

# Harvesting Desktop Data for Semantic Blogging

Knud Möller and Stefan Decker

DERI, National University of Ireland, Galway  
{knud.moeller, stefan.decker}@deri.org

**Abstract.** A typical computer user’s desktop contains large amount of formal data, such as addresses, events or bibliographies. Especially within a corporate or organizational environment, it is often important to exchange this data between employees. However, state-of-the-art communication technologies such as email or bulletin boards don’t allow to easily integrate desktop data in the communication process, with the effect that the data remains locked within a user’s computer. In this paper, we propose that the recent phenomenon of blogging, combined with a tool to easily generate Semantic Web (SW) data from existing formal desktop data, can result in a form of semantic blogging which would help to overcome the aforementioned problem. We discuss a number of preconditions which must be met in order to allow semantic blogging and encourage users to author a semantic blog, and we present a prototype of the *semiBlog* editor, which was created with the purpose of user-friendly semantic blogging in mind. We argue that such a semantic blog editor should integrate tightly with a user’s desktop environment, as this would make integration of existing data into the blog as easy as possible.

## 1 Introduction

Computer users generate and collect a large amount of data on their desktop (we use the term *desktop* as a metaphor for the entire working environment within a computer). While the major part of this data usually consists of unstructured resources such as texts and images, a substantial portion of it is also structured or semi-structured data — users store their contacts in electronic address books, organize their appointments in a calendar application or manage bibliographic references in a bibliography tool. Within organizations and companies, this data is often part of the organizational knowledge, and needs to be distributed within and in between project groups and employees. The process of distribution is often complicated and cumbersome, as the data tends to be locked within each individual user’s desktop. Communication channels such as email or bulletin boards exist, but they do not normally offer ways to integrate existing formal data. Applications like MS Outlook<sup>1</sup> or MeetingMaker<sup>2</sup> offer data exchange for specific kinds of data, but they cannot be used as a general means for data exchange. Also, the exchange is restricted to users of these applications.

The recent phenomenon of weblogging (or “blogging”) [13] has been suggested as a light-weight, general purpose tool for data exchange and knowledge management within organizations [10]. With blogging moving more and more into the mainstream of web technologies, analysts such as Forrester have now taken up the topic. A report by Forrester [6] suggests blogging as a valuable publicity tool for companies and a means to keep in touch with their customer base. Furthermore, the report stresses that blogging can also serve as an inter-company communication channel, in the sense that employees could provide each other with important data about the work

<sup>1</sup> Microsoft Outlook: <http://www.microsoft.com/outlook/>

<sup>2</sup> MeetingMaker: <http://www.meetingmaker.com>

they are doing (either through their own personal blog, or through a group blog), such as project details, etc. In a similar line of thought, [12] suggests that blogging can be used as a way to perform *Knowledge Management* within an organization or company. While these observations show an interesting use case for blogging, the data that is being exchanged in this way is currently locked within the blog (or blogosphere), and disconnected from other resources a user might have on their desktop. An employee can blog about an upcoming meeting, but there is no simple way he or she could reuse existing data from their electronic calendar, addressbook, etc. In the same way, another employee can read the entry and manually add the relevant information to the various desktop applications — e.g. when and where the meeting takes place or details about people attending it — but again there is no simple way to automatically import this data.

### 1.1 Contributions of this Paper

In this paper, we suggest a form of semantic blogging to overcome the limitations posed by both the ordinary blogging approach and standard applications such as Outlook or MeetingMaker, in the sense that a user can easily add metadata about the things they blog, and other users can just as easily import this metadata and in turn incorporate it into their desktop applications. The term *semantic blogging* has largely been coined by Cayzer [2]. We argue that semantic blogging as a means of data exchange can best be implemented as a desktop-based approach, as this allows better access to and tight integration with the relevant data, a thing that is difficult or impossible in web-based approaches. Desktop applications are also capable of providing greater user-friendliness, as they (ideally) behave within parameters set by the UI paradigm of the user's desktop platform and commonly allow the use of visual techniques such as drag-and-drop. As a proof-of-concept for our approach, we have built a desktop-based blog editor called “semiBlog”, as well as an accompanying blog reader.

### 1.2 Outline of the Paper

We will begin with a short introduction to semantic blogging and blogging in general in section 2, followed by an illustration of how semantic blogging could be used as a means for the exchange of formal desktop data in section 2.1. In section 2.2 we will then discuss a number of preconditions which have to be met in order to allow this kind of semantic blogging, and to make it as user-friendly as possible. Section 3 will present our implementation of semantic blogging, the *semiBlog* application, starting with a simple use case and subsequently diving deeper into the actual design of the system. Finally, we will end the paper with a look at other approaches at semantic blogging, as well as related technologies in section 4.

## 2 Semantic Blogging

Blogging has made it possible for ordinary users to publish on the Web, and thus become content producers instead of content consumers. A number of web-based blogging platforms such as Blogger<sup>3</sup> or Movable Type<sup>4</sup> allow users to publish various kinds of data (though most blogs consist mainly of text and pictures). Blog authors manage their own content in their own blog, structure it through time in the form of discrete blog entries and are often able to categorise these entries. Usually, blogs

---

<sup>3</sup> Blogger: <http://www.blogger.com>

<sup>4</sup> Movable Type: <http://www.sixapart.com/movabletype/>

also offer so-called newsfeeds, which are essentially XML-based tables of contents. Using these newsfeeds, users can subscribe to blogs and aggregate information from different blogs. Finally, bloggers are able to comment on posts in other people's blogs or refer to them through links (trackbacks<sup>5</sup>, pingbacks<sup>6</sup>), thus creating what is called the *blogosphere*.

*Semantic blogging* extends the blogging paradigm with the possibility to add additional metadata to a blog entry. This metadata can cover a variety of aspects, ranging from information about the entry itself, such as the author or date of publication (using vocabularies such as RDF Site Summary (RSS 1.0)<sup>7</sup>), information about the structure of the blog and relations to other communication sites (e.g. by using the SIOC [1] ontology), to metadata describing the topics mentioned in the entry (a blog entry about a meeting would e.g. include metadata about that meeting, such as date and location, people attending or details of project related to the meeting, whereas an entry discussing a book would include bibliographic metadata for that book). Moreover, this metadata is expressed in a semantic format such as the Resource Description Framework (RDF), which allows further inferencing and aggregation.

## 2.1 Semantic Blogging as Data Exchange

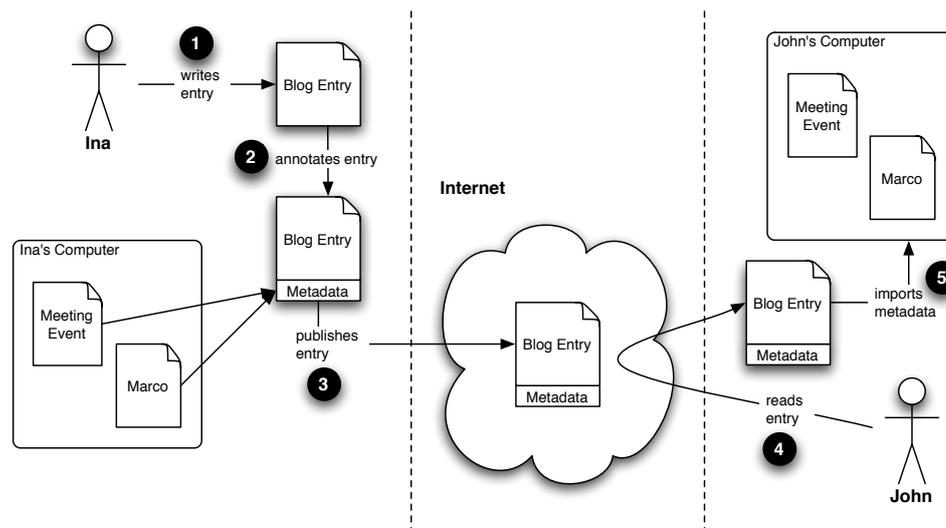


Fig. 1: Exchanging data through a blog

A specific application of semantic blogging is its use as a means of data exchange between users, e.g. employees in a company. A user would enrich a blog entry with metadata originating from some existing desktop data. Readers of the blog could then import this metadata into the appropriate applications on their desktop. Figure 1 illustrates this idea. A user named Ina wants to inform her colleagues about an upcoming project meeting. Instead of writing an email, and sending it to a specific set of people, she creates a blog entry announcing the meeting. Because she already

<sup>5</sup> Trackbacks: <http://www.movabletype.org/trackback/>

<sup>6</sup> Pingbacks: <http://www.hixie.ch/specs/pingback/pingback/>

<sup>7</sup> RSS 1.0: <http://web.resource.org/rss/1.0/>

has the relevant data (when and where) in her calendar application, she can reuse this data and attach it as metadata to the blog entry. Similarly, she might want to tell her colleagues that a visitor from another company (Marco) is going to join the meeting, and adds Marco's address book entry to the blog. In this way, semantic metadata can be made available with little or no overhead. Among other people interested in this project, an employee named John has earlier subscribed to the blog. He reads the entry and decides to go to the meeting. Instead of scribbling down the relevant data, he just imports it into his calendar and address book applications, where it is now ready to be used.

## 2.2 Preconditions for Semantic Blogging

A semantic blogging system in general, and more specifically a system that wants to offer the kind of semantic blogging functionality outlined in section 2.1, would have to meet certain preconditions, which will be discussed in the following sections.

- **Reference Mechanism** In order to reference and access desktop objects, a suitable reference mechanism has to be devised.
- **Metadata Representation Format** A suitable format has to be chosen to represent semantic metadata within the blog.
- **Usability** Since authoring a semantic blog should be just as easy as authoring an ordinary blog, a blogging platform has to be designed to maximize ease of use and user-friendliness.

**Referencing Objects on the Desktop** If we want to be able to integrate data objects from a user's desktop into a blog, we have to be able to somehow reference and access these objects. While it would be possible to import the complete metadata for some desktop object (e.g. someone's addressbook entry) once during the creation of a new blog entry, this would mean that the blog entry would only reflect the desktop object at that particular point in time. Subsequent changes to the desktop object (e.g. changing a person's phone number) would be invisible to the blog. Therefore, a simple referencing mechanism is needed, so that the blog reflects the current state of a desktop object every time it gets published. One possible solution is a URL-based reference mechanism, as URLs are well understood and supported. In principle, however, any other kind of reference mechanism can be imagined as well. It should be noted that references to desktop objects are only relevant internally as *locators* within the blog-authoring platform — blog consumers would not see these references, but instead see a formal representation of the thing that is described by the desktop object. I.e. the metadata attached to a blog entry is not about the desktop object from which it was generated, but about the thing that this desktop object describes.

Technically, the reference mechanism could be implemented in a variety of ways. Ideally, the desktop environment would already provide this. However, while research and development in the area of semantic desktops (which would provide each object on the desktop with a URL) such as the "Gnowsis" platform [11] has already come a long way, semantic desktops are still far from being mainstream, and can therefore not be assumed to be accessible to a user wishing to author a semantic blog. Another possibility are metadata-enabled file systems, such as the one that Mac OS X 10.4 ("Tiger")<sup>8</sup> offers. Here, every file object (and, by means of a work-around, even non-file objects) can have arbitrary metadata attached, which effectively means a semantic desktop "light". Finally, a solution that doesn't impose any requirements on the underlying desktop system is the definition of interfaces

<sup>8</sup> Tiger: <http://www.apple.com/macosex/>

for different kinds of data within the blogging platform itself. We have chosen this solution for our *semiBlog* application, but will look into support for Gnowsis-type systems in the future.

**Metadata Representation Format** While software like MS Outlook or MeetingMaker already allow the exchange of specific kinds of data, they use static, proprietary formats and bind a user to a specific piece of software. Within our vision of semantic blogging, however, we want to allow the user to exchange *any kind of data*, regardless of the software they use. To represent metadata in a blog, we would therefore need an open, application independent format. To accomodate any kind of data, we also need a format that allows to represent arbitrarily complex objects. Finally, the format should have a means to reference and link to other objects. Considering this list of requirements, we believe that RDF would be a good choice for a representation format. Generally, the graph structure of RDF (as opposed to the tree structure of simpler formats like XML) makes it more adept at expressing semantic relations and metadata.

```
<rdf:RDF
  <!-- ... namespace declarations,
    metadata about the blog in general, etc. ... --> >
  <rss:item rdf:about=
    "http://www.example.org/blog#YARSMeeting">
    <rss:title>YARS and Space Travel</rss:title>
    <rss:link>
      http://www.example.org/blog#YARSMeeting
    </rss:link>
    <rss:description>
      Today I had a meeting with Andreas. We went over
      his paper and talked about possibilities of using
      his YARS RDF store for a manned mission to Alpha
      Centauri. ...
    </rss:description>
    <dc:date>2005-04-06</dc:date>
    <dc:subject>
      <foaf:Person rdf:ID="andreas">
        <foaf:homepage>http://sw.bla.org/~aharth</foaf:homepage>
        <foaf:surname>Harth</foaf:surname>
        <foaf:firstName>Andreas</foaf:firstName>
        <!-- ... more properties ... -->
        <rdf:value>Andreas Harth</rdf:value>
      </foaf:Person>
    </dc:subject>
    <dc:subject>
      <bibtex:InProceedings>
        <bibtex:title>Yet Another RDF Store: Perfect Index
          Structures for Storing Semantic Web Data With
          Contexts</bibtex:title>
        <bibtex:author rdf:resource="#andreas" />
        <!-- ... more properties ... -->
        <rdf:value>YARS Paper</rdf:value>
      </bibtex:InProceedings>
    </dc:subject>
  </rss:item>
  <!-- ... more entries ... -->
</rdf:RDF>
```

Fig. 2: Example of an RSS 1.0 feed, enriched with additional metadata

Using RDF also makes the technical side of adding metadata to the blog easier. Blogs usually provide a table of contents in the form of a newsfeed. These feeds come in a variety of formats, one of which is RSS 1.0. Due to the open nature of RDF, it is possible to add arbitrary additional RDF triples to such a feed, without breaking its validity. Figure 2 shows an excerpt from a newsfeed in RSS 1.0,

using RDF/XML syntax<sup>9</sup>. Each entry in the feed (here only one) is represented by an object of type `rss:item`. Using the `dc:subject` property, additional metadata has been added (the example shows metadata about a person using the Friend of a Friend (FOAF)<sup>10</sup> vocabulary, and about a publication using a vocabulary which implements the BibTeX format).

**Usability** Authoring a semantic blog should be made just as easy as authoring an ordinary blog. While the metadata added to a blog entry could in principle be hand coded or added through specific form fields in a web-based blog editor, we believe that this would be far too complicated to appeal to a non-technical user, like the average employee we are aiming at. As noted in the article by Hendler [4], semantic metadata should be produced as a by-product of tasks that a user is already used to perform on a day-to-day basis, such as entering people in an address book application, organizing events in a calendar or managing publications in a bibliographic database. We believe that the easiest way to allow access to this data from a blogging platform is to implement it as a desktop application. Since a (good) desktop application also integrates nicely into the UI paradigms of its specific platform, this will guarantee that the user feels at home in the application. While a platform-independent solution would certainly be desirable, we believe that it will always be necessary to implement parts or the whole application in a platform-specific way, as this would allow the blog editor to interact with platform-specific data sources.

Annotating a blog entry should be as easy as dragging an address book or calendar entry onto the blog editor. To further support the user in authoring a semantic blog, a visual history of the most recently and most frequently used annotations should be kept. In this way, it will not even be necessary to open the address book or other external applications in order to annotate blog entries with frequently occurring objects. Also, if an index of all desktop objects exist (as it would be in the case of a semantic desktop like Gnowsis or a metadata-enabled file system), the user could be supported with a type-ahead functionality, which suggests matching desktop objects even while the user still types an entry. Taking the example in figure 2, the system could suggest to annotate the entry with matching metadata from the address book (or simply go ahead and do it), as soon as the user types the string “Andreas”. Effectively, this would allow a simple kind of (semi-)automatic annotation.

### 3 The semiBlog Editor

The main focus in the development of semiBlog was to make the authoring of a semantic blog as easy as possible. In particular, we wanted to allow the user to easily add semantic metadata to the blog about entities that already exist on their desktop. This led to the early design decision to make semiBlog a desktop-based application, rather than web-based. As discussed in section 2.2, access to other desktop applications and their data (e.g. through their public APIs), control of the clipboard, and techniques like drag-and-drop are difficult or impossible to implement in a web-based environment. Another design decision was to make the first prototype a native, platform-dependent application — this was done for similar reasons (access to application APIs, etc.), but also because it allowed for a much easier and quicker development process. As a result, semiBlog is currently only available for the Mac OS X operating system. Future versions of semiBlog will possibly be more platform

<sup>9</sup> RDF/XML: <http://www.w3.org/TR/rdf-syntax-grammar/>

<sup>10</sup> FOAF: <http://xmlns.com/foaf/0.1/>

independent, particularly the components comprising the semiBlog Core block and the exporter plugins in figure 4. However, we believe that for an application that wants to interface directly to existing desktop data, a certain degree of platform dependence is always necessary.

### 3.1 Example Scenario

A screenshot of an example session in semiBlog is shown in figure 3. The user has just had a meeting with his colleague Andreas, where they discussed an academic paper about one of Andreas' projects, named "YARS". He creates a new entry for this in his blog, adds some text and a picture. Then he decides to annotate the entry with some semantic metadata. To do so, he simply selects the name of his colleague, drags the corresponding card from his address book and drops it onto the text area. The entry (or rather a part of it) is now linked to a piece of data on the user's desktop. In a similar fashion he drags and drops the URL of the YARS project page from his web browser, as well as the BibTEX entry for the paper from his bibliography tool. Once a piece of text has been annotated, it is highlighted in the editor, so that each type of data is represented by a different colour. After the blog has been published, other users can read it, and — given their browser supports this — drag the annotations from the blog onto their desktop applications, and thus gain access to the enclosed metadata.

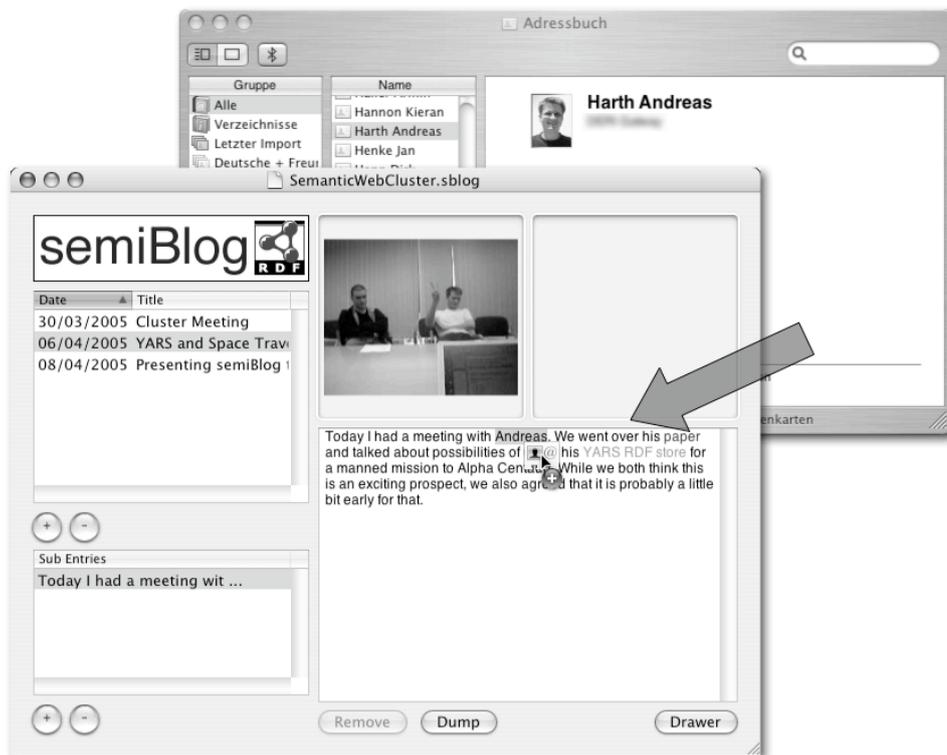


Fig. 3: Annotating a blog with semiBlog

Internally, semiBlog tracks annotations for each individual string within an entry. However, how an annotation is handled once the blog is published on the web,

depends on the publication channel. In the HTML rendering of the entry, the annotation will manifest itself as a simple HTML link to a web page (if such a page can be extracted from the desktop object). In the RDF metadata contained in the newsfeed, the annotation will apply to the entry as a whole, as there is no obvious means to make assertions about specific pieces of text in RDF.

### 3.2 Architecture and Flow of Data

Figure 4 gives an abstract overview of the architecture and flow of data in semi-Blog. The left-hand side of the figure shows some examples of desktop data that a user might have. We show address book entries, bibliographical metadata and web pages, but any other kind of data is conceivable as well. semiBlog's architecture allows custom wrapper plug-ins for each data source, which take the object information from the various desktop applications and transform it into RDF metadata form. Together with the textual entry provided by the user, this metadata is then combined in an intermediate XML representation. To generate the actual blog, the XML is transformed into the various publication channels (currently a non-semantic HTML rendering and the semantic RSS 1.0 feed, which links to the HTML). For the actual publication process, a number of exporter plugins for various kinds of blogging platforms (e.g. Blogger, MovableType, Wordpress, etc) can be chosen. The blog is now ready for publication on the web.

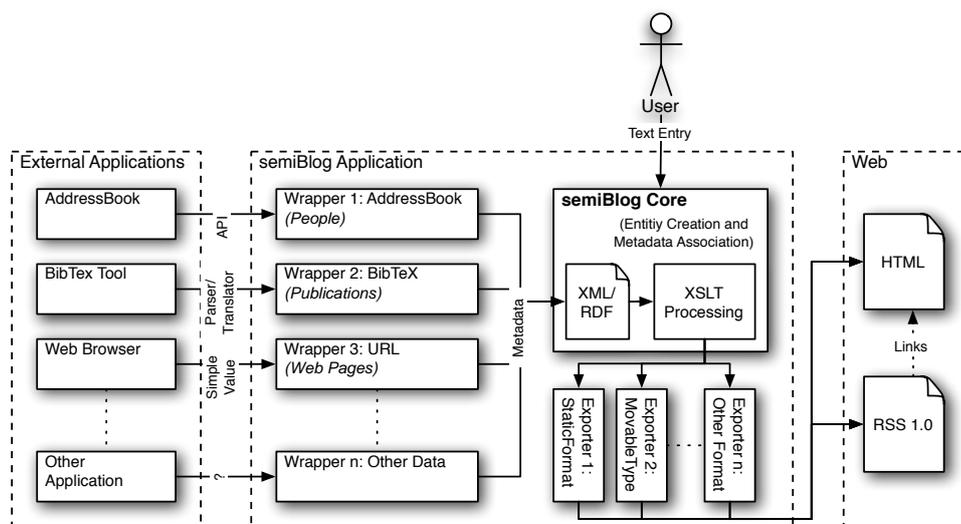


Fig. 4: semiBlog architecture and flow of data

**Wrappers** For each individual wrapper, access to the data can be handled in a different way: if the data is tied to a specific application (e.g. the AddressBook application for Mac OS X), then access via an application API might be possible. Other, more generic data (e.g. BibTeX) will be handled by appropriate parsers/translators. Wrappers are implemented as plug-ins to the platform, which makes it easy to add new functionality and cover more data sources as the need arises. Plug-ins are basically implementations of an abstract AnnotationType class, which defines the basic API each plug-in has to provide. Parts of this API are getter and setter methods for

the supported data types (objects in a drag-and-drop operation provide their data in various types or data flavors); colours for highlighting in the editor; a method `annotationFromData` to generate RDF; and a method `mainAnchorFromData` to generate a hyperlink reference for the HTML rendering of the blog. Instead of generating a static version of the desktop object in RDF during the annotation process, `semiBlog` simply stores a URL reference to the object. Each wrapper plugin defines a custom URL scheme for the kind of data it supports. Only when the blog is published at a later stage, will the URL be resolved and an RDF graph be built. This ensures that the metadata contained in the blog is always up-to-date.

**Intermediate XML** During the publication process, `semiBlog` uses XML as an intermediate data format, as shown in figure 5, regardless of the exporter plugin chosen. Each `entry` is represented by an `entry` element, the textual content is contained in a `text` element. This element allows mixed content of `text` and `annotation` elements, which provides the possibility to add metadata to individual substrings of an entry. We chose this inline annotation technique over external annotation by reference, because it makes the XSLT transformations in the next step easier to accomplish. Each annotation contains a `mainAnchor` to be used in the HTML rendering of the blog (or in any other non-semantic representation), the RDF `metadata` and the actual substring of the entry to be annotated.

```
<!-- ... -->
<text>
  Today I had a meeting with
  <annotation>
    <mainAnchor>http://sw.deri.org/~aharth</mainAnchor>
    <metadata>
      <foaf:Person>
        <foaf:name>Andreas Harth</foaf:name>
        <!-- ... other properties -->
      </foaf:Person>
    </metadata>
    <content>Andreas</content>
  </annotation>.
  We went over his
  <annotation>
    <mainAnchor>
      http://sw.deri.org/2004/06/yars/doc/summary
    </mainAnchor>
    <metadata>
      <bibtex:InProceedings>
        <bibtex:title>
          Yet Another RDF Store:...
        </bibtex:title>
        <!-- ... other properties -->
      </bibtex:InProceedings>
    </metadata>
    <content>paper</content>
  </annotation>
  and talked about possibilities of using his YARS RDF
  store for a manned mission to Alpha Centauri. ...
</text>
<!-- ... -->
```

Fig. 5: Intermediate XML

**XSLT Transformation** Once the user's textual entry and the semantic metadata have been combined into the intermediate XML format, we can use XSLT to transform it into arbitrary publication channels. This is a rather straight-forward process: for each exporter module, there is one XSLT stylesheet per channel, and

each one picks those elements out of the XML which are appropriate for the corresponding publication channel. For example, the XML ->HTML stylesheet will ignore the `metadata` element and instead pick the `content` and `mainAnchor` tags to produce a hyperlink for specific substrings of the entry. The XML ->RSS stylesheet will only look at the `metadata` element and annotate the entry as a whole. Once this is done, the blog can be uploaded to a server (again depending on the current exporter module).

**Extracting Metadata** While the functionality to extract metadata from the blog could in principal be added as a plugin to existing blog readers, we chose to build a stand-alone application for demonstration purposes. A user can load a blog's newsfeed and browse its entries. For each entry, the user is presented with a list of annotations, from which he can choose to drag individual items onto his own address book, calendar or other application (see Fig. 6). Technically, the blog reader uses the same plugins as the semiBlog editor, which contain functionality to transform data into both directions (i.e. from the source format into RDF and back). Each plugin defines a pair consisting of an `rdf:type` and a target application, so that the right plugin can be chosen depending on the data dragged and the target of the drag operation.



Fig. 6: A metadata-aware blog reader

## 4 Related Work

The concept of Semantic Blogging is not our invention; a number of recent papers have investigated the topic from different angles. [5] discusses a semantic blogging prototype built on top of the SW browser Haystack [9]. They interpret blog entries mainly as annotations of other blog entries and web resources in general, and devise a platform to realise this in terms of the SW. We extend the scope of this

interpretation and include the possibility of annotating resources that originally do not exist on the Web, but only on a user's desktop. The paper also underlines the inherent semantic structure of blogs and their entries as such, and presents a way of formalising these semantics in a SW fashion. This point is also made (on a more general level, encompassing various kinds of web based information exchange such as e-mail, bulletin boards, etc.) by [1] and [8]. [3] puts a strong emphasis on the use of semantic technologies to enhance the possibilities of blog consumption, by allowing viewing, navigation and querying with respect to semantics. The paper describes a prototype for both creation and browsing of semantic blogs, which was developed as part of the SWAD-E project . While the prototype only deals with bibliographic metadata as annotations to blog entries, the authors point out that the same technologies can be used for any kind of metadata. [7] describes a platform called Semblog, which uses the FOAF ontology as an integral part. FOAF descriptions of blog authors are linked to their blogs. In this way, the blog as a whole is annotated with metadata about its author. On a more fine-grained level individual blog entries are classified by linking them to a personalised ontology. To implement their platform, the authors provide both a Perl CGI-based tool called RNA and a standalone Windows-based tool called Glucose.

As already mentioned, software like MS Outlook and MeetingMaker allow the transfer of formal data over the web. However, these applications are restricted to specific kinds of data, while our semantic blogging approach aims at covering all kinds of data. Also, the systems typically use proprietary, closed formats, which are aimed specifically at the kind of data they support, while an approach that tries to cover a very broad range of possible kinds of data would have to be open and domain-independent.

Semantic desktop efforts such as Gnowsis [11] assign URLs to every object on a user's desktop, thus allowing to treat them as nodes in an RDF graph and make assertions about them. Also, Gnowsis contains interfaces to extract RDF from various kinds of desktop objects. This approach would offer the perfect infrastructural basis for our semantic blogging efforts, as it would allow us to make our application slimmer and focus on usability aspects, instead of having to provide functionality for referencing and accessing desktop objects ourselves.

Finally, meta-data enabled file systems like the one implemented in Mac OS X 10.4 ("Tiger"), BeFS or WinFS<sup>11</sup> could be viewed as a lighter version of semantic desktops and could play a similar role as an infrastructural basis. Arbitrary meta-data can be added to each file object. By providing a hidden file representation for non-file objects such as entries in the AddressBook application, also non-file objects can benefit from the same functionality. Moreover, since all metadata events are handled at kernel level, changes get noted instantly, and there is little to no chance that changes remain unnoticed, thus compromising the semantic stability of the metadata system.

## 5 Future Work

Although the full chain of steps involved in publishing a semantic blog – writing a textual entry, adding pictures, annotating the entry with semantic metadata and transforming the internal data format into the actual components of the blog – is implemented and can be performed within semiBlog, the software is still very much a prototype and can be improved in many areas. Since the main focus was on the interaction between semiBlog and other desktop applications, we have so far neglected aspects such as layout capabilities. As a result, the user is currently bound to a static layout scheme for the blog. In future versions, we will have to address

<sup>11</sup> WinFS: <http://msdn.microsoft.com/data/WinFS/>

this and allow a freer layout, perhaps also offering to the user to choose between different Cascading Style Sheets (CSS) templates. Also, it is currently not possible to add metadata other than that provided by the wrapper plug-ins. Users should have the option to manually annotate entries with arbitrary metadata, if they wish to do so (however, since we were interested in easy drag-and-drop based annotation, this was of no concern to us for the prototype). Also, by using an ontology like SIOC [1], more semantic metadata could be generated easily by making use of the internal and external semantic structure of the blog.

We will also look into adding functionality for automatic or semi-automatic annotation, since even simple and easy manual annotation techniques like drag-and-drop might still be a barrier for many users.

Future versions of the editor will allow integration with generic Semantic Desktop solutions like Gnowsis, or perhaps with metadata-enabled file systems. This would eliminate the need for application specific data wrappers as they are currently used in semiBlog.

## **6 Conclusion**

In this paper, we have argued that semantic blogging can be used as a universal means to perform data exchange, e.g. in companies and organizations (but in no way restricted to these environments). This kind of communication helps to keep up the flow of information in an organization, and can make the work of individual members easier. We have shown that other technologies fail to perform this task. There are certain preconditions that must be met in order to enable semantic blogging in the sense discussed in this paper: we have highlighted the need for a suitable referencing and access mechanism, as well as a suitable metadata exchange format and stressed the importance of usability aspects, such as tight desktop integration and visual UI techniques such as drag-and-drop, to make semantic blogging appealing to non-technical users. Finally, we have presented our semiBlog application, which tries to implement these requirements.

## **7 Acknowledgement**

This material is based upon works supported by the Science Foundation Ireland under Grant No. SFI/02/CE1/I131.

## Bibliography

- [1] J. G. Breslin, A. Harth, U. Bojars, and S. Decker. Towards Semantically-Interlinked Online Communities. In *The 2nd European Semantic Web Conference (ESWC '05), Heraklion, Greece, Proceedings, LNCS 3532*, pages 500–514, May 2005.
- [2] S. Cayzer. Semantic Blogging and Decentralized Knowledge Management. *Communications of the ACM*, 47(12):47–52, December 2004.
- [3] S. Cayzer. Semantic Blogging: Spreading the Semantic Web Meme. In *XML Europe 2004, Amsterdam, Netherlands, Proceedings*, April 2004.
- [4] J. Hendler. Agents and the Semantic Web. *IEEE Intelligent Systems*, 16(2):30–37, March/April 2001.
- [5] D. R. Karger and D. Quan. What Would It Mean to Blog on the Semantic Web? In *Third International Semantic Web Conference (ISWC2004), Hiroshima, Japan, Proceedings*, pages 214–228. Springer, November 2004.
- [6] C. Li. Blogging: Bubble Or Big Deal? Forrester Research, Inc., November 5 2004.
- [7] I. Ohmukai and H. Takeda. Semblog: Personal Publishing Platform with RSS and FOAF. In *1st Workshop on Friend of a Friend, Social Networking and the (Semantic) Web, Proceedings*, pages 217–221, Galway, September 2004.
- [8] D. Quan, K. Bakshi, and D. R. Karger. A Unified Abstraction for Messaging on the Semantic Web. In *The Twelfth International World Wide Web Conference (WWW2003), Budapest, Hungary, Proceedings*, May 2003.
- [9] D. Quan, D. Huynh, and D. R. Karger. Haystack: a Platform for Authoring End User Semantic Web Applications. In *Second International Semantic Web Conference (ISWC2003), Proceedings*, 2003.
- [10] M. Röhl. Business Weblogs — A Pragmatic Approach to Introducing Weblogs in Medium and Large Enterprises. In *BlogTalk - A European Conference on Weblogs, Vienna, Austria*, May 2003.
- [11] L. Sauermann. The Gnowsis — Using Semantic Web Technologies to build a Semantic Desktop. Master’s thesis, Technische Universität Wien, December 2003.
- [12] B. Stone. *Blogging - Genius Strategies for Instant Web Content*. New Riders, 2002.
- [13] J. Walker. Weblog. In D. Herman, M. Jahn, and M.-L. Ryan, editors, *Routledge Encyclopedia of Narrative Theory*, page 45. Routledge, London and New York, 2005.