Semantic-Guided Communication and Composition in a Widget/Dashboard Environment

Peter Wehner and Robert Krüger

Fink & Partner Media Services GmbH, 01309 Dresden, Germany
{peter.wehner,robert.krueger}@finkundpartner.de

Abstract. Composite websites based on widget/dashboard environments offer top grade adaptability to theoretically any task requirement or user preference. In productive use, however, the usability of a running dashboard depends on the intelligence that has been incorporated when developing the underlying software framework. This is especially true for widget/dashboard frameworks that allow for an ad hoc inter-widget-communication. The paper shows a press media software company effort to semantically enrich the inter-widget-communication of a widget/dashboard framework called NewsDesk. Employing this enrichment, NewsDesk widgets will be able to share ontology-based data and operate in activities and roles. Preliminary results of the ongoing work will be reported, featuring the enhancement of the inter-widget-communication protocol as well as an approach to offer widget composition proposals based on widget template and instance usage patterns.

Keywords: Composite Websites, Domain Knowledge, Industry Experience, Semantics, Usability, User Interface Integration.

1 Introduction

Despite the exponentially growing amount and complexity of multimedia data, current web-based access to asset collections is still characterized by a static cycle of manual keyword search and list-/chessboard-style result representation, usually showing the details of only one result item at a time. The introduction of rich client widget/dashboard environments did overcome this obstacle of static web page rendering by offering free widget compositions and thus much more task- and user-oriented user interfaces. However, this new flexibility is not for free because an ad hoc inter-widget-communication, if provided at all, has to be designed and implemented in the framework in advance. Such a way, a framework-based approach to inter-widget-communication may start from simple descriptions of the exchange of raw types like strings. However, such simple communication descriptions will not intelligently serve complex application scenarios where a dashboard composition comprises more than one widget of the same type or where the data on exchange has a record- or even a graph-based layout. The evolvement of semantic technologies seems to provide means to grasp complex application scenarios featuring more appropriate description methods. Due to well defined data retrieval endpoints, semantic

Q.Z. Sheng and J. Kjeldskov (Eds.): ICWE 2013 Workshops, LNCS 8295, pp. 16–26, 2013.

technologies also allow for a seamless integration into running widget/dashboard frameworks that aspire to instantly query an ontology-based fact database. Obviously, the support for more sophisticated application scenarios amounts to framework enhancements that include a semantic enrichment of the underlying inter-widget-communication protocol. The enhancements are supposed to enable the widget/dashboard framework to adopt semantic patterns for the composition and communication of widgets. Mostly important to the industry, this will raise the degree of usability and eventually stimulate the productivity of the end user.

The paper continues as follows. The following subsections provide an insight into NewsDesk, the widget/dashboard framework in question, as well as into SENSE, a running research project that features the semantic advancement of NewsDesk. Section 2 introduces related work by reference. Section 3 discusses the semantic enhancements of the NewsDesk-framework in detail, namely the different layers of widget-communication descriptions as well as the widget-composition automation and shows some examples. The paper concludes in section 4 with an outlook to the remaining work regarding NewsDesk and SENSE.

1.1 NewsDesk: A Widget/Dashboard-Framework

NewsDesk is a software framework for widget/dashboard environments that has been developed using pure XHTML and JavaScript over the last couple of years by fink & Partner Media Services GmbH, Germany. It aims at providing a high-flexible user interface runtime to promote usability by adaptability [1]. As a software framework, NewsDesk assists both the developer in advancing and the end user in employing the overall system. The target audience ranges from cross media journalists, browsing any type of electronic content, to newsroom teams, keeping track of the latest news of the world. To support this wide consumer range, NewsDesk scales up to multiple browser windows that may be mapped to monitor walls efficiently.

A widget, being not only the visual but also the conceptual core component of the NewsDesk-framework always adheres to a dedicated lifecycle that comprises the stages of implemented, template, instance (configured instance) and favorite, as being depicted in Fig. 1.

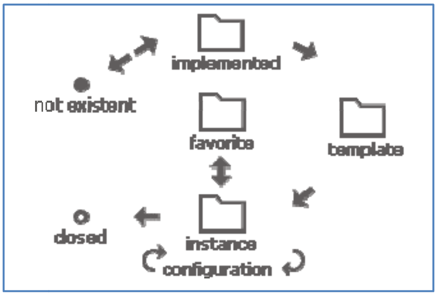
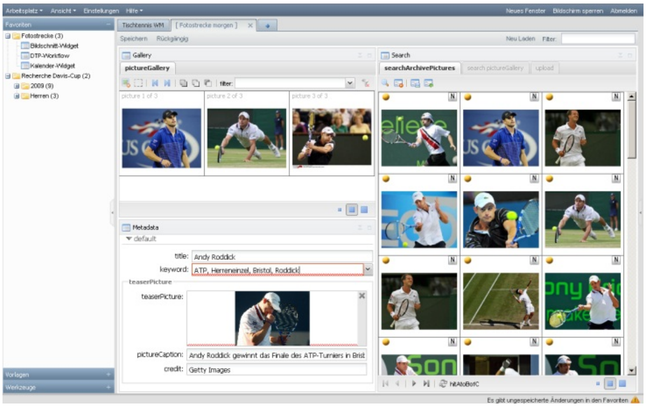


Fig. 1. Widget representation lifecycle

To exemplify, end users will usually instantiate a widget by dragging a representation from the left hand templates area to the dashboard area of NewsDesk. They may then configure the widget according to the requirements and may eventually save the configured widget for later re-instantiation as a favorite by dragging a representation from the dashboard area to another left hand favorite's area of NewsDesk. Note that a dashboard, containing widgets, as well as a workspace, containing dashboards, may also be saved down as favorites. The ongoing dashboard composition process course may in addition benefit from several positioning options such as fixed, absolute, snap-in-column and free-grid layouts. Fig. 2 shows a typical press media scenario where pictures become lined up to accompany textual content.



title: Andy Roddick
keyword: ATP, Herreneinzel, Bristol, Roddick
pictureCaption: Andy Roddick gewinnt das Finale des ATP-Turniers in Brist
credit: Getty Images

Fig. 2. A picture gallery for online publication

Besides the composition and organization of widgets, NewsDesk also offers ad hoc inter-widget-communication, which is determined to be independent from the current widget composition on a dashboard. That is, no end user side checks for compatible communication partners have to be performed during widget instantiation. Instead, any compatible communication wire will be established automatically by using some event broker architecture on provider and consumer roles of a widget comparable to Faison's approach [12]. That is, NewsDesk always tries to connect widgets with a compatible communication description. This is done by computing compatible pairs of widgets using a dedicated algorithm. This preeminent feature has so far being evolved to comprise communication checks over several channel, event and object types, including a visual "drag and drop" action.

Communication channels implement typical communication interactions in widget/dashboard environments. The first channel type models a "fire and forget" to allow a widget to just notify its neighborhood, no response is expected, and includes "drag and drop". The second channel type serves a "request / response", such that a widget may request certain information from another widget by providing some data by itself. The third channel type is equal to the second one but operates asynchronously. The fourth channel type eventually enables widgets to exchange system messages with the backend of NewsDesk.

Event types are essentially strings that are required to have a descriptive nature regarding the communication, for example “refreshView” or “publishPicture”. They serve the known “publish / subscribe” approach for event-based communication.

Object types, finally, may or may not accompany events on a channel. They have to be registered as identifiers in the scope of NewsDesk’s runtime to establish an object type vocabulary over time. The following Fig. 3 and Fig. 4 show some simplified XHTML-based provider and consumer communication descriptions for a widget.

```
…<ndw:communication>
   <ndw:provider>
      <ndw:channel type="sendChannel">
         <ndw:event type="itemSelectedWidgetEvent">
            <ndw:data type="String" />
         </ndw:event>
      </ndw:channel>
   </ndw:provider>
</ndw:communication>…
```

Fig. 3. Provider communication description

```
…<ndw:consumer>
   <ndw:channel type="sendChannel">
      <ndw:event type="itemSelectedWidgetEvent">
         <ndw:data type="String"
            callbackfunction="SendConsumer.onPing"/>
      </ndw:event>
   </ndw:channel>
</ndw:consumer>…
```

Fig. 4. Consumer communication description

1.2 NewsDesk and SENSE

The SENSE research project, where SENSE is an acronym for “Intelligent Storage and Exploration of large Document Sets”, is a joint effort of five partners from industry and science [2]. SENSE aspires to set up a continuous flow of semantically enriched data between and within different tiers of modern software architectures, focusing on widget/dashboard environments and hierarchical storage management systems. The basic idea of SENSE differs from common approaches in the semantic community by not only regarding application but also technical domain annotations to actual data. The SENSE consortium shares the opinion that furnishing modern software architectures with additional semantics from the technical domain will particularly promote the scalability of systems running large document collections over time.

SENSE features a semantic repository at the heart of its concept to mediate the different interconnected system tiers through dedicated application and technical domain ontologies. Recalling a widget/dashboard environment as just one of the considered system tiers in the project, this is where NewsDesk comes into play. In fact, SENSE facilitates another opportunity to evolve the NewsDesk-framework. Tackling terms of semantics this time, the evolvement will primarily affect the inter-widget-communication but will eventually also pay off in achieving a smarter guided widget-composition process and therefore promote NewsDesk in terms of usability.

2 Related Work

Widget/dashboard environments have been introduced to allow end users to compose personalized applications, focusing on certain functionalities provided by widgets. Essentially, Sire describes a dashboard as a place in the user interface that offers limited space to compose a set of widgets, probably underpinned by spatial arrangement tools as snap-in-columns [3]. This is in contrast to web service frontends that bundle or integrate a set of data streams by using mostly generic user interfaces [4].

The interactive widget composition did become most popular due to AJAX-based community sites like netvibes [14], script libraries like gwt that powered iGoogle [15] or productive systems like Oracle Metalink [18]. The frontend flexibility and user experience of widget/dashboard environments was such superior to known backend web portal approaches that even newer standards like JSR-286 [16] missed a wider acceptance. However, most widget-based sites or script libraries simply disregard widget-communication aspects. Most probably, this is due to the potential problems of data compatibility or even conversion. Approaches to layout configurable wires upon established widget-compositions seem to turn out into developing and running actual adapter toolkits, see IBM mashup center for example [17].

Approaches to set up a flexible communication scheme range from loosely coupled event broker architectures as introduced by Krüger [6] to complex model oriented descriptions as proposed by Pietschmann [5]. Chudnovskyy takes a more lightweight approach by analyzing and learning the interactions users do perform but does not promote automation [7]. He additionally uses semantic concepts to describe data which has been published alongside the communication channels. Paulheim explicitly employs different ontologies to separate communication aspects from facts of a certain application domain [9]. He thereby integrates whole applications at the desktop level by annotating their communication capabilities and needs. Concerning the general task of mapping facts between ontologies, Kalfogulou and Schorlemmer published a comprehensive overview of the ontology mapping methods available [10].

In parallel to enhancing the communication infrastructure semantically, even user interface interaction starts to capitalize on the benefits of semantically described data. For example, Dachselt limits the amount of data presented to the user by means like FacetZooming [8]. Hyvönen, on the other hand, enhances auto-completion tools by employing semantically different concepts with suggestion components [11].

3 Enhancements of the NewsDesk-Framework

Running NewsDesk-based applications in industry revealed a couple of extension options both at the developers and at the end users side. Concerning the communication description of object types, see section 1.1, a more advanced infrastructure to model common and complex interaction patterns was necessary. The data on exchange needs to be equipped with structure and context information and may even just reference uniform resources from the semantic web. End users, on the other hand, claim to gain a more intelligent framework support when managing the multitude of widget composition possibilities. This may include simple functionalities like temporarily locking a widget out of the dashboard communication. Another sophisticated functionality may adapt the widget selection shown in the templates area of NewsDesk according to the current dashboard composition.

Building on semantic technology, see section 1.2, most of the enhancements of the NewsDesk-framework employ existing or new ontologies, certain inference techniques and a living fact database. They improve the auto wiring algorithm for the widget instantiation process and in turn reapply the auto wiring algorithm against non-occupied communication slots to offer proposals for widget composition. The following sections 3.1 to 3.3 report:

- how domain knowledge has been introduced at the object types layer,
- how the idea of activity roles in widgets compositions has been implemented and
- how domain knowledge as well as widget activity roles become applied to offer composition proposals for an existing dashboard layout.

3.1 Communication Data Description

While using object types, see section 1.1, as filters to compute inter-widget-communication patterns has proved to be a manageable criterion, past NewsDesk installations suffered from an uncontrolled growth in more or less undocumented object type/attribute naming. That is, similar name tokens have been used with no respect to already registered vocabulary. Connecting the NewsDesk-framework to the technology stack of SENSE, the semantic repository in particular, allowed employing RDFS as a modeling platform for the data structures on exchange between widgets. Semantic technology also allowed adopting the concept of domain dependent and independent knowledge.

The design of the developed “pmedia” ontology, that will furthermore furnish the object type layer, represents domain knowledge and abstracts known concepts from existent ontologies or extends them when necessary. For the press media domain, the IPTC rNews standard, which is in parts sourced from the schema.org initiative [20], has been utilized. The XHTML-based communication description of widgets, that actually expresses the available facilities, has been extended to include the semantic descriptions in RDFa format [19]. Such a way, any requirements to provide a mapping from the system-side communication scheme of the widgets and the domain dependent “pmedia” ontology can be accomplished easily [9]. To give an example, a video widget showing a number of videos in a structured manner may describe its capability of publishing a sns:VideoObject by selection as depicted in Fig. 5.

```
…<ndw:communication xmlns:sns=“http://sense-projekt.de/″>
   <ndw:provider>
      …<ndw:event type="itemSelect">
         <ndw:data type="sns:VideoObject"/>
      </ndw:event>…
   </ndw:provider>
</ndw:communication>…
```

Fig. 5. Publishing an event along with video data

A maps widget then again, which is capable of locating any kind of sns:Place on a map, may define such a communication offer as shown in Fig. 6.

```
…<ndw:data type="sns:Place"
   callbackfunction="ND.MapsWidget.onLocation"/>…
```

Fig. 6. Consuming an event accepting place data

A sns:VideoObject that includes two predicates, namely contentLocation and regionsAllowed of rdfs:domain sns:Place, may therefore be easily connected to a maps widget, that may show all the provided locations and color them differently according to the predicates rdfs:range. That is, computing the data wise communication compatibility of two widgets can be achieved by detecting whether one widget communication description is nothing less than a sub graph of one another. Widget communication descriptions may therefore be specified in a hierarchical, more generic manner without tempting the wiring algorithm to traverse proof-by-case code. It should finally be stressed, that a continuous use of persistent uniform resource identifiers (URI) is mandatory when mapping communication structures that way.

3.2 Communication Intent Description

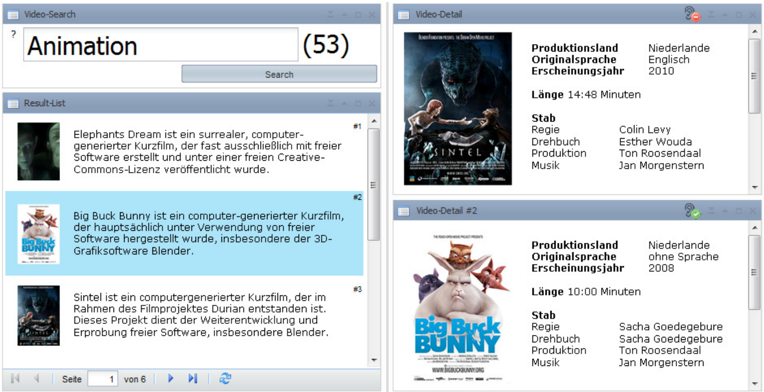
Providing the object type layer of NewsDesk with graph-based data structures added a lot of expressiveness to the communication descriptions. The next enhancement, which is an intermediate step towards composition modeling, just adapts the communication event layer to these new possibilities. The original descriptive event naming scheme, which served the “publish / subscribe” approach for event-based communication, has been upgraded to a so-called “actrole” ontology to indicate the activity role of widgets to be associated. Amongst others, typical activity roles comprise “locateObject”, “displayResultsList” or “displayResultDetail”. The “actrole” ontology in fact rather constitutes a tag library than a semantic network. However, it is determined to be independent from the application domain, press media here, and therefore from the “pmedia” ontology. Fig. 7 shows an extended communication description for an activity role of “locateObject”.

```
…<ndw:communication xmlns:sns=http://sense-projekt.de/
      xmlns:act=“http://sense-projekt.de/activity”>
   <ndw:consumer>
      …<ndw:event type="itemSelect">
         <ndw:data type="sns:VideoObject"
            activity="act:locateObject"
            callbackfunction="widget.onVideoRecieved" />
      </ndw:event>…
   </ndw:consumer>
</ndw:communication>…
```

Fig. 7. Consuming an event in a locateObject activity role

The reason for setting up a definition-oriented, domain independent ontology is the introduction of the third and last new model component, the “composition” ontology, which builds upon the before mentioned “pmedia” and “actrole” ontologies on a superior level. That is, the “composition” ontology integrates “pmedia” and “actrole” and renders the accumulated knowledge executable without mixing up domain dependencies in the communication descriptions. Moreover, the “composition” ontology does no longer model communication in a widget-aligned context but from an overall dashboard point of view. It actually strives to improve the auto wiring of widgets by providing a comprehension of the composition and communication structure patterns. Such patterns may denote a search, an exploration or a comparison scenario etc.

For example, imagine the existence of two identically typed widgets, both of them consuming a `sns:VideoObject` and both of them describing their activity role as “displayResultDetail”. This may imply that the end user most certainly tries to compare two detail views of the same object type. Since some aspects or rules of the “composition” ontology may suggest that one video detail view widget is usually sufficient on a dashboard, NewsDesk may deduce a comparison scenario. The detail view widgets may be altered to obtain an additional lock icon to interrupt the widget communication, which essentially provides a still view, see Fig. 8.



Elephants Dream ist ein surrealer, computer-generierter Kurzfilm, der fast ausschließlich mit freier Software erstellt und unter einer freien Creative-Commons-Lizenz veröffentlicht wurde.

Big Buck Bunny ist ein computer-generierter Kurzfilm, der hauptsächlich unter Verwendung von freier Software hergestellt wurde, insbesondere der 3D-Grafiksoftware Blender.

Sintel ist ein computergenerierter Kurzfilm, der im Rahmen des Filmprojektes Durian entstanden ist. Dieses Projekt dient der Weiterentwicklung und Erprobung freier Software, insbesondere Blender.

Produktionsland Niederlande
Originalsprache Englisch
Erscheinungsjahr 2010

Länge 14:48 Minuten

Stab
Regie Colin Levy
Drehbuch Esther Wouda
Produktion Ton Roosendaal
Musik Jan Morgenstern

Produktionsland Niederlande
Originalsprache ohne Sprache
Erscheinungsjahr 2008

Länge 10:00 Minuten

Stab
Regie Sacha Goedegebure
Drehbuch Sacha Goedegebure
Produktion Ton Roosendaal
Musik Jan Morgenstern

Fig. 8. Adjust user interface components

To continue the example in Fig. 8 and pinpoint another utilization of the “composition” ontology, imagine now that another search and result widgets have been added to the composition. Again keeping to one-to-one cardinalities between the different widget descriptions would rewire the communication to produce two separated chains of search, result and display or two distinct search scenarios, respectively.

The modeling of the “composition” ontology should be carried out by domain experts in dedicated RDFS resources or inference rules upon the “pmedia” and “actrole” ontologies. This also includes the specification of so-called fixed compositions that quite have some relevance in industry. They lower the notable learning curve coming along with free widget/dashboard user interaction and provide a fast time-to-market of new widget families.

3.3 Composition Proposal Generation

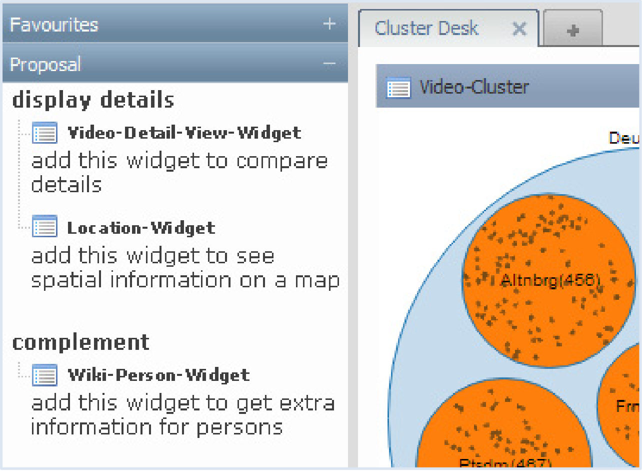
Sections 3.1 and 3.2 reported how existing communication skills of the NewsDesk-framework have been extended by modeling and employing different ontologies. This section deals with adding composition guidance features to the NewsDesk-framework that builds upon that knowledge. Composition guidance is an aspect becoming primarily relevant when the number of widgets in the templates area, see section 1.1, arrives at a critical amount. Actually, continuously raising the number of available widget templates will at some point decrease the chances of the end user to complement a dashboard composition efficiently.

A more intelligent approach to cope with larger widget template collections obviously consists of preselecting and proposing widgets that would fit in or supplement a current composition. Hence the NewsDesk-framework interface has been enhanced to provide a dedicated section in the left hand area of the workbench where generated composition proposals will be shown for the current dashboard. The proposal list will be grouped by the activity role, see section 3.2, of the offered widgets. A short guidance text based on the deduced scenario of the current composition will be added as well.

The proposal engine references the “composition” ontology and its inferior ontologies in much the same way as in the section 3.2 example. However this time, the evaluation result will be determined by not yet connected communication skills of widgets from the dashboard as well as the templates area and will also regard possible transformations of the dashboard to further composition scenarios.

It is quite exciting how this simple change in viewing the semantic repository facts apparently opened a way up from the actually aspired automation of inter-widget-communication to at least semi-automatic widget composition on a dashboard. Furthermore, with any instantiation of another widget, see the widget lifecycle in section 1.1, communication automation and composition proposals will stimulate each other and push productivity in setting up new dashboards.

To showcase the composition proposal enhancement in an example, suppose that the current dashboard just contains a venn diagram widget as a typical provider of differently grouped multimedia assets, publishing a number of `sns:VideoObject` on node selection. Some composition proposals to complement the current composition are shown in Fig. 9.



display details

Video-Detail-View-Widget
add this widget to compare details

Location-Widget
add this widget to see spatial information on a map

complement

Wiki-Person-Widget
add this widget to get extra information for persons

Fig. 9. Widget proposals for a venn widget publishing video

4 Conclusions and Outlook

The reported enhancements of the NewsDesk-framework affected both the object type and the event layer of the inter-widget-communication infrastructure. Several domain dependent and independent ontologies have been modeled to base the enhancements on semantic technologies. The framework has such a way been enabled to improve the ad hoc algorithm-based predictability of NewsDesk's inter-widget-communication due to an extended communication description of widgets as well as fact data from the semantic repository of SENSE. Example applications of the improvements include the evaluation of exchanged data graphs and the mapping of widget activity roles in a composition. The enhancements even allowed for an employment of the ad hoc wiring algorithm to not only bind communications but also visually propose widgets to complement a current composition.

However, the effort necessary to transfer the semantic annotations between the ontologies and the communication descriptions as well as the semantic repository facts and the web-browser was extensive. Consistently using persistent uniform resource identifiers with namespaces all along the way showed up to be imperative to ensure proper disambiguation of concepts.

Current work continues by evaluating the introduced enhancements from an end users point of view at remote sites of associated partners. The use case study comprises interactive exploration scenarios against large multimedia document collections, video yet, as being investigated by the SENSE-project. According to the results of the use case study, future work may take a closer look at ideas to either deduce scenario patterns from the history of user interaction or to implement cascading communication wiring where a widget gets allowed to commute itself from a consumer to a provider and vice versa.

Acknowledgements. The SENSE research project is funded by the German Federal Ministry of Education and Research (bmbf) under grant No. 01IS11025A and constitutes a part of the “KMU-Innovativ: IKT” initiative.

References

1. Wehner, P.: NewsDesk - Ein hochflexibles, Widget-basiertes Framework für Informationsportale. In: Proc. of the GeNeMe 2010 Conference - Gemeinschaften in Neuen Medien, Technische Universität Dresden (2010)
2. The SENSE project, `http://sense-projekt.de`
3. Sire, S., Paquier, M., Vagner, A., Bogaerts, J.: A Messaging API for Inter-Widgets Communication. In: ACM Proc. of the 18th International Conference on World Wide Web (2009)
4. Cáceres, M.: Widgets 1.0: Packaging and Configuration. W3C Working Draft 22 (2008)
5. Pietschmann, S., Radeck, C., Meißner, K.: Semantics-Based Discovery, Selection and Mediation for Presentation-Oriented Mashups. In: Proc. of the 5th International Workshop on Web APIs and Service Mashups, ACM ICPS (2011)
6. Krüger, R.: Kompositions- und Kommunikationsmodell für Web-Widgets. Master Thesis. Lehrstuhl für Multimediatechnik, Technische Universität Dresden (2009)
7. Chudnovskyy, O., Müller, S., Gaedke, M.: Proc. of the 4th International Workshop on Lightweight Integration on the Web, pp. 93–96 (2012)
8. Dachselt, R., Frisch, M., Weiland, M.: FacetZoom: a continuous multi-scale widget for navigating hierarchical metadata. In: Proc. of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1353–1356 (2008)
9. Paulheim, H.: Ontology-based Modularization of User Interface. In: Proc. of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems. ACM (2009)
10. Kalfogulou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. University of Southampton, The University of Edinburgh (2003)
11. Hyvönen, E., Mäkelä, E.: Semantic autocompletion. In: Mizoguchi, R., Shi, Z.-Z., Giunchiglia, F. (eds.) ASWC 2006. LNCS, vol. 4185, pp. 739–751. Springer, Heidelberg (2006)
12. Faison, T.: Event-based Programming: taking events to the limit. Apress, Berkeley (2006)
13. Hohpe, G., Woolf, B.: Enterprise Integration Patterns: Designing, Buliding and Deploying Messaging Solutions. Pearson Education, Boston (2003)
14. Netvibes/Universal Widget Technology, `http://netvibes.com`
15. Google Web Toolkit, `http://google.com/ig`, `https://developers.google.com/web-toolkit`
16. Hepper, S.: Java specification request JSR-286. Portal2.0 Documentation. SUN Inc. (2008)
17. IBM Mashup Center, `http://ibm.com/developerworks/lotus/products/mashups`
18. Oracle Metalink, `https://supporthtml.oracle.com`
19. W3C RDFa Core 1.1 - Syntax and processing rules for embedding RDF through attributes (2012), `http://www.w3.org/TR/2012/REC-rdfa-core-20120607`
20. International Press Telecommunications Council: The IPTC rNews Standard and the schema.org Initiative (2011), `http://dev.iptc.org/rNews/`, `http://schema.rdfs.org`