Ontology?v=194q **[dbpedia]**

**DBpedia:** DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and to make this information available on the Web. http://dbpedia.org/About **[dbpedia] [eco] [emergency] [math] [search] [simil] [standards] [travel] [urban]**

**DCCI:** Delivery Context Client Interfaces (DCCI), Nokia-specific sensor specification. **[eco]**

**Decision tree:** A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal. Another use of decision trees is as a descriptive means for calculating conditional probabilities. http://en.wikipedia.org/wiki/Decision_tree **[simil]**

**Demultiplexer:** Multiplexing map data from parallel incoming lines to one channel where they are transported in sequence. Demultiplexing separates the data again and distributes them again on parallel lines. http://www.wisc-online.com/objects/ViewObject.aspx?ID=DIG5704 **[sensors]**

**Description logics (DLs):** Description logics (DLs) is a family of knowledge representation (KR) formalisms that represent the knowledge of an application domain by first defining the relevant concepts of the domain (its terminology), and then using these concepts to specify properties of objects and individuals occurring in the domain (the world description). http://www.inf.unibz.it/~franconi/dl/course/dlhb/dlhb-02.pdf **[dbpedia] [standards]**

**Digital signal processing:** Digital signal processing (DSP) is the mathematical manipulation of an information signal to modify or improve it. It is characterized by the representation of discrete time, discrete frequency, or other discrete domain signals. DSP includes subfields like: audio and speech signal processing, sonar and radar signal processing, sensor array processing, etc. http://en.wikipedia.org/wiki/Digital_signal_processing **[speech]**

**Digital Video Broadcasting (DVB):** Digital Video Broadcasting (DVB) is a suite of internationally accepted open standards for digital television. http://en.wikipedia.org/wiki/Digital_Video_Broadcasting **[standards]**

**Digital-to-analog (D/A) converter:** A digital-to-analog converter converts a digital (usually binary) code to an analog signal. A common use of digital-to-analog converters is generation of audio signals from digital information in music players. http://en.wikipedia.org/wiki/Digital-to-analog_converter **[speech]**

**Dijkstra algorithm:** Dijkstra's algorithm is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms. http://en.wikipedia.org/wiki/Dijkstra's_algorithm **[urban]**

**DIN:** Deutsches Institut für Normung (German Institute for Standardization). **[standards]**

**Disambiguation:** Word-sense disambiguation is the process of identifying the sense of a word in a sentence. **[eco] [search]**

**Discrete Fourier transformation (DFT):** The discrete Fourier transform (DFT) decomposes a discrete time-bound periodic signal to its frequency spectrum, so that the frequencies can be treated separately. Inputs are often created by sampling a continuous function, such as the amplitude of a person's voice over time. The Fast Fourier transform (FFT) algorithm is used most frequently. http://en.wikipedia.org/wiki/Discrete_Fourier_transform **[sensors]**

**Document Object Model (DOM):** The Document Object Model (DOM) is a cross-platform and language-independent convention for representing of and interacting with objects in HTML and XML documents. http://en.wikipedia.org/wiki/Document_Object_Model **[simil] [standards]**

**Document Type Definition (DTD):** A Document Type Definition (DTD) provides a grammar for an XML document. It defines the document structure with a list of legal elements and attributes. http://www.w3.org/TR/REC-xml, http://www.w3schools.com/dtd/dtd_intro.asp **[standards]**

**Domain Name System (DNS):** The Domain Name System (DNS) is a hierarchical distributed naming system for computers, services, or any resource connected to the Internet or a private network. It resolves queries for these names into IP addresses for the purpose of locating computer services and devices worldwide. By providing a worldwide, distributed keyword-based redirection service, the Domain Name System is an essential component of the functionality of the Internet. http://en.wikipedia.org/wiki/Domain_Name_System **[standards]**

**Domain-specific description language (DSL):** A domain-specific language (DSL) is a programming language or specification language dedicated to a particular problem domain, a particular problem representation technique, and/or a particular solution technique. **[eco] [emergency]**

**Drosophila Melanogaster:** The fruit fly, a preferred genomics animal because of its fast generation sequence. **[eco]**

**DublinCore:** The Dublin Core metadata terms are a set of vocabulary terms which can be used to describe resources. The terms can be used to describe a full range of web resources: video, images, web pages etc. and physical resources such as books and objects like artworks. http://en.wikipedia.org/wiki/Dublin_Core **[standards]**

**DVB-S2:** Digital Video Broadcasting–Satellite–Second Generation (DVB-S2) is a digital television broadcast standard that has been designed as a successor for the popular DVB-S system. DVB-S2 is envisaged for broadcast services including standard and HDTV, interactive services including Internet access, and (professional) data content distribution. http://en.wikipedia.org/wiki/DVB-S2 **[standards]**

## E

**ECMA International:** ECMA is the Industrial standards organization for Information and Communications Technology (ICT) and Consumer Electronics (CE) standards. http://www.ecma-international.org **[standards]**

**EDGE:** Enhanced Data rates for GSM Evolution (EDGE) is a digital mobile phone technology that allows improved data transmission rates as a backward-compatible extension of GSM (Global System for Mobile Communications). http://en.wikipedia.org/wiki/Enhanced_Data_Rates_for_GSM_Evolution **[speech]**

**Electronic Product Code (EPC):** The Electronic Product Code (EPC) is a universal identifier that provides a unique identity for every physical object anywhere in the world, for all time. Its structure is defined in the EPCglobal Tag Data Standard, which is an open standard freely available for download from the website of EPCglobal. http://en.wikipedia.org/wiki/Electronic_Product_Code, http://www.gs1.org/epcglobal **[sensors] [standards]**

**Embedded system/device:** An embedded system is a computer system designed for specific control functions within a larger system (washing machine, smartphone etc.), often with real-time computing constraints. It consists of microcontrollers or digital signal processors (DSP), and possibly also hardware/mechanical parts. http://en.wikipedia.org/wiki/Embedded_system **[eco] [math] [sensors] [standards]**

**EMMA (Extensible Multi-Modal Annotations):** EMMA (Extensible Multi-Modal Annotations) is an XML-based data exchange format for the interface between input processors and interaction management systems. http://www.w3.org/TR/emma/ **[speech]**

**Encryption:** Encryption is the process of encoding messages (or information) in such a way that only authorized parties can read it. The information is encrypted using an encryption algorithm and decoded by the reviewer using a decryption algorithm. http://en.wikipedia.org/wiki/Encryption **[standards]**

**End-User Development (EUD):** End-User Development can be defined as a set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify or extend a software artifact. http://en.wikipedia.org/wiki/End-user_development **[eco]**

**Enforcement Management Model (EMM):** The Enforcement Management Model (EMM) is a logical system that helps inspectors to make enforcement decisions in response to breaches of health and safety legislation. http://www.hse.gov.uk/enforce/emm.pdf **[emergency]**

**Enterprise Mashup Markup Language (EMML):** Enterprise Mashup Markup Language (EMML) is an XML markup language for creating enterprise mashups. Mashed data produced by enterprise mashups are presented in graphical user interfaces as mashlets, widgets, or gadgets. EMML can also be considered a declarative mashup domain-specific language (DSL). http://en.wikipedia.org/wiki/EMML **[eco]**

**Environmental intelligence:** Insight into the environment via sensor data. **[eco] [sensors]**

**ESRI:** ESRI is the leading worldwide supplier of Geographic Information System (GIS) software and geodatabase management applications. http://www.esri.com **[eco] [urban]**

**Euclidean distance:** The Euclidean distance between points is the length of the line segment connecting them. http://en.wikipedia.org/wiki/Euclidean_distance **[eco] [simil]**

**Executable paper:** Paper with data and process environment that enables users to rerun the experiment. **[math]**

# F

**FIFO buffer:** First In, First Out buffer. **[sensors]**

**FIPS:** US Federal Information Processing Standard. http://www.nist.gov/itl/fips.cfm **[standards]**

**FlyBase:** Database documenting the genome of the fruit fly. http://flybase.org **[eco]**

**Folksonomy:** A folksonomy is a system of classification derived from the practice and method of collaboratively creating and managing tags to annotate and categorize content. This practice is also known as collaborative tagging, social classification, social indexing, and social tagging. Even without any central controlled vocabulary, consensus around stable distributions and shared vocabularies does emerge. http://en.wikipedia.org/wiki/Folksonomy **[dbpedia]**

**Form Interpretation Algorithm (FIA):** The Form Interpretation Algorithm (FIA) drives the interaction between the user and a VoiceXML form or menu. http://vxmlfaq.com/notes-05-Form-Interpretation-Algorithm.html **[speech]**

**Formulas:** A formula symbolically models functional relationships among entities. Current examples are chemical formulas and chemical or mathematical equations. Formula specifications exist, e.g. for financial reporting, but formulas may also be stated according to common practice. For financial reports formulae specs see http://www.xbrl.org/SpecRecommendations/ **[math]**

**Freebase:** Freebase is an open, Creative Commons licensed repository of structured data of almost 23 million entities. An entity is a single person, place, or thing. Freebase connects entities together as a graph. http://wiki.freebase.com/wiki/What_is_Freebase%3F **[dbpedia] [eco] [search] [sensors]**

**Friend of a Friend (FOAF):** FOAF (Friend of a Friend) is an ontology describing persons, their activities, and their relations to other people and objects. Anyone can use FOAF to describe him or herself. FOAF allows groups of people to describe social networks without the need for a centralized database. The FOAF vocabulary is expressed using RDF and OWL. http://en.wikipedia.org/wiki/FOAF_(software), http://www.foaf-project.org/about **[dbpedia] [simil]**

# G

**Game With a Purpose (GWAP):** A Game with a Purpose (GWAP) is a human-based computation technique in which a computational process performs its function by outsourcing certain steps to humans in an entertaining way. http://en.wikipedia.org/wiki/Human-based_computation_game, http://www.cs.cmu.edu/~biglou/ieee-gwap.pdf **[urban]**

**General Packet Radio Service (GPRS):** The General Packet Radio Service (GPRS) is a packet oriented mobile data service on the 2G and 3G cellular communication system's global system for mobile communications (GSM). GPRS was originally standardized by the European Telecommunications Standards Institute (ETSI). It is now maintained by the 3rd Generation Partnership Project (3GPP). http://en.wikipedia.org/wiki/General_Packet_Radio_Service **[speech] [travel]**

**Genetic algorithm:** A genetic algorithm is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. During procreation of the next generation, mutations occur. This motivates the hope that the new population will be better than the old one. Solutions (chromosomes of the offspring) are selected according to their fitness—the more suitable they are, the more chances they have to reproduce. This is repeated until some condition (for example a number of populations or improvement of the best solution) is satisfied. http://www.obitko.com/tutorials/genetic-algorithms/ga-basic-description.php **[simil]**

**Geocoding:** Geocoding is the process of finding associated geographic coordinates (often latitude and longitude) from other geographic data, such as street addresses, or ZIP codes. With geographic coordinates the features can be mapped and entered into Geographic Information Systems, or the coordinates can be embedded into media such as digital photographs via geotagging. http://en.wikipedia.org/wiki/Geocoding **[eco] [emergency] [sensors] [urban]**

**Geographic Information System (GIS):** A Geographic Information System (GIS) is a system designed to capture, store, manipulate, analyze, manage, and present all types of geographical data. GIS applications are tools that allow users to create interactive queries (user-created searches), analyze spatial information, edit data in maps, and present the results of all these operations. http://en.wikipedia.org/wiki/Geographic_information_system **[eco] [urban]**

**Geography Markup Language (GML):** The Geography Markup Language (GML) is the XML grammar defined by the Open Geospatial Consortium (OGC) to express geographical features. GML serves as a modeling language for geographic systems as well as an open interchange format for geographic transactions on the Internet. http://en.wikipedia.org/wiki/Geography_Markup_Language **[standards]**

**GeoNames:** The GeoNames geographical database covers all countries and contains over eight million place names that are available for download free of charge. http://en.wikipedia.org/wiki/Geography_Markup_Language, http://www.geonames.org **[dbpedia] [eco] [emergency] [search] [sensors]**

**GeoWGS84 Vocabulary:** The Basic Geo Vocabulary is a simple RDF Schema vocabulary for representing latitude, longitude, and altitude information in the WGS84 geodetic reference datum. http://semanticweb.org/wiki/Basic_Geo_Vocabulary **[sensors] [urban]**

**Global Positioning System (GPS):** The Global Positioning System (GPS) is a space-based satellite navigation system that provides location and time information in all weather, anywhere on or near the Earth, where there is an unobstructed line of sight to four or more GPS satellites. It is maintained by the United States government and is freely accessible to anyone with a GPS receiver. http://en.wikipedia.org/wiki/Global_Positioning_System **[eco]**

**Google Insights:** Google Insights has been merged into Google Trends. Google Trends is a public web facility based on Google Search, which shows how often a particular search-term is entered relative to the total search-volume across various regions of the world, and in various languages. http://www.google.com/trends/ **[emergency]**

**Granularity:** Granularity is the extent to which a structure is broken down into small parts. It is the extent to which a larger entity is subdivided. Coarse-grained units consist of fewer, larger components than fine-grained ones. Granularity levels may be nested. In natural language, one observes many levels of granularity, spanning e.g. from single phones to sentences and discourses. http://en.wikipedia.org/wiki/Granularity **[sensors] [speech] [travel] [urban]**

**Graph:** A graph is a representation of a set of objects where some pairs of the objects are connected by links. http://en.wikipedia.org/wiki/Graph_(mathematics) **[emergency]**

# H

**HTML:** HyperText Markup Language (HTML) is the main markup language for displaying web pages and other information that can be displayed in a web browser. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. http://en.wikipedia.org/wiki/HTML **[dbpedia] [eco] [math] [search] [sensors] [simil] [speech] [standards]**

**HTML5:** HTML5 (a heavily re-engineered version of earlier HTML) is a markup language for structuring and presenting content for the web and a core technology of the Internet. HTML5 adds many new features. These include the new <video>, <audio> and <canvas> elements, as well as the integration of scalable vector graphics (SVG) content and MathML for mathematical formulas. http://en.wikipedia.org/wiki/HTML5 **[math] [simil] [speech] [standards] [travel]**

**HTTP:** The HyperText Transfer Protocol (HTTP) is an application protocol for distributed, collaborative hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. HTTP/1.1 is the HTTP version in current common use. http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol **[eco] [math] [search] [simil] [speech] [standards]**

**Human computation:** Human computation centers around harnessing human intelligence to solve computational problems that are beyond the scope of existing Artificial Intelligence (AI) algorithms. For instance, Games with a purpose (e.g., the ESP Game) make online gamers generate useful data (e.g., image tags) while playing an enjoyable game. http://www.youtube.com/watch?v=tx082gDwGcM **[urban]**

**HyperText Query Language (HTQL):** HyperText Query Language (HTQL) is a language for the querying and transformation of HTML, XML and plain text documents. HTQL is developed in C++ with fast and efficient data extraction algorithms. HTQL provides COM and Python interfaces for use in JavaScript, Visual Basic, .NET, ASP, and Python applications. http://htql.net **[simil]**

**ID3 algorithm:** ID3 (Iterative Dichotomizer 3) is an algorithm used with decision trees in machine learning. We are given a set of records. Each record has the same structure, consisting of a number of attribute/value pairs. One of these attributes represents the category of the record. The problem is to determine a decision tree that on the basis of answers to questions about the non-category attributes correctly predicts the value of the category attribute. Usually the category attribute takes only the values true, false, or success, failure, or something equivalent. In any case, one of its values will mean failure. Implementations in several languages are available. http://www.cis.temple.edu/~ingargio/cis587/readings/id3-c45.html **[simil]**

**ID3 metadata:** ID3 is a metadata container most often used in conjunction with the MP3 audio file format. It allows information such as the title, artist, album, track number, and other information about the file to be stored in the file itself. http://en.wikipedia.org/wiki/ID3 **[standards]**

**IEEE 802.15.4:** The IEEE 802.15.4 standard specifies the physical layer and media access control for low-rate wireless personal area networks. http://en.wikipedia.org/wiki/IEEE_802.15.4 **[sensors]**

**IEEE:** Institute of Electrical and Electronics Engineers. http://www.ieee.org/index.html **[standards]**

**IMDb:** IMDb is the world's most popular and authoritative source for movie, TV and celebrity content. http://www.imdb.com **[dbpedia] [simil]**

**In-situ mashup:** Mashup of a real-world element enhanced with augmented reality information. **[eco]**

**Infobox:** The infobox is a template for structured data that appears on the upper right side of the Wikipedia website. **[dbpedia] [search]**

**Information seeking:** Information seeking is attempting to obtain information in both human and technological contexts. http://en.wikipedia.org/wiki/Information_seeking **[search]**

**Instant messaging:** Instant messaging (IM) is a form of communication over the Internet that offers quick transmission of text-based messages from sender to receiver. In push mode instant messaging basically offers real-time direct written language-based online chat. More advanced instant messaging allows enhanced modes of communication, such as live voice or video calling, video chat, and inclusion of hyperlinks to media. http://en.wikipedia.org/wiki/Instant_messaging **[standards]**

**Interface metaphor:** A GUI (graphical user interface) or other interface metaphor is a set of user interface visuals, actions and procedures that exploit specific knowledge that users already have of other domains. An example is the desktop metaphor with sheets, files, and folders. http://en.wikipedia.org/wiki/Interface_metaphor **[search] [standards]**

**Internet Engineering Task Force (IETF):** The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. http://www.ietf.org/about/ **[standards]**

**Interoperability:** Interoperability is the ability of two or more systems or components to exchange information and to use the information that has been exchanged (from IEEE glossary). http://en.wikipedia.org/wiki/Interoperability **[dbpedia] [eco] [sensors] [standards]**

**Intonation:** Intonation is the variation of pitch (frequency rising or falling pitch) while speaking. Intonation, rhythm, and stress are the three main elements of linguistic prosody. http://en.wikipedia.org/wiki/Intonation_(linguistics) **[speech]**

**Inverse Candidate Frequency weight (ICF):** The Inverse Candidate Frequency (ICF) weight renders the discriminative power of a word. ICF is inversely proportional to the number of resources it is associated with, as opposed to the Inverse Document Frequency (IDF) weight that represents the general importance of the word in the collection. http://www.wiwiss.fu-berlin.de/en/institute/pwo/bizer/research/publications/Mendes-Jakob-GarciaSilva-Bizer-DBpediaSpotlight-ISEM2011.pdf **[dbpedia]**

**IPv6:** IPv6 (Internet Protocol version 6) is the latest revision of the Internet Protocol (IP), the primary communications protocol upon which the entire Internet is built. It is intended to replace the older IPv4, which is still employed for the vast majority of Internet traffic as of 2012 but which runs out of addresses. IPv6 uses 128-bit addresses, whereas IPv4 uses 32-bit addresses, so that IPv6 widely enlarges the address space. IPv6 lowpan is IPv6 over Low power Wireless Personal Area Network. http://en.wikipedia.org/wiki/IPv6 **[sensors] [eco]**

**ISO:** International Organization for Standards. http://www.iso.org/iso **[standards]**

**Item-based k-nearest neighbor:** The k-nearest neighbor algorithm (k-NN) is a method for classifying objects based on closest training examples in the feature space. http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm **[urban]**

**ITU:** International Telecommunication Union. http://www.itu.int/en/Pages/default.aspx **[standards]**

# J

**Java Server Pages (JSP):** Java Server Pages (JSP) technology provides a simplified, fast way to generate dynamic web content. http://en.wikipedia.org/wiki/JavaServer_Pages **[standards]**

**JavaScript:** JavaScript is a scripting language commonly implemented as part of a web browser in order to create enhanced user interfaces and dynamic websites. http://en.wikipedia.org/wiki/JavaScript **[eco] [math] [simil] [speech] [standards] [travel]**

**JDOM:** JDOM provides a complete, Java-based solution for accessing, manipulating, and outputting XML data from Java code. http://www.jdom.org **[standards]**

**JSON:** JSON (the JavaScript Object Notation) is a text-based open standard designed for human-readable data interchange. In JavaScript it is used for representing simple data structures and associative arrays, called objects. Despite its relationship to JavaScript, it is language-independent, with parsers available for many languages. http://en.wikipedia.org/wiki/JSON **[dbpedia] [eco] [math] [simil] [speech] [standards]**

# K

**K-means:** k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. http://en.wikipedia.org/wiki/K-means_clustering, http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html **[sensors]**

**Keyhole Markup Language (KML):** The Keyhole Markup Language (KML) is an XML language for expressing geographic annotation and visualization within Internet-based, two-dimensional maps and three-dimensional Earth browsers. http://en.wikipedia.org/wiki/Keyhole_Markup_Language **[standards]**

**Kohana framework:** The Kohana framework is a PHP5 framework for building web applications. http://kohanaframework.org **[emergency]**

# L

**Language model:** A statistical language model assigns a probability to a sequence of words by means of a probability distribution. Language modeling is used in many natural language processing applications such as speech recognition, machine translation, part-of-speech tagging, parsing and information retrieval. http://en.wikipedia.org/wiki/Language_model **[speech]**

**LaTeX:** LaTeX is a document markup language and document preparation system for the TeX typesetting program. LaTeX is widely used in academia, in particular in technical and scientific domains. http://en.wikipedia.org/wiki/LaTeX **[math]**

**Learning Object Metadata (LOM):** Learning Object Metadata (LOM) is a data model, usually encoded in XML, describing a learning object and similar digital resources used to support learning and to make learning objects reusable, to aid their discovery, and to facilitate their interoperability, usually in the context of online learning management systems (LMS). http://en.wikipedia.org/wiki/Learning_object_metadata **[math]**

**Linear classifier:** A linear classifier decides on an object's characteristics to which class (or group) the object belongs to. It achieves this by making a classification decision based on the value of a linear combination of the characteristics. An object's characteristics are also known as feature values and are typically presented to the machine in a vector called a feature vector. http://en.wikipedia.org/wiki/Linear_classifier **[simil]**

**Linked Data Semantic Distance (LDSD):** The LDSD algorithm measures the distance of two objects on their direct and indirect linking in the linked data cloud. http://swl.slis.indiana.edu/files/ldrec.pdf **[dbpedia] [simil]**

**Linked data:** Linked data describes a method of publishing structured data so that it can be interlinked and become more useful. It builds upon standard Web technologies such as HTTP and URIs, and it extends them to share information in a way that can be read automatically by computers. The normal encoding format is RDF. Data from different sources can be connected and queried. http://en.wikipedia.org/wiki/Linked_data **[dbpedia] [eco] [emergency] [math] [search] [sensors] [simil] [standards] [travel] [urban]**

**Linked Open Data (LOD):** Linked Open Data (LOD) extends the Web with a data commons by publishing various open datasets as RDF and by setting RDF links between data items from different data sources. By September 2011 the datasets had grown to 31 billion RDF triples, interlinked by around 504 million RDF links. There is also an interactive visualization of the linked datasets to browse through the cloud. http://en.wikipedia.org/wiki/Linked_data **[dbpedia] [eco] [emergency] [search] [sensors] [standards] [urban]**

**Linked Sensor Middleware (LSM):** The Linked Sensor Middleware (LSM) is a platform that brings together the live real world sensed data and the Semantic Web. An LSM deployment provides many functionalities such as: wrappers for real-time data collection and publishing, a web interface for data annotation and visualization, and a SPARQL endpoint for querying unified Linked Stream Data and Linked Data. http://code.google.com/p/deri-lsm/ **[emergency]**

**LinkedGeoData:** LinkedGeoData is an effort to add a spatial dimension to the Web of Data/Semantic Web. It uses the information collected by the OpenStreetMap project and makes it available as an RDF knowledge base according to the Linked Data principles. It interlinks these data with other knowledge bases in the Linking Open Data initiative. http://linkedgeodata.org/About **[emergency] [urban]**

**Load shedding:** Load shedding is the deliberate switching off of electrical supply to parts of the electricity network, and hence to the customers in those areas. This practice is a core part of the emergency management of all electricity networks. http://www.thefreedictionary.com/load-shedding **[sensors]**

**Location awareness:** Location awareness refers to devices that can passively or actively determine their location. Navigational instruments provide location coordinates, surveying equipment identifies location with respect to a well-known locational wireless communications device. Network location awareness (NLA) describes the location of a node in a network. http://en.wikipedia.org/wiki/Location_awareness **[travel]**

**Location-based services (LBS):** Location-based services (LBS), mostly on mobile devices, include specific controls for location and time data as control features in computer programs. LBS are used in a variety of contexts, such as health, indoor object search, entertainment, work, personal life, etc. http://en.wikipedia.org/wiki/Location-based_service **[urban] [travel]**

**Lovins algorithm:** The Lovins algorithm is a stemming algorithm. It is noticeably bigger and faster than the Porter algorithm, because of its very extensive endings list. http://snowball.tartarus.org/algorithms/lovins/stemmer.html, http://www.cs.waikato.ac.nz/~eibe/stemmers/index.html **[simil]**

## M

**Machine learning:** A machine learns whenever it changes its structure, program, or data (based on its inputs or in response to external information) in such a manner that its expected future performance improves. For example, when the performance of a speech-recognition machine improves after hearing several samples of a person's speech, we can say that the machine has learned. http://robotics.stanford.edu/-nilsson/MLBOOK.pdf **[dbpedia] [emergency] [simil] [sensors] [speech] [travel] [urban]**

**MapQuest:** Free digital map server. http://www.mapquest.com **[eco]**

**Mathematical Knowledge Management (MKM):** Mathematical Knowledge Management aims at efficient, new techniques—based on sophisticated formal mathematics and software technology—to take advantage of the enormous amount of knowledge available in current mathematical sources and to organize mathematical knowledge in new ways, testing innovative theoretical and technological solutions for content-based systems, interoperability, management of machine understandable information, and the Semantic Web. http://www.mkm-ig.org/index.html **[math]**

**Mathematics Subject Classification (MSC):** The main purpose of the Mathematics Subject Classification (MSC) scheme is to help users find the items of present or potential interest to them as readily as possible—in products derived from the Mathematical Reviews Database (MRDB), in Zentralblatt MATH, or anywhere else where this classification scheme is used. http://www.ams.org/mathscinet/msc/msc2010.html **[math]**

**MathJax:** MathJax is an open source JavaScript display engine for mathematics that works in all modern browsers. http://www.mathjax.org **[math]**

**MathML:** MathML is a low-level specification for describing mathematics as a basis for machine to machine communication which provides a foundation for the inclusion of mathematical expressions in Web pages. http://www.w3.org/Math/ **[math]**

**MCDL:** Mashup Component Description Language. Several approaches target MCDL: Gagne et al. (2006), Aghaee and Pautasso (2011) and http://www.mmt.inf.tu-dresden.de/Forschung/Projekte/CRUISe/mcdl/ **[eco]**

**Media mashup:** A media mashup integrates (multi)media such as audio, video and augmented reality. **[eco]**

**MetaObject Facility (MOF):** The MetaObject Facility (MOF) is an Object Management Group (OMG) standard for model-driven engineering with use of the Unified Modeling Language (UML). http://en.wikipedia.org/wiki/Meta-Object_Facility **[eco]**

**Microformat:** Microformats are a set of simple open data formats built upon existing and widely adopted standards such as XML and XHTML. Examples are hCard and hCalendar. http://microformats.org/about **[eco] [simil] [standards]**

**Micropost:** A micropost is a brief post (or series of posts) to a personal blog on a microblogging web site such as Twitter. People can read microposts online or request that updates are delivered in real time to their desktop via instant messaging (IM) or sent to a mobile device via SMS text message. http://www.netlingo.com/word/micropost.php **[urban]**

**Middleware:** Middleware is computer software that provides services to software applications beyond those available from the operating system. http://en.wikipedia.org/wiki/Middleware **[emergency] [search]**

**MIDI (Musical Instrument Digital Interface):** MIDI (Musical Instrument Digital Interface) is an electronic musical instrument industry specification that allows electronic musical instruments, performance controllers, computers and related devices to communicate, as well as a hardware standard that guarantees compatibility between them. http://en.wikipedia.org/wiki/MIDI **[speech] [standards]**

**MIME:** Multipurpose Internet Mail Extensions (MIME) is an Internet standard for supporting text in character sets other than ASCII, non-text attachments, message bodies with multiple parts, and header information in non-ASCII character sets. http://en.wikipedia.org/wiki/MIME **[standards]**

**MP3:** MP3 (integrating MPEG-1 or MPEG-2 Audio Layer III) is a patented encoding format for digital audio which uses a form of lossy data compression. It is a common audio format for consumer audio storage, as well as a de facto standard of digital audio compression for the transfer and play-back of music on most digital audio players. http://en.wikipedia.org/wiki/MP3 **[standards]**

**MP4:** MP4 (also called MPEG-4 Part 14, formally ISO/IEC 14496-14:2003) is a multimedia container format standard that is most commonly used to store digital video and digital audio streams, especially those defined by MPEG, but can also be used to store other data such as subtitles and still images. MP4 allows streaming over the Internet. A separate hint track includes streaming information in the file. http://en.wikipedia.org/wiki/MPEG-4_Part_14 **[standards]**

**MPEG-1:** MPEG-1 is a standard for lossy compression of video and audio. MPEG-1 is used in a large number of products and technologies. http://en.wikipedia.org/wiki/MPEG-1 **[standards]**

**MPEG-2:** MPEG-2 is a standard for "the generic coding of moving pictures and associated audio information". It describes a combination of lossy video compression and lossy audio data compression methods which permit storage and transmission of movies using currently available storage media and transmission bandwidth. MPEG-2 is widely used for digital television signals. http://en.wikipedia.org/wiki/MPEG-2 **[standards]**

**MPEG-4:** MPEG-4 is a method of defining compression of audio and visual (AV) digital data. MPEG-4 is still a developing standard and is divided into a number of parts. It absorbs many of the features of MPEG-1 and MPEG-2 and other related standards, adding new features such as (extended) VRML support for 3D rendering, object-oriented composite files (including audio, video and VRML objects), support for externally specified Digital Rights Management and various types of interactivity. http://en.wikipedia.org/wiki/MPEG-4 **[standards]**

**MPEG:** The Moving Picture Experts Group (MPEG) is a working group of ISO/IEC in charge of the development of standards for coded representation of digital audio and video and related data. http://mpeg.chiariglione.org **[standards]**

**MPEGLA:** MPEGLA is a packager of patent pools for standards and other technology platforms used in consumer electronics, as well as chemical, eCommerce, education, energy, environment, healthcare and biotechnology, manufacturing and materials, transportation, and wireless technology. http://www.mpegla.com/main/Pages/About.aspx **[standards]**

**MRCPv2:** The Media Resource Control Protocol (MRCP) is a communication protocol used by speech servers to provide various services (such as speech recognition and speech synthesis) to their clients. The MRCPv2 protocol allows client hosts to control media service resources such as speech synthesizers, recognizers, verifiers and identifiers residing in servers on the network. http://www.iana.org/assignments/mrcpv2-parameters/mrcpv2-parameters.xml, http://en.wikipedia.org/wiki/Media_Resource_Control_Protocol **[speech]**

**Multimedia Messaging Service (MMS):** The Multimedia Messaging Service (MMS) is a standard way to send messages that include multimedia content to and from mobile phones. It extends the core SMS (Short Message Service) capability that allows exchange of text messages only up to 160 characters in length. http://en.wikipedia.org/wiki/Multimedia_Messaging_Service **[travel]**

**Multimodality:** A multimodal system processes input and delivers output in several modalities, including text, image, speech and tactile information. **[dbpedia] [math] [speech]**

**MySQL:** MySQL is the world's most used open source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. http://en.wikipedia.org/wiki/MySQL, http://www.mysql.com **[search] [sensors]**

# N

**N-gram:** An n-gram is a contiguous sequence of n items from a given sequence of text or speech. An n-gram could be any combination of letters. However, the items in question can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus. http://en.wikipedia.org/wiki/N-gram **[emergency]**

**N-Quad:** N-Quad is a format that extends N-Triples with context. Each triple in an N-Quads document can have an optional context value. http://sw.deri.org/2008/07/n-quads/, http://en.wikipedia.org/wiki/N-gram **[dbpedia]**

**N-Triples:** N-Triples is a basic text serialization of RDF graphs, with one line per RDF triple. http://en.wikipedia.org/wiki/N-Triples **[dbpedia]**

**N3:** N3 is a shorthand non-XML serialization of RDF models, designed with human-readability in mind: N3 is much more compact and readable than XML RDF notation. http://en.wikipedia.org/wiki/Notation3 **[search]**

**Named Entity Recognition (NER):** Named Entity Recognition (NER) labels sequences of words in a text which are the names of things, such as person and company names, or gene and protein names. http://nlp.stanford.edu/software/CRF-NER.shtml **[dbpedia]**

**National Center for Biomedical Ontology (NCBO):** The goal of the National Center for Biomedical Ontology is to support biomedical researchers in their knowledge-intensive work, by providing online tools and a Web portal enabling them to access, review, and integrate disparate ontological resources in all aspects of biomedical investigation and clinical practice. A major focus involves the use of biomedical ontologies to aid in the management and analysis of data derived from complex experiments. http://www.bioontology.org **[standards]**

**Natural Language Processing (NLP):** Natural Language Processing (NLP) is an interdisciplinary field that uses computational methods to investigate the properties of human language and to model the cognitive mechanisms underlying the understanding and production of language (scientific focus), and to develop novel practical applications involving the intelligent processing of human language by computer (engineering focus). http://nlp.shef.ac.uk **[dbpedia] [emergency] [speech] [urban]**

**Near Field Communication (NFC):** Near Field Communication (NFC) is a set of standards for smartphones and similar devices to establish radio communication with each other by touching them together or bringing them into close proximity. Present and anticipated applications include contactless transactions such as Wi-Fi. Communication is also possible between an NFC device and an unpowered NFC chip, called a "tag". NFC standards cover communications protocols and data exchange formats, and are based on existing radio-frequency identification (RFID) standards. http://en.wikipedia.org/wiki/Near_field_communication **[sensors]**

**Nearest neighbor:** Nearest neighbor is a method of calculating distances between clusters. The distances between possible nearest neighbors are calculated e.g. with the Euclidean or City-Block metrics. **[sensors] [simil] [urban]**

**Neogeography:** Movement in geography that puts cartography into the reach of non-professional users and developers. http://en.wikipedia.org/wiki/Neogeography **[eco]**

**NIST:** National Institute of Science and Technology. http://www.nist.gov/ **[standards]**

# O

**OASIS:** OASIS (Organization for the Advancement of Structured Information Standards) is a not-for-profit consortium that drives the development, convergence and adoption of open standards for the global information society. https://www.oasis-open.org **[standards]**

**OAuth:** OAuth is an open protocol to allow secure authorization in a simple and standard method for web, mobile and desktop applications. http://oauth.net **[standards]**

**OBO:** Open Biomedical Ontologies (OBO) is an effort to create controlled vocabularies for shared use across different biological and medical domains. OBO is a part of the resources of the U.S. National Center for Biomedical Ontology, where it will form a central element of the NCBO's BioPortal. The OBOFoundry collects biomedical ontologies. http://en.wikipedia.org/wiki/Open_Biomedical_Ontologies, http://en.wikipedia.org/wiki/OBO_Foundry **[standards]**

**Observations & Measurements (O&M):** Observations and Measurements (O&M) is an international standard which defines a conceptual schema encoding for observations, and for sampling features in observations. While the O&M standard was developed in the context of geographic information systems, the model is derived from generic patterns and is not limited to spatial information. O&M defines a core set of properties for an observation. It provides the response model for the Sensor Observation Service (SOS). http://en.wikipedia.org/wiki/Observations_and_Measurements **[sensors]**

**OGC:** Open Geospatial Consortium. http://www.opengeospatial.org **[sensors] [standards]**

**Ogg Vorbis:** Ogg Vorbis is an audio compression format. It is roughly comparable to other formats used to store and play digital music, such as MP3, VQF, AAC. It is different from these other formats because it is completely free, open, and unpatented. http://www.vorbis.com/faq/#what **[standards]**

**OMDoc:** OMDoc is a markup format and data model for Open Mathematical Documents. https://trac.omdoc.org/OMDoc **[math]**

**Ontology:** An ontology formally represents knowledge as a set of concepts within a domain, and the relationships among those concepts. In computer science and information science, an ontology formally represents knowledge as a set of concepts within a domain, and the relationships among those concepts. It can be used to describe the domain and to reason about the entities within that domain. http://en.wikipedia.org/wiki/Ontology_(information_science) **[eco] [emergency] [math] [search] [sensors] [standards] [travel] [urban]**

**OOI:** Object of interest, cf. POI (Point of interest). **[eco]**

**Open Directory Project (DMOZ):** The Open Directory Project is the largest, most comprehensive human-edited directory of the Web. It is constructed and maintained by a vast, global community of volunteer editors. http://www.dmoz.org/World/ **[dbpedia] [search]**

**Open Mashup Alliance (OMA):** The Open Mashup Alliance (OMA) is an organization charted to promote the adoption of mashup solutions in the enterprise through the evolution of enterprise mashup standards like an open enterprise mashup markup language. http://www.openmashup.org **[eco]**

**OpenGL:** OpenGL (Open Graphics Library) is a cross-language, multi-platform API for rendering 2D and 3D computer graphics. The API is typically used to interact with a graphics processing unit to achieve hardware-accelerated rendering. OpenGL is widely used in CAD, virtual reality, visualization, flight simulation, and video games. It is managed by the non-profit technology consortium Khronos Group. http://www.opengl.org **[standards]**

**OpenID:** OpenID is an open standard for user authentication in a decentralized manner, eliminating the need for services to provide their own ad hoc systems and allowing users to consolidate their digital identities. Users may create accounts with their preferred OpenID identity providers, and then use those accounts as the basis for signing on to any website which accepts OpenID authentication. http://en.wikipedia.org/wiki/OpenID, http://openid.net **[standards] [travel]**

**OpenMath:** OpenMath is a standard for representing mathematical objects with their semantics, allowing them to be exchanged between computer programs, stored in databases, or published on the worldwide web. OpenMath provides a sublanguage for defining content dictionaries (CDs). http://www.openmath.org **[math]**

**OpenStreetMap (OSM):** OpenStreetMap (OSM) is a collaborative project to create a free editable map of the world. Rather than the map itself, the data generated by the OpenStreetMap project are considered its primary output. These data are available for use in both traditional applications, like Craigslist and Foursquare to replace Google Maps, and for more unusual roles, like replacing default data included with GPS receivers. http://en.wikipedia.org/wiki/OpenStreetMap, http://www.openstreetmap.org **[eco] [travel] [urban]**

**Operations Research:** Operations research, or operational research, deals with the application of advanced analytical methods to decision making. It is often considered to be a sub-field of mathematics. http://en.wikipedia.org/wiki/Operations_research **[urban]**

**OSGi:** The OSGi Alliance is a worldwide consortium of technology innovators that advances open specifications that enable the modular assembly of software built with Java technology. The alliance provides specifications, reference implementations, test suites and certification to foster a valuable cross-industry ecosystem. http://www.osgi.org/Main/HomePage **[dbpedia] [standards]**

**OWL Horst reasoning:** OWL Horst reasoning is a ruleset which incorporates RDFS and RDF entailment and extends that with some basic support for OWL. As such, it is very much an extension of Description Logic Programming (DLP). http://answers.semanticweb.com/questions/9125/what-is-owl-horst-reasoning **[urban]**

**OWL:** The OWL 2 Web Ontology Language (OWL 2) is an ontology language for the Semantic Web with formally defined meaning. OWL 2 ontologies comprise classes, properties, individuals, and data values. OWL 2 ontologies can be used along with information written in RDF, and OWL 2 ontologies themselves are primarily exchanged as RDF documents. http://www.w3.org/TR/owl2-overview/ **[dbpedia] [eco] [standards] [urban]**

# P

**P2000:** P2000 is a one-way communications network for pagers based on Motorola's FLEX-protocol in the Netherlands. The network is used by all emergency services and provides nationwide coverage. http://en.wikipedia.org/wiki/P2000_(network) **[emergency]**

**Panoramio:** Panoramio is a geolocation-oriented photo sharing website. Accepted photos uploaded to the site can be accessed as a layer in Google Earth and Google Maps, with new photos being added at the end of every month. The site's goal is to allow Google Earth users to learn more about a given area by viewing the photos that other users have taken at that place. The website is available in several languages. http://www.panoramio.com, http://en.wikipedia.org/wiki/Panoramio **[eco] [sensors]**

**Parametric statistical model:** A parametric model or parametric family or finite-dimensional model is a family of distributions that can be described using a finite number of parameters. http://en.wikipedia.org/wiki/Parametric_model **[sensors]**

**Peer-to-peer (P2P):** A peer-to-peer (P2P) computer network is one in which each computer in the network can act as a client or server for the other computers in the network, allowing shared access to various resources such as files, peripherals, and sensors without the need for a central server. P2P is a distributed application architecture that partitions tasks or workloads among peers. P2P networks can be used for sharing content such as audio, video, data, or anything in digital format. http://en.wikipedia.org/wiki/Peer-to-peer **[speech] [travel]**

**Personalization:** Personalization technology enables the dynamic insertion, customization or suggestion of content in any format that is relevant to the individual user, based on the user's implicit behavior and preferences, and explicitly given details. Of particular interest is search personalization, i.e. of incorporating information about the user needs into query processing. http://en.wikipedia.org/wiki/Personalization **[eco] [search] [travel]**

**Petri net:** A Petri net (also known as a place/transition net or P/T net) is a mathematical modeling language for the description of distributed systems. A Petri net is a directed bipartite graph, in which the nodes represent transitions (i.e. events that may occur, signified by bars) and places (i.e. conditions, signified by circles). The directed arcs describe which places are pre- and/or postconditions for which transitions (signified by arrows) occur. Petri nets offer a graphical notation for stepwise processes that include choice, iteration, and concurrent execution, and they have an exact mathematical definition of their execution semantics, with a well-developed mathematical theory for process analysis. http://en.wikipedia.org/wiki/Petri_net **[eco]**

**Phone:** A phone is a realization of a phoneme. **[speech]**

**Phoneme:** A phoneme is the smallest segmental unit of sound employed to form meaningful contrasts between utterances (cf. "bill" and "kill"). It is an abstraction of a set (or equivalence class) of speech sounds (phones) which are perceived as equivalent to each other in a given language. http://en.wikipedia.org/wiki/Phoneme **[speech]**

**Phonetic dictionary:** Language model of word pronunciation, where each word present in the language model has an entry with one or more possible pronunciations (also called transcriptions). **[speech]**

**Phonetics:** Phonetics studies the sounds of human speech. It is concerned with the physical properties of speech sounds or signs (phones): their physiological production, acoustic properties, auditory perception, and neurophysiological status. http://en.wikipedia.org/wiki/Phonetics **[speech]**

**PHP:** PHP is an open source general-purpose server-side scripting language originally designed for Web development to produce dynamic Web pages. The code is interpreted by a Web server with a PHP processor module which generates the resulting Web page. PHP is installed on more than 20 million Web sites and 1 million Web servers. Software that uses PHP includes MediaWiki, Joomla, Wordpress, Concrete5, MyBB, and Drupal. http://en.wikipedia.org/wiki/PHP **[eco] [emergency] [simil] [speech] [standards] [urban]**

**Physical mashup:** A physical mashup may include sensor and actuator networks, embedded devices, electronic appliances and digitally enhanced everyday objects, so that they have their place in the physical world, and they are connected on the web with common techniques. **[eco] [sensors]**

**Picasa:** Picasa is a photo sharing platform. http://picasa.google.com **[sensors]**

**Pipe(s):** A pipe is a sequence of processes accepting inputs and delivering an output. **[eco] [emergency] [speech] [urban] [travel]**

**Point of interest (POI):** Points of interest (POI) are locations that may be important for somebody for some reason. Usually they are defined by their geographic longitude and latitude. **[dbpedia] [eco] [emergency] [standards] [travel] [urban]**

**Polymath Project:** The Polymath Project is a collaboration among mathematicians to solve important and difficult mathematical problems by coordinating many mathematicians to communicate with each other on finding the best route to the solution. http://en.wikipedia.org/wiki/Polymath_Project **[math]**

**Polysemy:** Polysemy is the capacity of a sign (e.g., a word, phrase, etc.) to have multiple related meanings. It is usually regarded as distinct from homonymy, in which the multiple meanings of a word may be unconnected or unrelated. http://en.wikipedia.org/wiki/Polysemy **[search]**

**Portable Document Format (PDF):** Portable Document Format (PDF) is a file format for platform-independent representation of documents. Each PDF file encapsulates a complete description of a fixed-layout flat document, including the text, fonts, graphics, and other information needed to display it. http://en.wikipedia.org/wiki/Portable_Document_Format **[standards]**

**Programming by demonstration:** Programming by demonstration (PbD) is an end-user development technique for teaching a computer or a robot new behaviors by demonstrating the task to transfer directly instead of programming it through machine commands. http://en.wikipedia.org/wiki/Programming_by_demonstration **[eco]**

**Pronunciation:** Pronunciation refers to the use the correct stress, rhythm and intonation of a word or word sequence in spoken language. Symbols from a phonetic alphabet (e.g. ARPA, IPA, SAMPA) are used to transcribe the pronunciation of a spoken utterance. http://en.wikipedia.org/wiki/Pronunciation **[speech]**

**Prosody:** Prosody is the rhythm, stress, and intonation of speech. Prosody may reflect various features of the speaker or the utterance: the emotional state of the speaker; the form of the utterance (statement, question, or command); the presence of irony or sarcasm; emphasis, contrast, and focus; or other elements of language that may not be encoded by grammar or choice of vocabulary. In terms of acoustics, the prosodics of oral languages involve variation in syllable length, loudness, and pitch. http://en.wikipedia.org/wiki/Prosody_(linguistics) **[speech]**

**Protégé:** Protégé is a free, open source ontology editor and knowledge base framework. The Protégé platform supports modeling ontologies via the Protégé-Frames and Protégé-OWL editors. Protégé ontologies can be exported into a variety of formats including RDFS, OWL, and XML Schema. http://protege.stanford.edu **[standards]**

**Public Switched Telephone Network (PSTN):** The Public Switched Telephone Network (PSTN) is the wired phone system over which landline telephone calls are made. The PSTN relies on circuit switching. To connect one phone to another, the phone call is routed through numerous switches operating on a local, regional, national or international level. http://electronics.howstuffworks.com/telephone-country-codes1.htm **[eco] [speech]**

**Python:** Python is a general-purpose, interpreted high-level programming language whose design philosophy emphasizes code readability. Python has a comprehensive standard library. Interpreters are available for many operating systems. http://www.python.org **[eco] [simil] [speech] [standards]**

# Q

**Quality of service (QoS):** The quality of service (QoS) refers to several related aspects of telephony and computer networks that allow the transport of traffic with special requirements. http://en.wikipedia.org/wiki/Quality_of_service **[dbpedia]**

**Query refinement:** Query refinement improves an initial query. Search engines may also automatically refine queries. For instance, Google OneBox promotes a vertical search database near the top of the search result; offers a "did you mean" link with the correct spelling near the top of the results; offers related search results in the search results. Some engines also suggest a variety of related search queries. Some search toolbars also aim to help searchers autocomplete their search queries by offering a list of most popular queries which match the starting letters that a searcher enters into the search box. http://seotermglossary.com/query-refinement/ **[search]**

# R

**Radio Frequency IDentification (RFID):** Radio Frequency IDentification (RFID) is the use of a wireless non-contact system that uses radio-frequency electromagnetic fields to transfer data from a tag attached to an object, for the purposes of automatic identification and tracking. Some tags require no battery and are powered and read at short ranges via magnetic fields (electromagnetic induction). Others use a local power source and emit radio waves (electromagnetic radiation at radio frequencies). The tag contains electronically stored information which can be read from up to several meters (yards) away. Unlike a bar code, the tag does not need to be within line of sight of the reader and may be embedded in the tracked object. http://en.wikipedia.org/wiki/Radio-frequency_identification **[eco] [sensors]**

**Ranking:** Ranking is the ordering of search results according to their assumed relevance, i.e. to what extent the topic of a result matches the topic of the query or information need. **[dbpedia] [eco] [emergency] [search] [urban]**

**RDF Schema (RDFS):** RDF Schema (RDFS) provides basic elements for the description of RDF ontologies/vocabularies, intended to structure RDF resources. A few RDFS components are included in the more expressive Web Ontology Language (OWL). http://en.wikipedia.org/wiki/RDF_Schema **[standards]**

**RDFa:** RDFa (Resource Description Framework in Attributes) provides a set of markup attributes to embed RDF graphs into HTML or XML. Thus, the existing visual, human-readable Web page content is augmented with machine-readable hints. http://www.w3.org/TR/rdfa-core/, http://www.w3.org/TR/xhtml-rdfa-primer/ **[eco] [math] [simil] [standards] [travel]**

**Recommendation system:** Recommender systems or recommendation systems are a subclass of information filtering system that seek to predict the 'rating' or 'preference' that a users would give to an item (such as music, books, or movies) or social element (e.g. people or groups) they had not yet considered, using a model built from the characteristics of an item (content-based approaches) or the user's social environment (collaborative filtering approaches). http://en.wikipedia.org/wiki/Recommender_system **[dbpedia] [eco] [math] [search] [simil] [travel] [urban]**

**Recurrent Neural Network (RNN):** A Recurrent Neural Network (RNN) is a class of neural network where connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. Unlike feed-forward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs. This makes them applicable to tasks such as unsegmented connected handwriting recognition, where they have achieved the best known results. http://en.wikipedia.org/wiki/Recurrent_neural_network **[urban]**

**Regular expression:** A regular expression ('regex') provides a concise and flexible means to "match" (specify and recognize) strings of text, such as particular characters, words, or patterns of characters. http://en.wikipedia.org/wiki/Regular_expression **[emergency] [simil]**

**Relax NG:** Relax NG is a schema language for XML. It is simple, easy to learn, has both an XML syntax and a compact non-XML syntax, and some additional practical features. http://relaxng.org **[standards]**

**Relevance feedback:** Relevance feedback in information retrieval takes the results that are initially returned from a given query and submits them for relevance decision. Information about whether or not those results are relevant is used in a follow-up query. http://en.wikipedia.org/wiki/Relevance_feedback **[search] [simil]**

**Representational State Transfer (REST):** Representational State Transfer (REST) is a software architecture for distributed systems. REST facilitates the transaction between web servers by allowing loose coupling between different services. REST-style architectures consist of clients and servers. Clients initiate requests to servers; servers process requests and return appropriate responses. Requests and responses are built around the transfer of representations of resources. A resource can be essentially any coherent and meaningful concept that may be addressed. Key goals of REST include the scalability of component interactions, the generality of interfaces, an independent deployment of components, and intermediary components to reduce latency, enforce security and encapsulate legacy systems. http://en.wikipedia.org/wiki/Representational_state_transfer **[dbpedia] [eco] [math] [sensors] [simil] [speech] [standards] [travel] [urban]**

**Request for Comments (RFC):** Requests for Comments (RFC) documents are the official record for Internet specifications, protocols, procedures, and events of the Internet Engineering Task Force (IETF). http://en.wikipedia.org/wiki/Request_for_Comments **[standards]**

**Resource Description Framework (RDF):** The Resource Description Framework (RDF) data model is similar to entity-relationship or class diagrams, as it is based upon the idea of making statements about resources (in particular Web resources) in the form of subject–predicate–object expressions. These expressions are known as triples in RDF terminology. The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object. http://en.wikipedia.org/wiki/Resource_Description_Framework **[dbpedia] [eco] [emergency] [math] [search] [sensors] [simil] [standards] [travel] [urban]**

**Reverse/inverse geocoding:** Reverse/inverse geocoding is the opposite of geocoding: finding an associated textual location such as a street address, from geographic coordinates, or stating which geographical coordinates of a sensor node belong to the area. http://en.wikipedia.org/wiki/Geocoding **[sensors]**

**RGB color:** The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. The main purpose of the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers. Before the electronic age, the RGB color model already had a solid theory behind it, based on human perception of colors. http://en.wikipedia.org/wiki/RGB_color_model **[standards]**

**Rocchio algorithm:** The Rocchio algorithm is based on a relevance feedback method from the Vector-Space-Model-based SMART Information Retrieval System around the year 1970. The algorithm assumes that most users have a general conception of which documents should be denoted as relevant or non-relevant. Therefore, the user's search query is revised to include an arbitrary percentage of relevant and non-relevant documents as a means of increasing the search engine's recall, and possibly the precision as well. The number of relevant and non-relevant documents allowed to enter a query is dictated by original query weights, related document weights and non-related document weights. http://en.wikipedia.org/wiki/Rocchio_algorithm **[simil]**

**RSOE EDIS:** Emergency and Disaster Information Service. http://hisz.rsoe.hu/alertmap/index2.php **[emergency]**

**RSOE:** Hungarian National Association of Radio Distress-Signalling and Infocommunications. http://hisz.rsoe.hu **[emergency]**

**RSS Feed:** RSS Rich Site Summary (Really Simple Syndication) is a family of web feed formats used to publish frequently updated works such as blog entries, news headlines, audio, and video in a standardized format. An RSS document (which is called a "feed", "web feed", or "channel") includes full or summarized text, plus metadata such as publishing dates and authorship. http://en.wikipedia.org/wiki/RSS **[eco] [emergency] [simil] [speech]**

**Ruby:** According to the Ruby promoters, Ruby is a dynamic, open source programming language with a focus on simplicity and productivity. It has an elegant syntax that is natural to read and easy to write. http://www.ruby-lang.org/en/ **[simil] [speech]**

**Rule Markup Language (RuleML):** The Rule Markup Language (RuleML) is a markup language for both forward (bottom-up) and backward (top-down) rules in XML for deduction, rewriting, and further inferential–transformational tasks. http://en.wikipedia.org/wiki/RuleML **[sensors] [standards]**

**Rule-based language model:** Rule-based language models complement the most widely used class of probabilistic language models, the so-called n-grams. N-grams only model short-range dependencies. Rule-based language models capture long-range dependencies that are present in natural language, such as subject–verb agreement. http://www.tik.ee.ethz.ch/spr/publications/Kaufmann:12.pdf **[speech] [urban]**

**Rule:** A rule normally is a condition–action rule, also called a production or production rule, of the form "if condition then action". http://www.cse.unsw.edu.au/~billw/aidict.html#firstC **[dbpedia] [eco] [emergency] [math] [sensors] [simil] [speech] [standards] [urban]**

# S

**Sampling frequency:** The sampling rate, sample rate, or sampling frequency defines the number of samples per unit of time (usually seconds) taken from a continuous signal to make it a discrete signal. For time-domain signals, the unit for sampling rate is Hertz (Hz). http://en.wikipedia.org/wiki/Sampling_rate **[sensors] [speech]**

**SAWSDL:** SAWSDL (Semantic Annotations for WSDL and XML Schema) defines a set of extension attributes for the Web Services Description Language (WSDL) and XML Schema definition language. SAWSDL defines how semantic annotation is accomplished using references to conceptual semantic models, e.g. ontologies. Semantic annotations for WSDL and XML Schema (SAWSDL) provide mechanisms by which concepts from the semantic models can be referred using annotations. http://en.wikipedia.org/wiki/SAWSDL **[eco]**

**Scalable Vector Graphics (SVG):** Scalable Vector Graphics (SVG) is an XML-based format for two-dimensional vector graphics, both static and dynamic (i.e., interactive or animated). SVG images and their behaviors are defined in XML text files. As XML files, SVG images can be created and edited with any text editor. All major modern web browsers support SVG to some degree. http://en.wikipedia.org/wiki/Scalable_Vector_Graphics **[math] [standards]**

**Second Life:** Second Life is an online world in which residents create virtual representations of themselves, called avatars, and interact with other avatars, places or objects. http://computer.howstuffworks.com/internet/social-networking/networks/second-life.htm **[eco]**

**Semantic mashup:** A semantic mashup applies semantic methods, both symbolic and probabilistic ones, such as semantic annotation, information extraction, or speech recognition. **[eco]**

**Semantic Web for Earth and Environmental Terminology (SWEET):** SWEET ontologies are written in the OWL ontology language. SWEET 2.3 is highly modular with 6,000 concepts in 200 separate ontologies. http://sweet.jpl.nasa.gov **[emergency] [sensors]**

**Semi-supervised learning:** Machine learning technique that makes use of both labeled and unlabeled data for training—typically a small amount of labeled data with a large amount of unlabeled data. http://en.wikipedia.org/wiki/Semi-supervised_learning **[urban]**

**Sensor and Sensor Network (SSN) ontology:** The Sensor and Sensor Network (SSN) ontology is a domain-independent and end-to-end model for sensing applications by merging sensor-focused (e.g. SensorML), observation-focused (e.g. Observation & Measurement) and system-focused views. http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/ **[sensors]**

**Sensor measurement:** A good sensor measurement is sensitive to the measured property only, is insensitive to any other property likely to be encountered in its application, and does not influence the measured property. In practice systematic and random errors are observed. http://en.wikipedia.org/wiki/Sensor **[sensors]**

**Sensor Observation Service (SOS):** The Sensor Observation Service (SOS) is a web service to query real-time sensor data and sensor data time series and is part of the Sensor Web. The offered sensor data comprise descriptions of sensors themselves, which are encoded in the Sensor Model Language (SensorML), and the measured values in the Observations and Measurements (O&M) encoding format. The web service is defined by the Open Geospatial Consortium (OGC). http://en.wikipedia.org/wiki/Sensor_Observation_Service **[eco] [sensors]**

**Sensor Planning Service (SPS):** Standard web interface for queries that provide information about the capabilities of a sensor and how to task the sensor. http://www.opengeospatial.org/standards/sps **[eco] [sensors]**

**Sensor Web Enablement (SWE):** The OGC's Sensor Web Enablement (SWE) standards enable developers to make all types of sensors, transducers and sensor data repositories discoverable, accessible and usable via the Web. http://www.opengeospatial.org/ogc/markets-technologies/swe **[sensors] [standards]**

**Sensor Web:** The sensor web is a type of sensor network that is especially well suited for environmental monitoring. It is associated with a sensing system which heavily utilizes the web. OGC's Sensor Web Enablement (SWE) framework defines a suite of web service interfaces and communication protocols for sensor (network) communication. http://en.wikipedia.org/wiki/Sensor_web **[sensors]**

**Sensor:** A sensor is a converter that measures a physical quantity and converts it into a signal which can be read by an observer or by an instrument. http://en.wikipedia.org/wiki/Sensor **[eco] [emergency] [sensors] [travel] [urban]**

**Service-oriented architectures (SOA):** A service-oriented architecture (SOA) is a set of principles and methodologies for designing and developing software in the form of interoperable services. These services are well-defined business functionalities that are built as software components that can be reused for different purposes. SOA generally provides a way for consumers of services, such as web-based applications, to be aware of available SOA-based services. http://en.wikipedia.org/wiki/Service-oriented_architecture **[eco] [speech]**

**Session:** A session is a semi-permanent interactive information interchange, also known as a dialog, a conversation or a meeting, between two or more communicating devices, or between a computer and user. A session is set up or established at a certain point in time, and torn down at a later point in time. http://en.wikipedia.org/wiki/Session_(computer_science) **[search] [speech] [travel]**

**SHA:** The Secure Hash Algorithm (SHA) is a family of cryptographic hash functions published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS). http://en.wikipedia.org/wiki/Secure_Hash_Algorithm **[standards]**

**Shapefile:** A shapefile is a popular geospatial vector data format for geographic information systems software. Shapefiles spatially describe geometries: points, polylines, and polygons, representing e.g. water wells, rivers, and lakes, respectively. http://en.wikipedia.org/wiki/Shapefile **[urban]**

**SIP address:** A Voice IP SIP address is a unique identifier for each user on the network, just like a phone number identifies each user on the global phone network, or an e-mail address. It is also known as a SIP URI (Uniform Resource Identifier). http://voip.about.com/od/sipandh323/g/What-Is-A-Sip-Address.htm **[speech]**

**SKOS:** SKOS (Simple Knowledge Organization System) adapts Knowledge Organization Systems (KOS) such as thesauri, classification schemes, subject heading lists, and taxonomies within the framework of the Semantic Web. http://www.w3.org/2004/02/skos/ **[math] [standards] [travel]**

**Sliding windows:** Sliding windows ("windowing") is used by the Internet's Transmission Control Protocol (TCP) as a method of controlling the flow of packets between two computers or network hosts. Sliding windows is a method by which multiple packets of data can be affirmed with a single acknowledgment. http://searchnetworking.techtarget.com/definition/sliding-windows **[sensors]**

**SMS:** Short Messaging Service. http://en.wikipedia.org/wiki/Short_Message_Service **[eco] [emergency] [speech] [travel]**

**SMTP:** Simple Mail Transfer Protocol (SMTP) is an Internet standard for electronic mail (e-mail) transmission across Internet Protocol (IP) networks. http://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol **[standards]**

**SOAP:** SOAP (Simple Object Access Protocol) is a protocol specification for exchanging structured information in the implementation of web services in computer networks. It relies on XML for its message format, and usually relies on other application layer protocols, most notably HTTP and SMTP, for message negotiation and transmission. http://en.wikipedia.org/wiki/SOAP **[eco] [simil] [speech] [standards] [urban]**

**Social awareness stream:** Social awareness streams are an important feature of applications such as Twitter or Facebook. Personal awareness streams usually allow users to post short, natural-language messages as a personal stream of data that is being made available to other users. Social awareness stream are the aggregation of such personal awareness streams. They usually contain a set of short messages from different users. http://www.markusstrohmaier.info/documents/2010_SemSearch2010_Tweetonomies.pdf **[travel]**

**SPARQL:** SPARQL is an RDF query language that is able to retrieve and manipulate data stored in RDF format. SPARQL allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns. Implementations for multiple programming languages exist. There exist tools that allow one to connect and semi-automatically construct a SPARQL query for a SPARQL endpoint, and tools that translate SPARQL queries to other query languages, for example to SQL and to XQuery. http://en.wikipedia.org/wiki/SPARQL **[dbpedia] [eco] [emergency] [math] [search] [simil] [standards] [travel] [urban]**

**Spectral techniques:** Spectral techniques are used in applied mathematics and scientific computing to numerically solve certain differential equations, often involving the use of the Fast Fourier Transform. The idea is to write the solution of the differential equation as a sum of certain "basis functions" (for example, as a Fourier series which is a sum of sinusoids) and then to choose the coefficients in the sum in order to satisfy the differential equation as well as possible. http://en.wikipedia.org/wiki/Spectral_method **[sensors]**

**Speech Application Language Tags (SALT):** Speech Application Language Tags (SALT) is an XML-based markup language that is used in HTML and XHTML pages to add voice recognition capabilities to web-based applications. SALT enables multimodal and telephony-enabled access to information, applications, and Web services from PCs, telephones, tablet PCs, and wireless personal digital assistants (PDAs). http://en.wikipedia.org/wiki/Speech_Application_Language_Tags **[speech]**

**Speech mashup:** A speech mashup is a web service that implements speech technologies, including automatic speech recognition and text-to-speech synthesis. **[speech]**

**Spreading activation:** Spreading activation is a method for searching associative networks, neural networks, or semantic networks. The search process is initiated by labeling a set of source nodes with weights or "activation" and then iteratively propagating or "spreading" that activation out to other nodes linked to the source nodes. Most often these "weights" are real values that decay as activation propagates through the network. http://en.wikipedia.org/wiki/Spreading_activation **[emergency]**

**SQL:** SQL (Structured Query Language) is a programming language designed for managing data in relational database management systems (RDBMS). http://en.wikipedia.org/wiki/SQL **[search] [sensors] [standards]**

**SQLite:** SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite is said to be the most widely deployed SQL database engine in the world. The source code for SQLite is in the public domain. http://www.sqlite.org **[standards]**

**Statistical language model (SLM):** A statistical language model that can estimate the distribution of natural language as accurate as possible. A statistical language model (SLM) is a probability distribution P(s) over strings S that attempts to reflect how frequently a string S occurs as a sentence. The n-gram model is the most widely used SLM today. http://homepages.inf.ed.ac.uk/lzhang10/slm.html **[speech]**

**Statistical pattern recognition:** Statistical pattern recognition is applied to new and emerging applications—such as data mining, web searching, multimedia data retrieval, face recognition, and cursive handwriting recognition—that require robust and efficient pattern recognition techniques. Statistical decision making and estimation are regarded as fundamental to the study of pattern recognition. **[speech]**

**Statistical Unit Node Set (SUNS):** SUNS is used in web-based machine learning. The Statistical Unit Node Set (SUNS) is defined for a statistical unit in a population. It includes all probabilistic nodes that correspond to all actual and potential statements in which the unit is either subject or object. A data matrix with triples from the SUNS can be used both for estimating SUNS in the whole data matrix (transduction) and for the SUNS in the population (induction). The learned probabilistic statements can be stored in the knowledge base as weighted triples using a number of approaches, e.g., using reification. http://wwwbrauer.informatik.tu-muenchen.de/~trespvol/papers/tresp_irmles09_MaQLK.pdf **[urban]**

**Stemming algorithm:** A stemming algorithm is used for stemming, i.e. reducing inflected (or sometimes derived) words to their stem, base or root form, generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. http://en.wikipedia.org/wiki/Stemming **[simil]**

**Stop word:** Stop words are words which are filtered out prior to, or after, processing of natural language data (text). http://en.wikipedia.org/wiki/Stop_words **[simil]**

**Stream processing engines (SPE):** Stream processing engines (SPE) are specifically designed to deal with streaming data. They perform SQL-like queries on the streaming data without necessarily storing anything. **[sensors]**

**Streaming media:** Streaming media is multimedia that is constantly received by and presented to an end-user while being delivered by a provider. http://en.wikipedia.org/wiki/Streaming_media **[speech]**

**Support vector machines (SVMs):** Support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. http://en.wikipedia.org/wiki/Support_vector_machine **[urban]**

**Syndication:** Syndication makes website material available to multiple other sites. Most commonly, syndication refers to the distribution of web feeds. http://en.wikipedia.org/wiki/Web_syndication **[simil] [standards]**

**Synonymy:** Synonymy is a kind of semantic relation. Two words (or phrases) are synonyms when they have the same meaning. Terms with subtle differences between meanings are termed near-synonyms. http://www.iva.dk/bh/lifeboat_ko/CONCEPTS/synonymy.htm **[emergency] [search] [simil] [standards]**

## T

**TCP/IP:** TCP/IP is the Internet protocol suite including the main protocols Transmission Control Protocol (TCP) and Internet Protocol (IP). TCP/IP provides end-to-end connectivity specifying how data should be formatted, addressed, transmitted, routed and received at the destination. It has four abstraction layers, each with its own protocols. http://en.wikipedia.org/wiki/Internet_protocol_suite **[standards]**

**Telco mashups:** Telecom mashups integrating web mashups with services distributed over company-own PSTN (Public Switched Telephone Network) telephone lines. **[eco] [speech]**

**Text Encoding Initiative (TEI):** The Text Encoding Initiative (TEI) is a consortium which collectively develops and maintains a standard for the representation of texts in digital form. Its chief deliverable is a set of guidelines which specify encoding methods for machine-readable texts, chiefly in the humanities, social sciences and linguistics. http://www.tei-c.org/index.xml **[standards]**

**Text REtrieval Conference (TREC):** The Text REtrieval Conference (TREC) supports research within the information retrieval community by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies. http://trec.nist.gov **[search]**

**Text-to-Speech system (TTS):** A Text-to-Speech (TTS) system converts normal language text into speech. **[speech]**

**tf-idf:** tf-idf, term frequency–inverse document frequency, is a numerical statistic which reflects how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others. http://en.wikipedia.org/wiki/Tf-idf **[travel] [dbpedia]**

**Time series analysis:** Time series analysis accounts for the fact that data points taken over time may have an internal structure (such as autocorrelation, trend or seasonal variation) that should be accounted for. http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc4.htm **[sensors]**

**Timestamp:** A timestamp is a sequence of characters or encoded information identifying when a certain event occurred, usually giving date and time of day. http://en.wikipedia.org/wiki/Timestamp **[search] [sensors] [search] [urban]**

**Topology:** Topology analyzes such concepts as space, dimension, and transformation. http://en.wikipedia.org/wiki/Topology **[urban]**

**Transverse Mercator coordinate system:** A transverse Mercator projection is a Mercator projection whose cylinder touches the earth not at the equator, but along a meridian of longitude chosen depending on the area of the map, in order to reduce the scale error. http://www.uwgb.edu/dutchs/fieldmethods/utmsystem.htm **[urban]**

**Tri-phone:** A tri-phone is a sequence of three phonemes, so that one phone is put into context of the preceding and the following phone. **[speech]**

**Turtle:** Terse RDF Triple Language. http://www.w3.org/TeamSubmission/turtle/ **[standards]**

**Tversky model:** The Tversky similarity model assumes that an object is represented by a set of features or attributes. Usually these are binary variables (e.g., voiced or unvoiced consonant) or parts that are present or not, e.g., eyes or tail), but they may also be ordered sets of properties like color or size. An important aspect of Tversky's model is that similarity depends not only on the proportion of features common to the two objects but also on their unique features. http://www.pigeon.psy.tufts.edu/avc/dblough/theory.htm **[simil]**

# U

**Ubiquitous computing:** Ubiquitous computing (ubicomp) is a post-desktop model of human–computer interaction in which information processing has been thoroughly integrated into everyday objects and activities. This paradigm is also described as pervasive computing, ambient intelligence or, more recently, everyware. http://en.wikipedia.org/wiki/Pervasive_Computing **[dbpedia] [eco] [speech] [travel] [urban]**

**UIMA:** UIMA is an Apache open, industrial-strength, scalable and extensible platform for creating, integrating and deploying unstructured information management solutions from powerful text or multimodal analysis and search components. http://uima.apache.org **[dbpedia]**

**UMTS:** Universal Mobile Telecommunications System. http://en.wikipedia.org/wiki/Universal_Mobile_Telecommunications_System **[eco] [urban]**

**Unified Modeling Language (UML):** The Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of object-oriented software engineering. UML includes a set of graphic notation techniques to create visual models of object-oriented software-intensive systems. UML is a standard of both ISO and OMG. http://en.wikipedia.org/wiki/Unified_Modeling_Language **[eco]**

**Unified Theory of Acceptance and Use of Technology (UTAUT):** The Unified Theory of Acceptance and Use of Technology (UTAUT) aims to explain user intentions to use an information system and subsequent usage behavior. The theory holds that four key constructs (performance expectancy, effort expectancy, social influence, and facilitating conditions) are direct determinants of usage intention and behavior. Gender, age, experience, and voluntariness of use are posited to mediate the impact of the four key constructs on usage intention and behavior. http://en.wikipedia.org/wiki/Unified_theory_of_acceptance_and_use_of_technology **[eco]**

**Unit selection synthesis:** The target utterance is created by determining the best chain of candidate units from the voice database (unit selection), typically using a specially weighted decision tree. http://en.wikipedia.org/wiki/Speech_synthesis **[speech]**

**User experience (UX):** User experience (UX) covers all aspects of the end-user interaction with the company, its services, and its products. The first requirement for an exemplary user experience is to meet the exact needs of the customer, without fuss or bother. Next comes simplicity and elegance that produce products that are a joy to own, a joy to use. True user experience goes far beyond giving customers what they say they want, or providing checklist features. In order to achieve high-quality user experience in a company's offerings there must be a seamless merging of the services of multiple disciplines, including engineering, marketing, graphical and industrial design, and interface design. http://www.allaboutux.org/ux-definitions **[eco]**

**User interface (UI):** The user interface is the space where the interaction between humans and machines occurs. Graphical user interfaces (GUIs) are frequent, but human–machine interaction can target all human senses offering e.g. tactile or audio interaction. Good user-oriented design is paramount. It ranges from simple buttons to virtual reality. http://en.wikipedia.org/wiki/User_interface **[dbpedia] [eco] [emergency] [math] [search] [sensors] [standards] [travel]**

**User model:** A user model represents a collection of personal data associated with a specific user. It is the basis for any adaptive changes to the system's behavior. http://en.wikipedia.org/wiki/User_modeling **[math] [simil] [travel]**

**Ushahidi:** Ushahidi is a non-profit tech company that specializes in developing free and open source software for information collection, visualization, and interactive mapping. http://www.ushahidi.com/ **[emergency]**

## V

**Value Difference Metric (VDM):** The Value Difference Metric (VDM) is designed to find reasonable distance values between nominal attribute values, largely ignoring continuous attributes and requiring discretization to map continuous values into nominal values. Updated versions of VDM exist. http://axon.cs.byu.edu/~randy/jair/wilson1.html#Contents **[simil]**

**Vector space model:** The Vector space model or term vector model is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers, such as, for example, index terms. It is used in information filtering, information retrieval, indexing, and relevancy rankings. Its first use was in the SMART Information Retrieval System. http://en.wikipedia.org/wiki/Vector_space_model **[sensors] [simil]**

**Video mashup:** Combination of multiple audiovisual sources. **[eco]**

**Virtual Reality (VR):** Virtual Reality (VR) is using computer technology to create a simulated, three-dimensional world that a user can manipulate and explore while feeling as if he were in that world. Scientists, theorists, and engineers have designed dozens of devices and applications to achieve this goal. http://electronics.howstuffworks.com/gadgets/other-gadgets/virtual-reality.htm **[eco] [standards]**

**Virtual sensor:** Virtual sensors allow the abstraction of data collection away from a fixed set of physical objects. Virtual sensor values are computed based on indirect or abstract measurements derived from multiple, distributed, often heterogeneous data streams. **[travel]**

**Voice-as-a-Service (VaaS):** Speech web service for TTS (Text-to-Speech) or automatic speech recognition (ASR). **[eco] [speech]**

**VoiceXML:** VoiceXML (VXML) is the W3C's standard XML format for specifying interactive voice dialogs between a human and a computer. VoiceXML documents are interpreted by a voice browser. A common architecture is to deploy banks of voice browsers attached to the Public Switched Telephone Network (PSTN) to allow users to interact with voice applications over the telephone. http://en.wikipedia.org/wiki/VoiceXML **[speech]**

**VoIP:** Voice over IP (VoIP, or voice over Internet Protocol) is a technology used by IP telephony to transport phone calls. VoIP systems employ session control protocols to control the set-up and tear-down of calls as well as audio codecs which encode speech allowing transmission over an IP network as digital audio via an audio stream. VoIP is available on many smartphones and Internet devices. http://en.wikipedia.org/wiki/Voice_over_IP **[speech]**

**VRML:** VRML (Virtual Reality Modeling Language) is a standard file format for representing three-dimensional (3D) interactive vector graphics, designed particularly with the World Wide Web in mind. It has been superseded by X3D. http://en.wikipedia.org/wiki/VRML **[standards]**

## W

**W3C Semantic Sensor Network Incubator Group:** The mission of the Semantic Sensor Network Incubator Group was to begin producing ontologies of sensors and sensor networks, and to develop semantic annotations of a key language used by services-based sensor networks. The group delivered a final report (http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/). http://www.w3.org/2005/Incubator/ssn/ **[sensors]**

**W3C time ontology:** The OWL-Time ontology of W3C describes the temporal content of Web pages and the temporal properties of Web services. The ontology provides a vocabulary for expressing facts about topological relations among instants and intervals, together with information about durations, and about datetime information. http://www.w3.org/TR/owl-time/ **[sensors]**

**WATSON:** AT&T speech recognition engine. http://www.research.att.com/projects/WATSON **[speech]**

**Waveform Audio (WAV):** Waveform Audio File Format (WAV) is a Microsoft and IBM audio file format standard for storing an audio bitstream. Though a WAV file can hold compressed audio, the most common WAV format contains uncompressed audio in the linear pulse code modulation (LPCM) format. http://en.wikipedia.org/wiki/WAV **[standards]**

**Web Feature Service Interface Standard (WFS):** The Web Feature Service Interface Standard (WFS) of OGC provides an interface allowing requests for geographical features across the web using platform-independent calls. One can think of geographical features as the "source code" behind a map, whereas the WMS interface or online mapping portals like Google Maps return only an image, which end-users cannot edit or spatially analyze. The XML-based GML furnishes the default payload-encoding for transporting the geographic features, but other formats like shapefiles can also serve for transport. http://en.wikipedia.org/wiki/Web_Feature_Service **[emergency]**

**Web Map Service Interface Standard (WMS):** The OpenGIS Web Map Service Interface Standard (WMS) provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. A WMS request defines the geographic layer(s) and area of interest to be processed. The response to the request is one or more geo-registered map images (returned as JPEG, PNG, etc.) that can be displayed in a browser application. http://www.opengeospatial.org/standards/wms **[standards]**

**Web Mashup Scripting Language (WMSL):** The Web Mashup Scripting Language (WMSL) enables an end-user working from his browser, e.g. not needing any other infrastructure, to quickly write mashups that integrate any web services on the Web. The end-user accomplishes this by writing a web page that combines HTML, metadata in the form of mapping relations, and a small piece of code, or script. The mapping relations enable not only the discovery and retrieval of the WMSL pages, but also affect a new programming paradigm that abstracts many programming complexities from the script writer. http://www.mitre.org/work/tech_papers/tech_papers_07/07_0393/ **[eco]**

**Web of Things (WoT):** The Web of Things (WoT) is a computing concept that describes a future where everyday objects are fully integrated with the Web. The prerequisite for WoT is for the "things" to have embedded computer systems that enable communication with the Web. Such smart devices would then be able to communicate with each other using existing Web standards. http://www.techopedia.com/definition/26834/web-of-things-wot **[eco] [sensors] [standards]**

**Web scraping:** Web scraping (web harvesting or web data extraction) is a computer software technique of extracting information from websites. Usually, such software programs simulate human exploration by either implementing the low-level Hypertext Transfer Protocol (HTTP), or by embedding a fully fledged web browser. http://en.wikipedia.org/wiki/Web_scraping **[simil]**

**Web service:** A Web service is a method of communication between two electronic devices over the World Wide Web. http://en.wikipedia.org/wiki/Web_service **[dbpedia] [eco] [math] [sensors] [speech] [standards] [urban] [travel]**

**Web Services Description Language (WSDL):** The Web Services Description Language (WSDL) is an XML-based language that is used for describing the functionality offered by a Web service. A WSDL description of a web service (also referred to as a WSDL file) provides a machine-readable description of how the service can be called, what parameters it expects, and what data structures it returns. WSDL is often used in combination with SOAP and an XML Schema to provide Web services over the Internet. http://en.wikipedia.org/wiki/Web_Services_Description_Language **[eco] [speech]**

**Web3D:** Web3D refers to all interactive 3D content which are embedded into web pages html, and that we can see through a web browser. http://en.wikipedia.org/wiki/Web3D **[standards]**

**WebCGM:** WebCGM is a profile of the CGM standard (ISO/IEC 8632) that describes how CGM vectors, raster, and hybrid graphics are to be used on the Web. WebCGM 2.1 is a W3C recommendation. https://www.oasis-open.org/committees/cgmo-webcgm/faq.php, http://www.w3.org/TR/2010/REC-webcgm21-20100301/ **[standards]**

**WebGL:** WebGL (Web Graphics Library) is a JavaScript API for rendering interactive 3D graphics and 2D graphics within any compatible web browser without the use of plug-ins. http://en.wikipedia.org/wiki/WebGL **[standards]**

**WebRTC:** WebRTC (Web Real-Time Communication) is an API definition being drafted by the World Wide Web Consortium (W3C). The goal of WebRTC is to enable applications such as voice calling, video chat and P2P file sharing without plugins. http://en.wikipedia.org/wiki/WebRTC **[speech]**

**Weighted finite-state machine (FSM):** A finite-state transducer is a finite automaton whose state transitions are labeled with both input and output symbols. A weighted transducer puts weights on transitions in addition to the input and output symbols. Weights may encode probabilities, durations, penalties, or any other quantity that accumulates along paths to compute the overall weight of mapping an input sequence to an output sequence. Weighted transducers are thus a natural choice to represent the probabilistic finite-state models prevalent in speech processing. http://www.cs.nyu.edu/~mohri/pub/csl01.pdf **[speech]**

**WGS84 coordinate system:** The World Geodetic System is a standard for use in cartography, geodesy, and navigation. Its latest revision is WGS84 (dating from 1984 and last revised in 2004), which was valid up to about 2010. http://en.wikipedia.org/wiki/World_Geodetic_System **[urban]**

**Widget:** A widget is a small application with presentation format that can be installed and executed within a web page by an end-user. Widgets are typically created in dynamic HTML, JavaScript, or Adobe Flash. http://en.wikipedia.org/wiki/Web_widget **[eco] [emergency] [math] [search] [sensors] [travel]**

**Windows Media Audio (WMA):** Windows Media Audio (WMA) is an audio data compression technology developed by Microsoft. The name can be used to refer to its audio file format or its audio codecs. It is a proprietary technology that forms part of the Windows Media framework. http://en.wikipedia.org/wiki/Windows_Media_Audio **[standards]**

**Wolfram Alpha:** Wolfram Alpha is a computational knowledge engine that does not search the web, but performs dynamic computations based on a vast collection of built-in data, algorithms, and methods. Wolfram Alpha's aim is to collect and curate all objective data and to implement every known model, method, and algorithm. http://www.wolframalpha.com/about.html **[math]**

**World Wide Web Consortium (W3C):** The World Wide Web Consortium (W3C) is an international community that develops open standards to ensure the long-term growth of the Web. http://www.w3.org **[dbpedia] [math] [sensors] [speech] [standards] [travel]**

# X

**X3D:** X3D is a royalty-free open standards file format and run-time architecture to represent and communicate 3D scenes and objects using XML. X3D has a rich set of componentized features that can be tailored for use in engineering and scientific visualization, CAD and architecture, medical visualization, training and simulation, multimedia, entertainment, education, and more. http://www.web3d.org/realtime-3d/x3d/what-x3d/ **[standards]**

**XHTML:** XHTML (Extensible HyperText Markup Language) is the XML-compliant encoding of HTML. http://www.w3.org/TR/html5/the-xhtml-syntax.html#the-xhtml-syntax **[math] [speech] [standards]**

**XML Schema (XSD):** An XML Schema (XSD) describes the structure of an XML document. In comparison with a DTD, an XML Schema (coded in XML) enables a more detailed document definition. The current XML Schema 1.1 is a superset of XML Schema 1.0. http://www.w3schools.com/schema/default.asp, http://www.xfront.com/xml-schema-1-1/ **[standards]**

**XML:** The Extensible Markup Language (XML) is a markup language created to structure, store, and transport data by defining a set of rules for encoding documents in a format that is both human-readable and machine-readable. XML is widely used for the representation of arbitrary data structures, for example in web services. Hundreds of XML-based languages have been developed, including RSS, Atom, SVG, MathML, XHTML, and SOAP. XML has also been employed as the base language for communication protocols, such as XMPP. http://en.wikipedia.org/wiki/XML, http://www.w3.org/XML/ **[dbpedia] [eco] [math] [search] [sensors] [simil] [speech] [standards] [travel] [urban]**

**XMPP:** XMPP is the Extensible Messaging and Presence Protocol, a set of open technologies for instant messaging, presence, multi-party chat, voice and video calls, collaboration, lightweight middleware, content syndication, and generalized routing of XML data. http://xmpp.org/about-xmpp/technology-overview/ **[standards]**

**XQuery:** XQuery is a query and functional programming language that is designed to query collections of XML data. The language is based on the XQuery and XPath Data Model (XDM) which uses a tree-structured model of the information content of an XML document, containing seven kinds of nodes: document nodes, elements, attributes, text nodes, comments, processing instructions, and namespaces. http://en.wikipedia.org/wiki/XQuery, http://www.w3.org/XML/Query/ **[simil] [standards]**

**XSLT:** XSLT is a language for transforming XML documents into other XML documents. http://www.w3.org/TR/xslt **[standards] [math]**

**XUL:** XUL (XML User Interface Language) is a user interface markup language implemented as an XML dialect; it allows for graphical user interfaces to be written in a similar manner to Web pages. http://en.wikipedia.org/wiki/XUL **[math]**

**XULRunner:** XULRunner is a run-time environment developed by the Mozilla Foundation to provide a common back-end for XUL-based applications. http://en.wikipedia.org/wiki/XULRunner **[math]**

# Y

**Yahoo! Pipes:** Pipes is a powerful composition tool to aggregate, manipulate, and mashup content from around the web. Like Unix pipes, simple commands can be combined together to create output that meets user needs. http://pipes.yahoo.com/pipes/ **[eco] [speech]**

# Z

**Zentralblatt MATH:** Zentralblatt MATH is a service providing reviews and abstracts for articles in pure and applied mathematics, published by Springer Science+Business Media. It is a major international reviewing service which covers the entire field of mathematics. It uses the Mathematics Subject Classification codes for organizing their reviews by topic. http://en.wikipedia.org/wiki/Zentralblatt_MATH **[math]**