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,

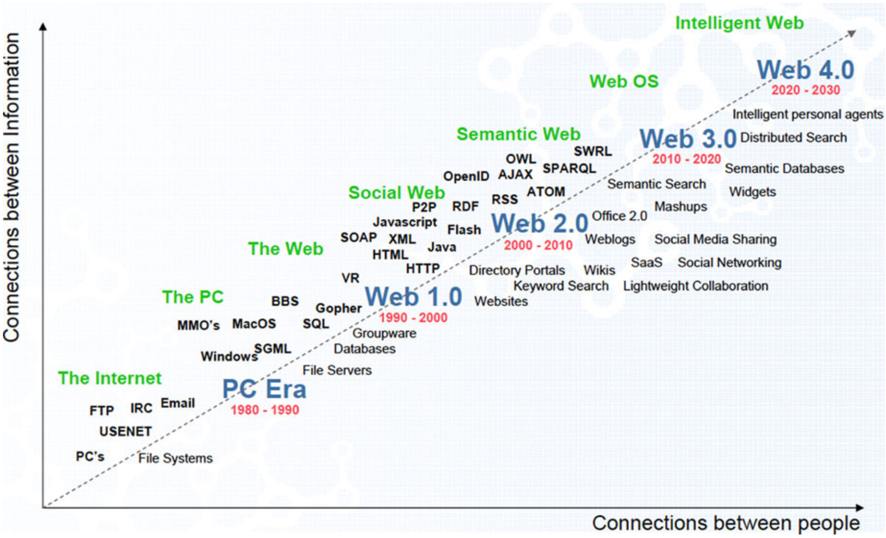


Fig. 1.1 Timeline of internet and web development (from [208]). Notice mashups coming up towards 2010

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])

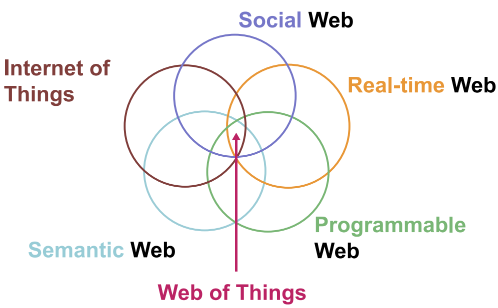
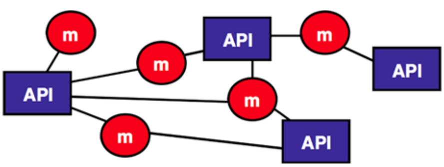


Fig. 1.3 The mashup ecosystem linking mashups and APIs. Mashups in *red circles*, APIs in *blue squares* (from [240])



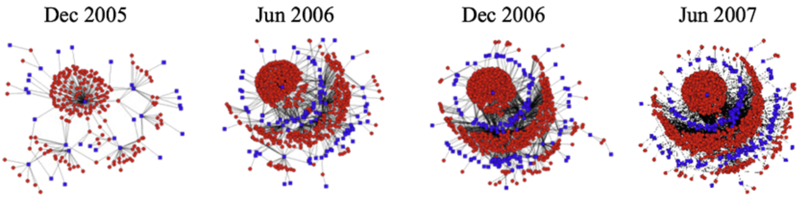


Fig. 1.4 The mashup ecosystem evolution. Mashups in *red*, APIs in *blue* (source: [240]) (Color figure online)

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.

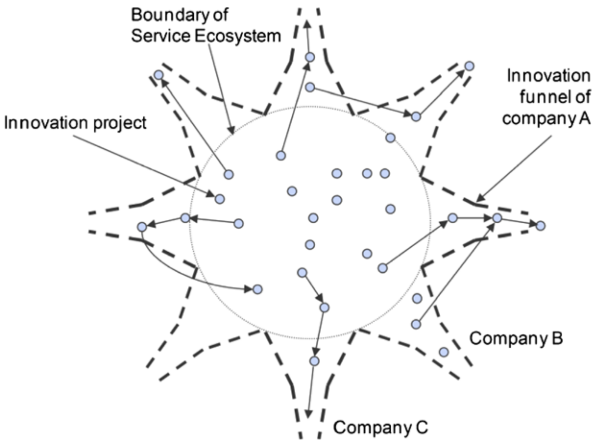


Fig. 1.5 Innovation expanding from a service ecosystem (source: [187])

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

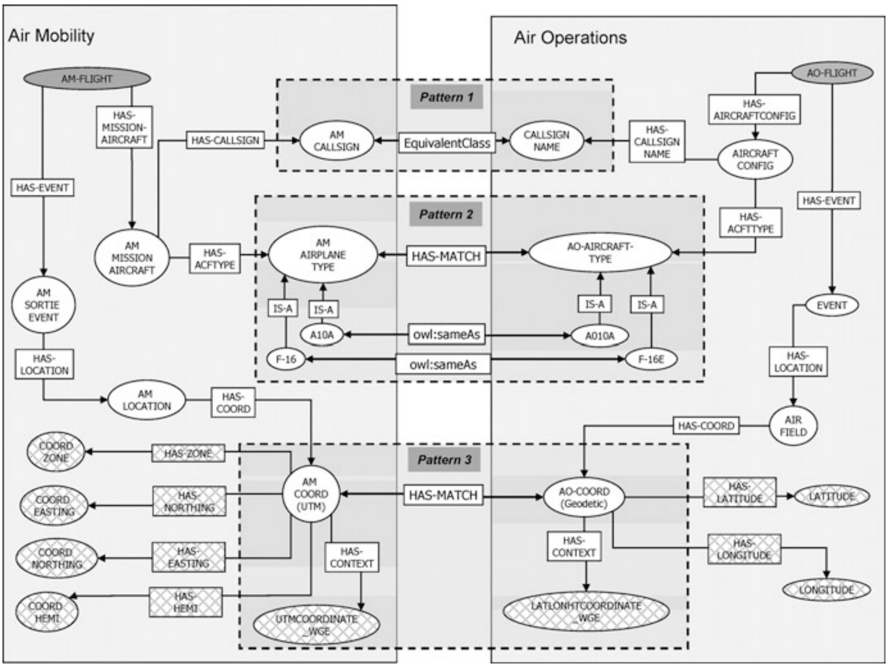


Fig. 1.6 Alignment of diverging OWL ontologies of Air Mobility and Air Operations (source: [192])

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.

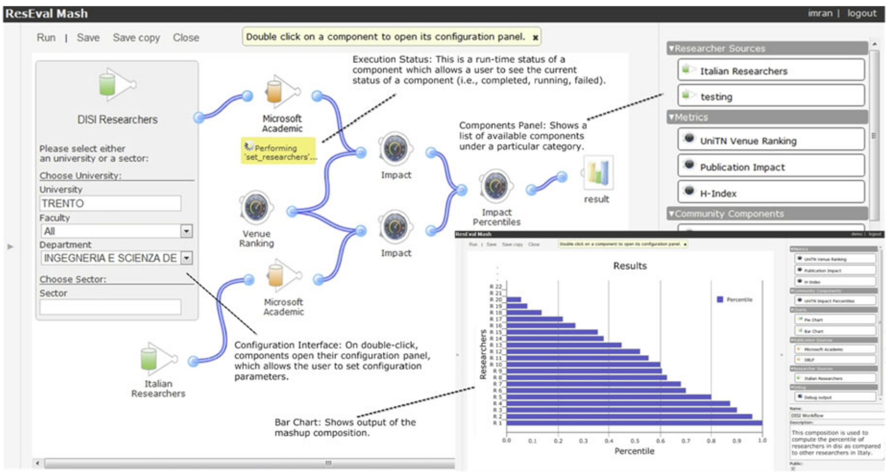


Fig. 1.7 ResEval Mash: The user interface (from [113])

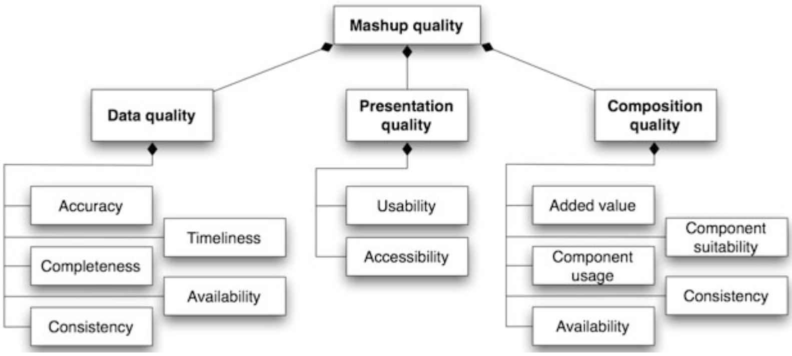


Fig. 1.8 Mashup quality model (from [38])

```
<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

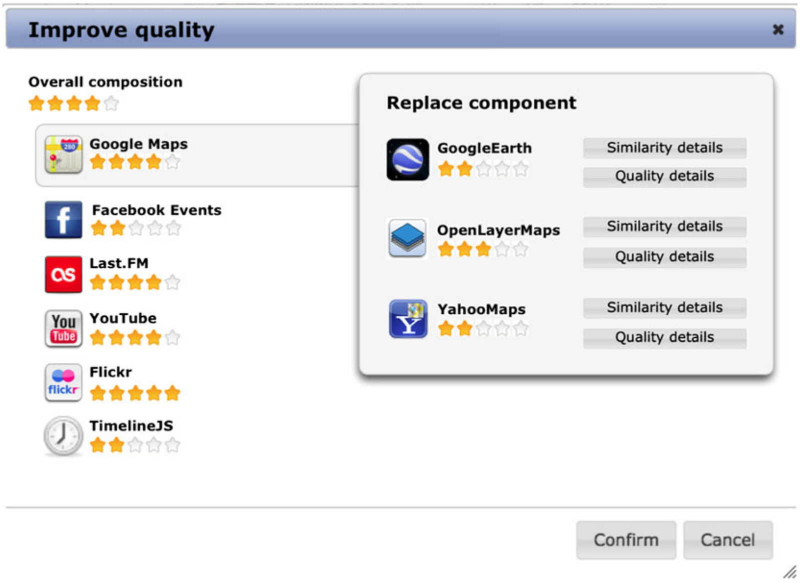


Fig. 1.10 Google Maps with alternatives on the PEUDOM user interface, the alternatives being ranked by quality (source: [38])

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/.

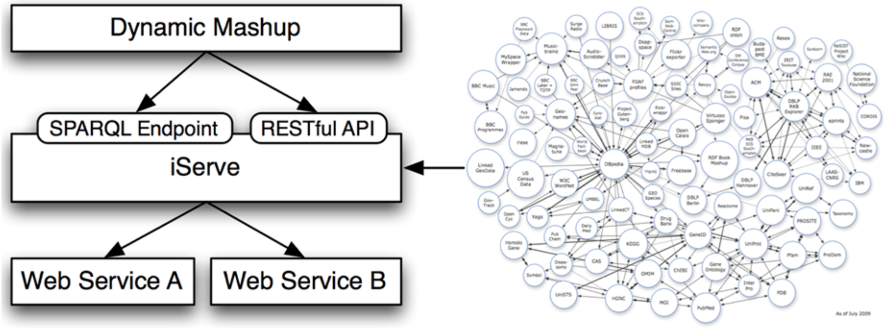


Fig. 1.11 The dynamic mashup using iServe support (source: [143])

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/.

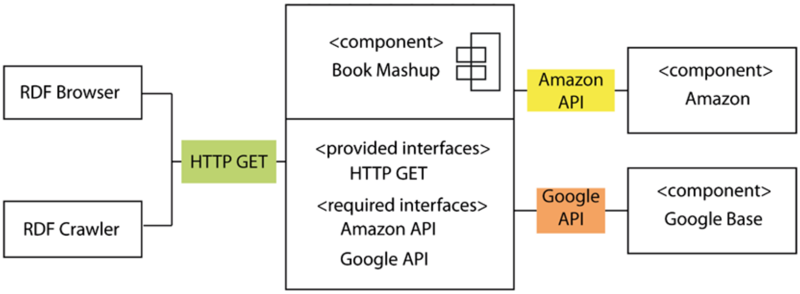


Fig. 1.12 RDF book mashup structure, remake (source: [24])

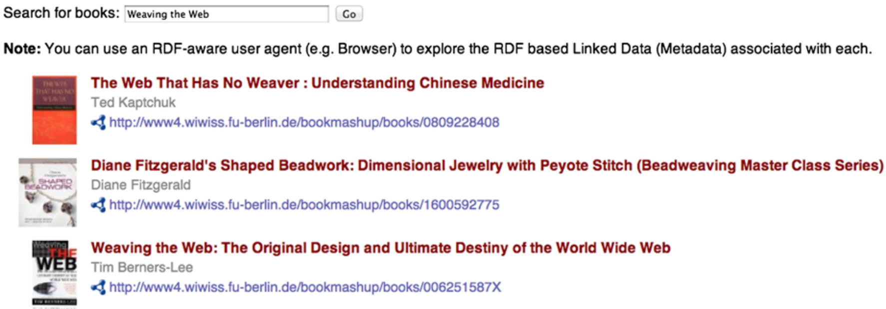


Fig. 1.13 RDF book search results

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/.

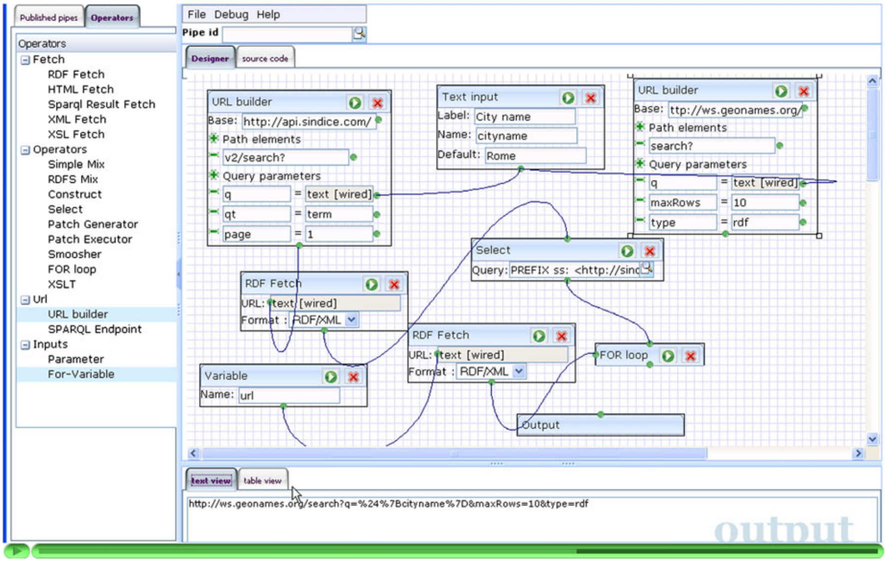


Fig. 1.14 SWS mashup development (from http://pipes.deri.org/cityfacts.html)

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

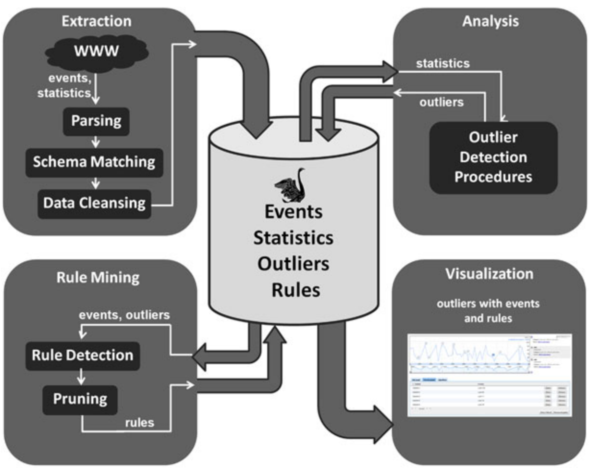


Fig. 1.15 Black Swan. The architecture overview (from http://blackswanevents.org/?page_id=179)

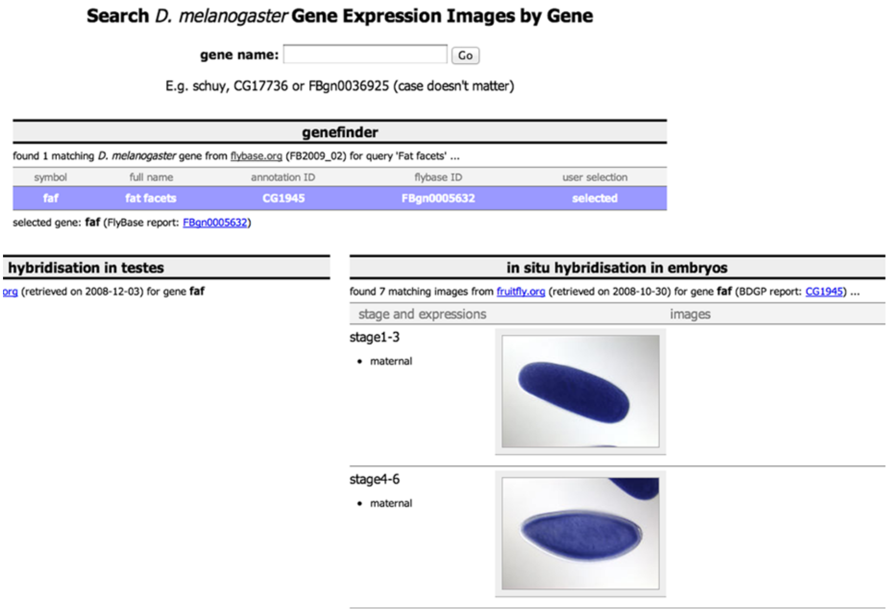


Fig. 1.16 Insitus mashup from FlyBase—abridged, many images cut off

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 non-professional 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.

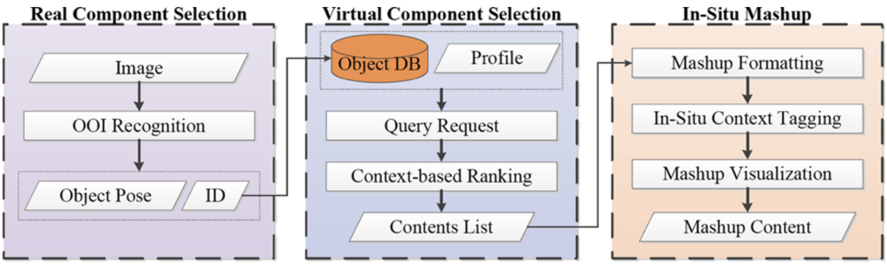


Fig. 1.17 Scheme of mashup with real and virtual component (from [238])

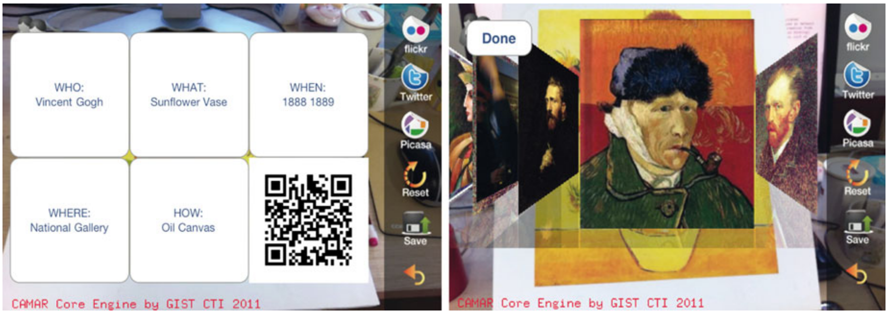


Fig. 1.18 Bar code identification of the object of interest and augmented reality output (from [238])

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].

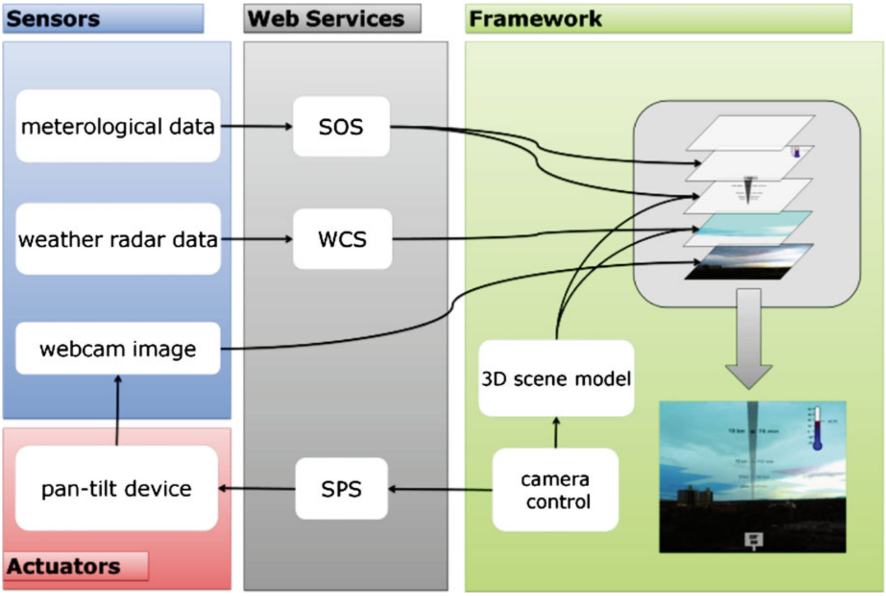


Fig. 1.19 Weather webcam: Standardized sensor data added layer-wise onto the webcam image (from [82])

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.

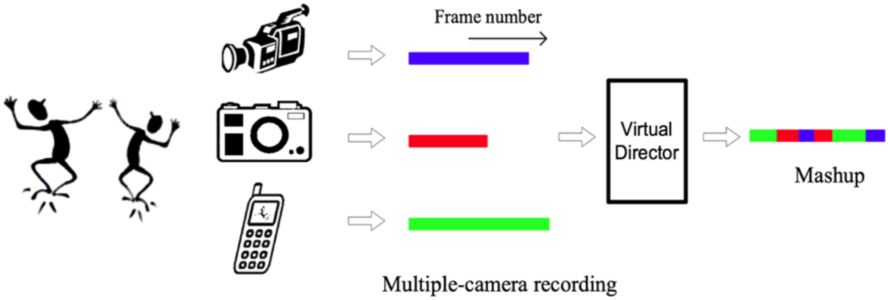


Fig. 1.20 The Virtual Director integrates multiple camera recordings into a video mashup (from [203])

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/.

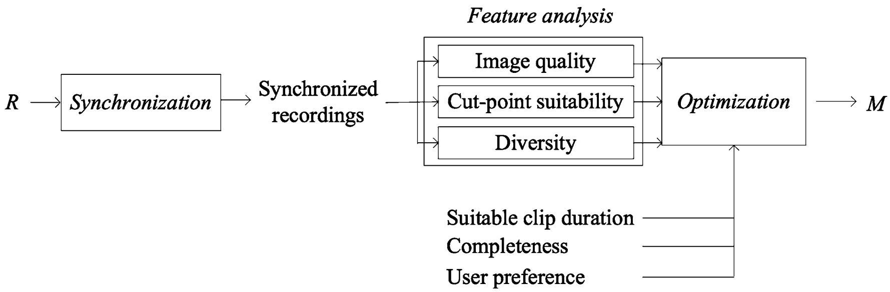


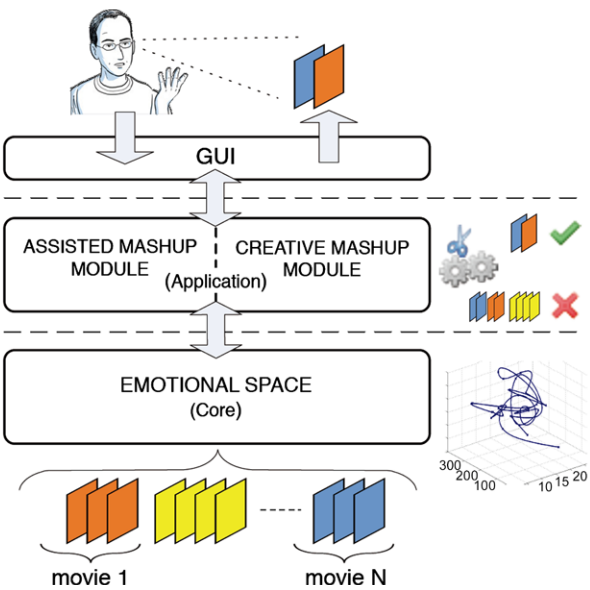
Fig. 1.21 Virtual Director overview of the mashup process. R: recording, M: mashup (from [203])

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.

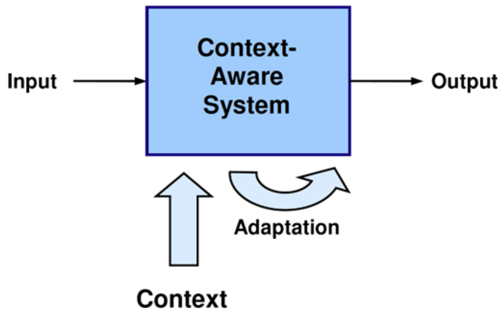


Fig. 1.23 Context-aware system with active adaptation (source: [254])

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.

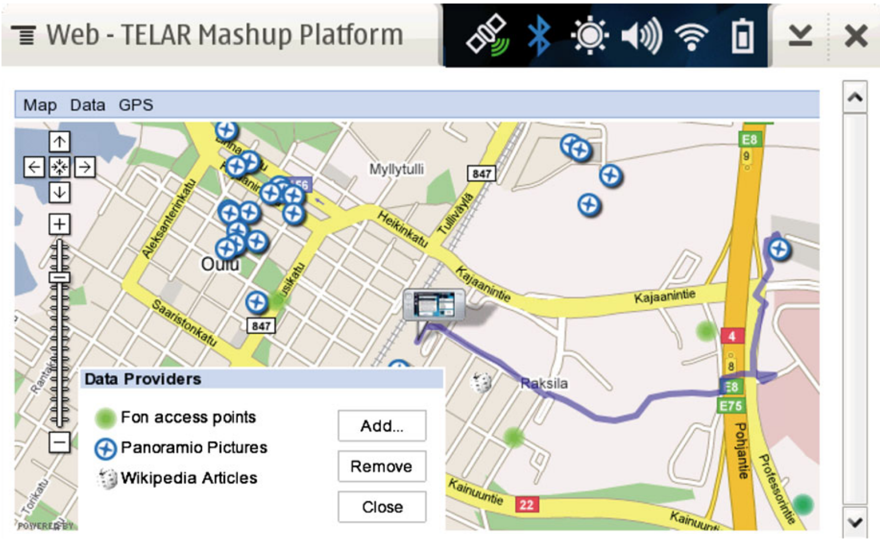


Fig. 1.24 Mobile TELAR mashup sample (source: [30])

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.

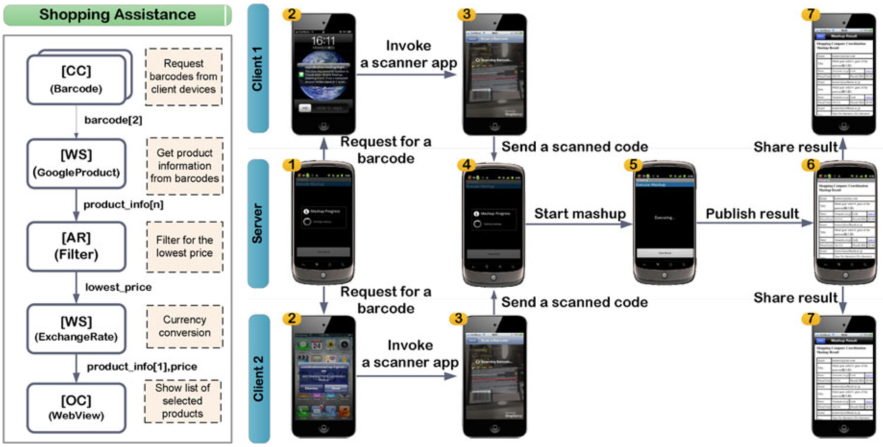


Fig. 1.25 Shopping assistance by cooperating mashups (source: [183])

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.

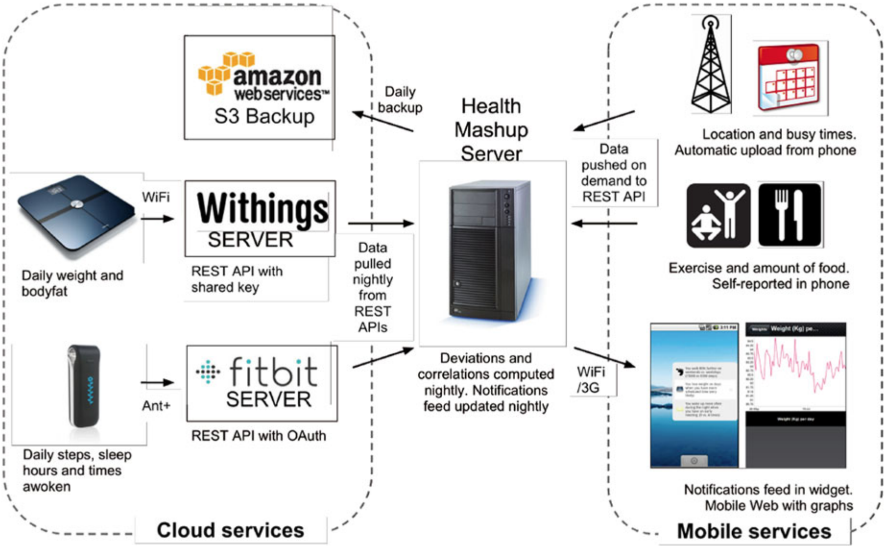


Fig. 1.27 The Personal Health Mashup server (from [213])

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.

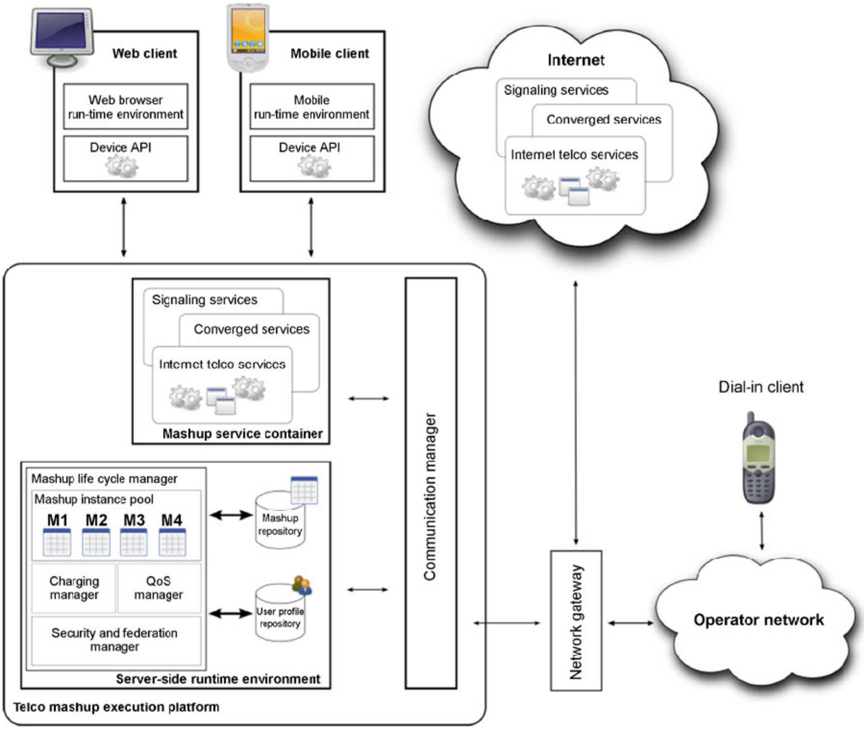


Fig. 1.29 Working environment of a telco mashup (from [55])

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

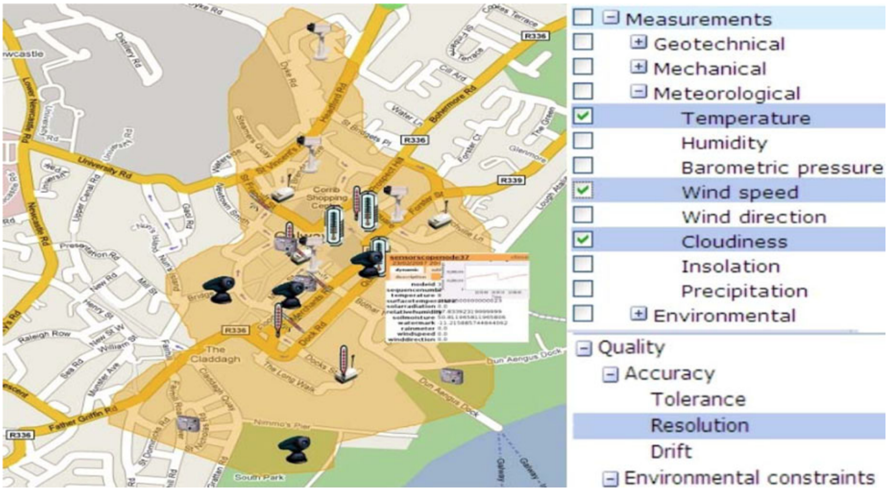


Fig. 1.30 Faceted selection of sensor feed on the SensorMasher user interface (from [135])

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.

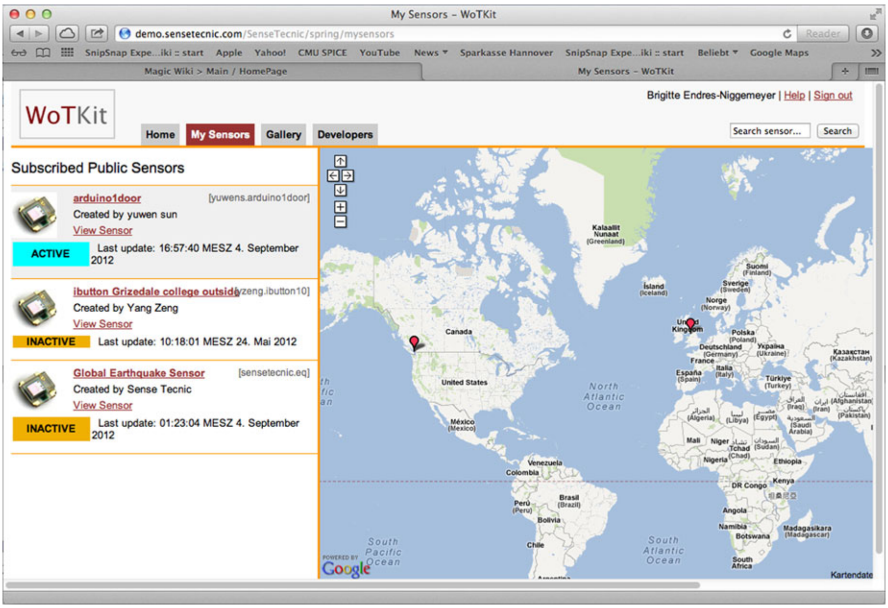


Fig. 1.31 Subscribed sensors of the author after a few minutes of testing

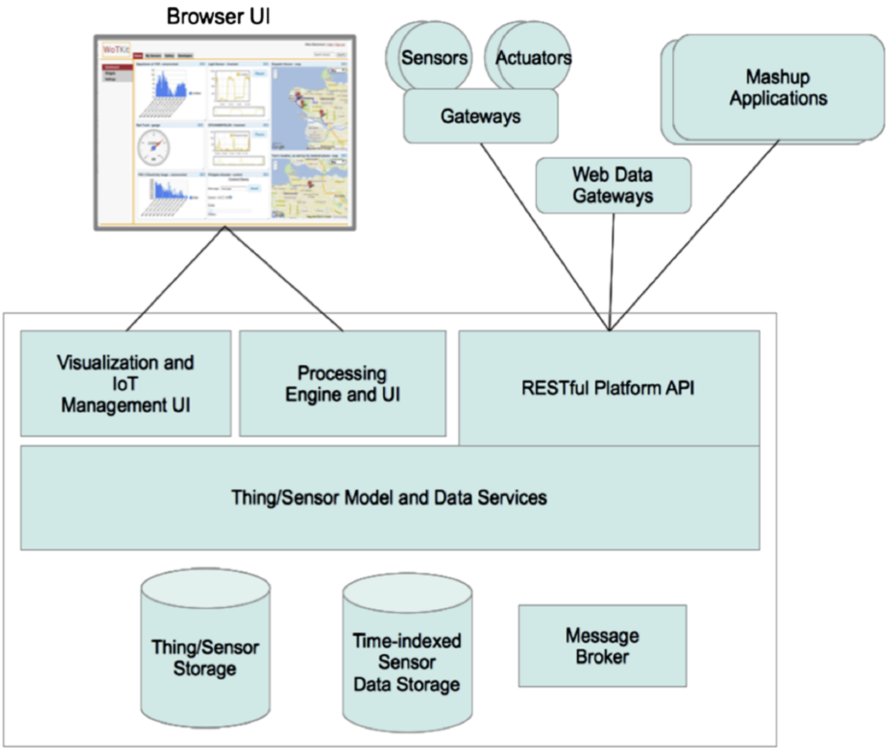


Fig. 1.32 WoTKit architecture (from [27])

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.

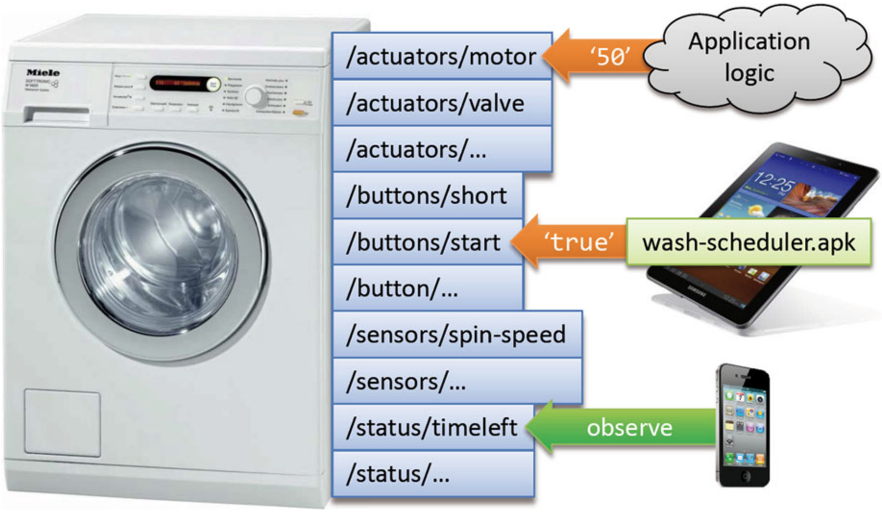


Fig. 1.33 IP-enabled washing machine (from [129])

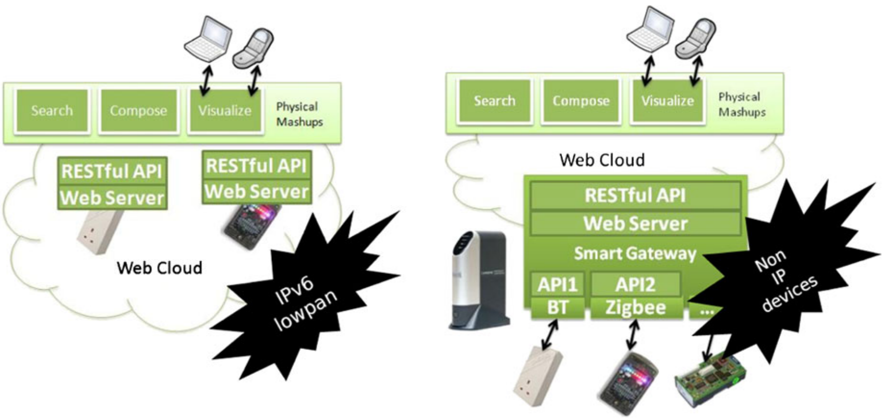


Fig. 1.34 Smart things linking to the web—directly (*left side*) or with the help of a smart gateway (*right side*) (source: [90])

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.

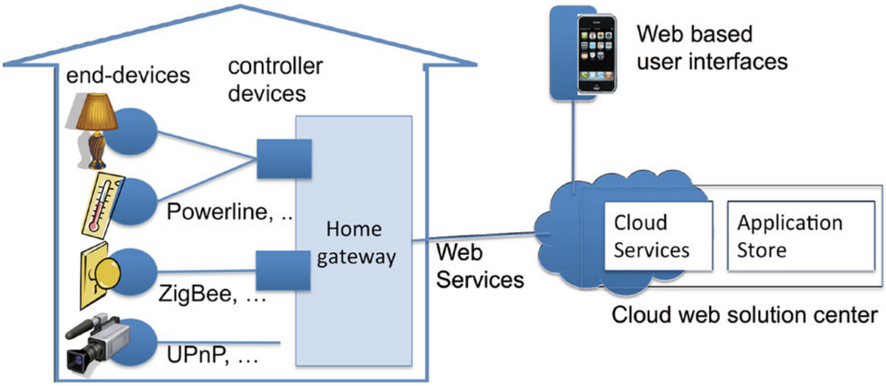


Fig. 1.35 High-level architecture for the smart home. Appliance control from a smartphone app (source: [127])

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.

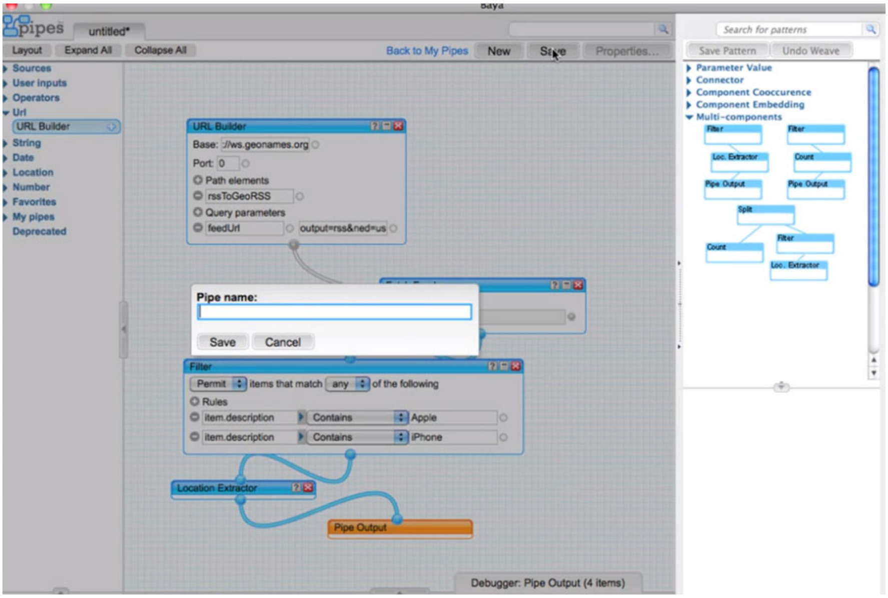


Fig. 1.37 Baya assisted mashup creation. Components on the *left* of the canvas, recommended patterns on the *right*

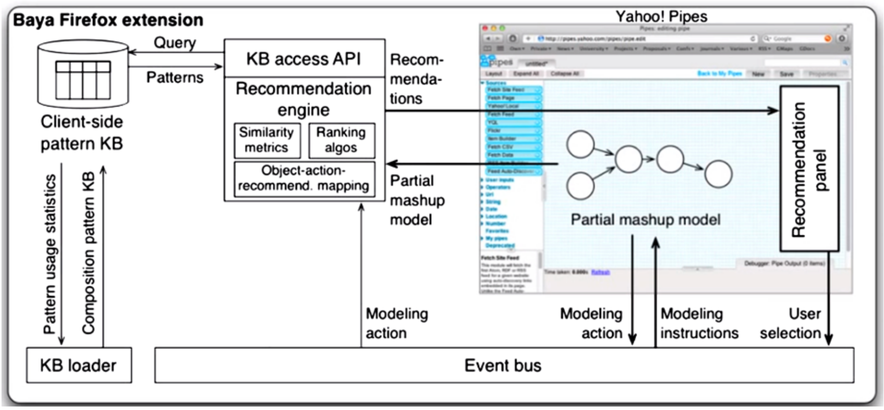


Fig. 1.38 Baya in recommendation activity

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.

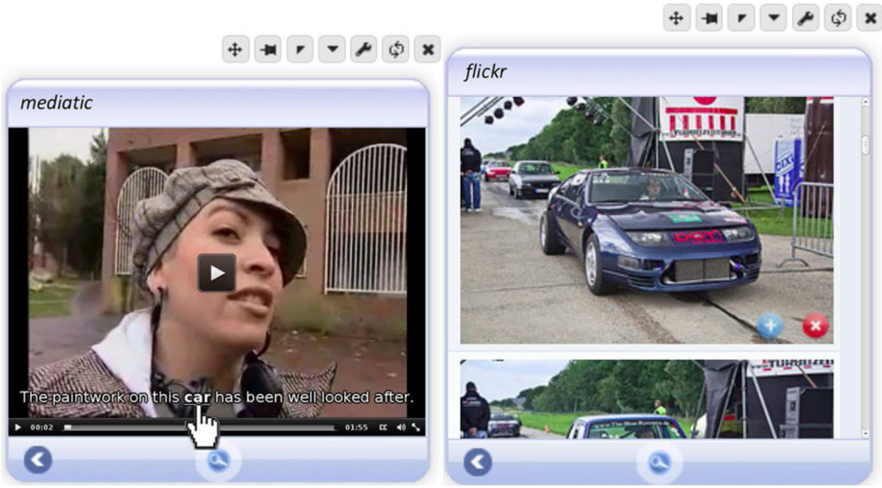


Fig. 1.39 Widget triggering neighbor widget: tapping "car" makes display cars from Flickr (source: [207])

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: AP-SCC 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: mash-up 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

http://www.springer.com/978-3-642-36402-0

Semantic Mashups
Intelligent Reuse of Web Resources
Endres-Niggemeyer, B. (Ed.)
2013, VII, 382 p., Hardcover
ISBN: 978-3-642-36402-0