SemClip - Overcoming the Semantic Gap Between Desktop Applications

Gerald Reif[1], Gian Marco Laube[1], Knud Möller[2], and Harald Gall[1]

[1] University of Zurich, Department of Informatics, Zurich, Switzerland
{reif,gall}@ifi.uzh.ch
[2] Digital Enterprise Research Institute, National University of Ireland, Galway
knud.moeller@deri.org

Abstract. When copying and pasting data between applications using the operating system clipboard, the semantics of the transfered information is usually lost. Using Semantic Web technologies these semantics can be explicitly defined in a machine process-able way and therefore be preserved during the data transfer. In this paper we introduce SemClip, our implementation of a Semantic Clipboard that enables the exchange of semantically enriched data between desktop applications and show how such a clipboard can be used to copy and paste semantic annotations from Web pages to desktop applications.

1 Introduction

The concept of a clipboard that has been introduced to modern operating systems enables the data transfer using the notions of "copy" and "paste". In our everyday work the clipboard is frequently used to exchange data between desktop applications. Current clipboards support the exchange of plain text, formatted text (*e.g.,* HTML, rich text format RTF), or binary data (*e.g.,* images). The semantic context of the exchanged data, however, is lost during the data transfer. Using such a clipboard therefore leaves a semantic gap between the source and the target application. Some clipboards also support the transfer of semantically richer data, for example between the applications within an office suite. This data transfer, however, relies on proprietary data formats.

The Semantic Web, on the other hand, *"provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries."* [7] This definition from the W3C homepage perfectly describes the functionality of a clipboard and indicates that the use of Semantic Web technologies can help to overcome the semantic gap.

In this paper we present SemClip [6, 3], our implemantation of a *Semantic Clipboard* [1] that enables the exchange of semantically enriched data between desktop applications. In particular, we demonstrate how SemClip can be used to copy and paste semantic RDF annotations from Web pages to desktop applications. SemClip uses ontology mediation if the target application is not able to handle the ontology used by the RDF graph pasted into the clipboard. In

addition, SemClip features a graph completion service that aims to add information that further describes the resources to be pasted. The implementation is based on Web standards, such as HTTP and Web Services. The integration into the operating system environment has been done for the Mac OS X operating system.

In this paper Section 2 discusses the insufficiencies of current clipboards based on everyday scenarios and highlight how the use of Semantic Web technologies offers a solution to this problem. Section 3 introduces the SemClip architecture and Section 4 details on the implementation. In Section 5 we discuss related work. Finally, Section 6 explores the lessons learned and gives an outlook on future work.

2   Scenarios

In this section we use an everyday scenario to illustrate the semantic gap that arises from using the operating system clipboard and highlight how a Semantic Clipboard, such as SemClip, offers a solution to this problem. For example, a user orders a cinema ticket in an online-store and is presented a Web page with the ticket receipt. Adding the event to the calendar or viewing the location of the cinema on a map cannot be done in one step when using the operating system clipboard. Instead, the user has to select each information item in an individual step (*e.g.,* the movie title as event name, the street name, *etc.*) and paste it to the according field of the target application. This way the user is responsible for manually restoring the semantic context of the data.

Using a clipboard that is based on Semantic Web technologies enables the transfer from RDF meta-data between applications. As such, the semantics of the data can be preserved during the data transfer. In the scenario described above, the clipboard can now paste the performance data in RDF format from the cinema's online-store receipt page to a desktop application. The desktop application interprets the well-defined semantics and performs the appropriate actions (e.g., creating a new entry for the event in the calendar, viewing the address on a map) without having the user to restore the semantic context of the data. The implications of the pasted data, however, depend on the target application. The target application processes the data received and handles it based on its semantic context. Therefore, pasting the same data to different target applications can cause different semantic implications. For example, pasting address data to a route planner and an address book will in one case result in displaying the address on a map and in the other case in creating a new address book card.

The application area of a Semantic Clipboard is not limited to semantically annotates Web pages. The source application can be any desktop application. For example a user copies a contact from his address book to his text processor which results in opening an empty template letter with the address already filled in.
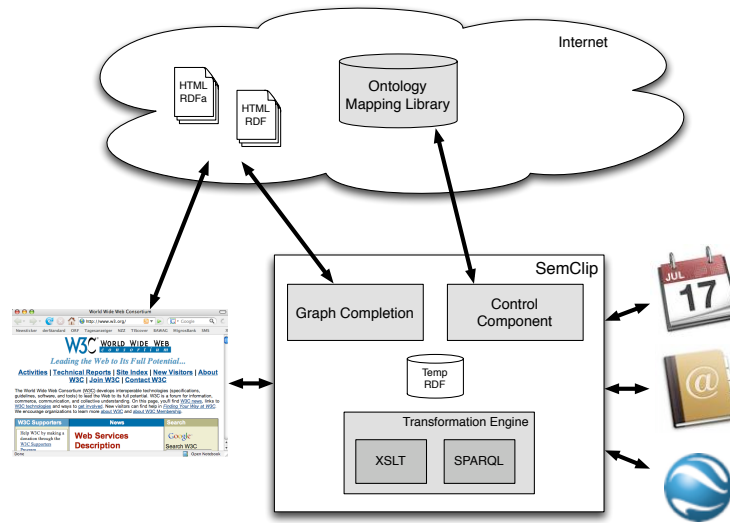
Fig. 1. Architecture of the Semantic Clipboard (SemClip).

3 Architecture

In this section we present the architecture of SemClip and introduce step by step the services involved. The architecture is shown in Figure 1. When a user pastes data from a desktop application, such as a Web browser, the data is stored in the temporary RDF storage of SemClip (*Temp RDF*). The *Control Component* takes now over the coordination of the following processing steps.

Over the years, the Semantic Web community defined numerous ontologies for various domains. In many cases more than one ontology can be found for similar or overlapping domains. For example the FOAF[3], vCard[4], and SWRC[5] ontologies can be used to represent a person's contact data. If the target application can only process data using one of these ontologies, it cannot handle data represented in the other ontologies although the application is capable to process data in this domain. To overcome this semantic gap SemClip uses the *Transformation Engine* and the *Ontology Mapping Library* to do ontology mediation. As first step of the ontology mediation we analyze which ontologies are used in the RDF graph to be copied. For each ontology used we look up in the Ontology Mapping Library to which other ontologies a mapping exists and store it in a hashtable with the ontology identifier (ID) as key and the reference to the mapping rules as value.

---

[3] `http://www.foaf-project.org`

[4] `http://www.w3.org/2006/vcard/ns`

[5] `http://ontoware.org/projects/swrc/`

After a user selected the target application and issued the paste action, a negotiation process is started. The application sends a request to SemClip using the ontology ID to indicate the preferred ontology. SemClip looks up the ontology ID in its hashtable. If the ontology ID was found, we have to consider two possibilities: (1) The requested ontology was originally used in the copied RDF graph and the graph can be handed over directly to the requesting target application. (2) The lookup in the hashtable was successful since there is a mapping in the Ontology Mapping Library that can be used to transform the copied RDF graph into the requested ontology. In this case the Transformation Engine translates the RDF graph into the requested ontology and the translated graph is returned to the target application. If the requested ontology ID was not found in the hastable, the requesting application is informed that no such data is available. If the application is able to handle also other ontologies it can issue a new request with a different ontology ID. SemClip also provides a method that returns a list of all ontology IDs currently available in SemClip and the target application chooses the preferred one before issuing the actual data request.

As described above, when an mismatch exists between the ontologies used in the pasted RDF graph and the requested ontology by the target application, the Transformation Engine is used to resolve this mismatch. The Control Component is responsible for issuing the download of the required mapping rules from the Ontology Mapping Library (if not cached locally) and for triggering the transformation process. The Transformation Engine supports two kinds of transformations: SPARQL and XSLT. For RDF-to-RDF graph transformations we use SPARQL construct queries. For RDF to XML or plain text transformations we use XSLT style sheets (*e.g.,* if the target application requests plain-text vCard as defined in rfc2426[6]).

The semantic meta-data describing one resource can be distributed over several graphs on the Web. Therefore, one Web page might only provide a subset of the information needed to be meaningfully processed by a desktop application. For example, Figure 2 shows the annotation of a scientific publication Web page using the SWRC ontology. The RDF graph only contains URI references of the authors of the paper. A bibliography manager the data is pasted to, however, would also need the full name of the authors to be able to create a new publication entry. If the Web application follows the recommendations of the W3C, information about the authors can be retrieved using the author's URI as URL. In SemClip the *Graph Completion* service retrieves this information directly after the RDF graph is loaded into the temporal RDF storage of SemClip.

For scalability reasons, the Graph Completion service does not recursively issue a request for every resource URI found in the RDF graph. The service only considers resource identifiers that are never used as subject in the RDF graph (*i.e.,* are not further described with properties except `rdf:type`). In the example of Figure 2 the Graph Completion service would request further information about the two author resources `.../id57` and `.../id32`. The service does not recursively repeat this procedure for the newly fetched RDF data, but only adds

---

[6] `http://www.ietf.org/rfc/rfc2426.txt`

```
1  <rdf:RDF
2      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3      xmlns:swrc="http://swrc.ontoware.org/ontology#">
4    <swrc:Article rdf:about="http://example.com/Publikationen/id1137">
5        <swrc:year>2005</swrc:year>
6        <swrc:title>Context-oriented knowledge management: an outlook</swrc:title>
7        <swrc:journal>Journal of Knowledge Management </swrc:journal>
8        <swrc:author rdf:resource="http://example.com/Personen/id57"/>
9        <swrc:author rdf:resource="http://example.com/Personen/id32"/>
10   </swrc:Article>
11 </rdf:RDF>
```

Fig. 2. RDF graph of a scientific publication using the SWRC ontology.

the Minimum Standalone Graphs (MSG) [8] to the originally pasted RDF graph. The MSG includes all statements with the requested resource as subject and recursively all statements linked via blank nodes. According to our experience, the MSG contains enough information to be meaningfully processed by the target application.

4 Implementation

This section discusses our implementation of SemClip. In general, every desktop application is a possible source application that can paste RDF data to Sem-Clip. Since there are not many applications available that support RDF, our current implementation focuses on semantically annotated Web pages and the Web browser as source application. We implemented a bookmarklet which makes any JavaSript enabled Web browser a source application for SemClip. When the user clicks on the "Copy RDF(a)" bookmarklet (Figure 3), the JavaScript parses the HTML page for RDF data that is attached through the `<link ref="meta" type="application/rdf+xml" href="link-to-external.rdf"/>` tag or included into the XHTML Web page as RDFa[7]. The bookmaktlet also checks for GRDDL[8] profiles. It opens a HTTP connection to SemClip and hands over the URLs of the external RDF descriptions and the URL of the Web page if it is annotated with RDFa or GRDDL. SemClip then loads the RDF descriptons and/or the Web page, extracts the semantic annotations, and puts them in the temporal RDF storage.

The Ontology Mapping Library, the Transformation Engine, and the Graph Completion service are implemented as Web Services and can also be used standalone beside SemClip. The Mapping Library provides a Web interface that allows developers to add new mappings.

Since most desktop applications are not able to import RDF data from the clipboard, we implemented a wrapper that pulls the data from SemClip via a

---

[7] http://www.w3.org/TR/xhtml-rdfa-primer/
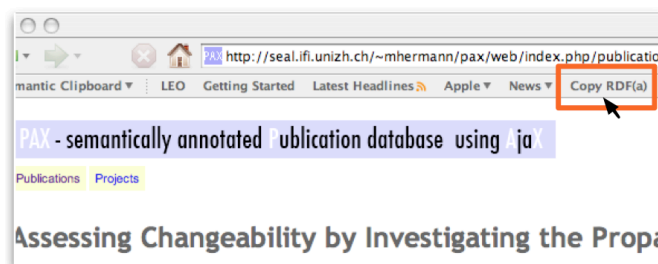[8] http://www.w3.org/TR/grddl/

Fig. 3. Copying semantic annotations to SemClip with the bookmarklet.

HTTP connection and integrates the data into the semantic context of the desktop application. The wrapper is implemented as AppleScript. To be able to trigger the AppleScript via a keyboard shortcut we use the Quicksilver tool[9]. When the keyboard shortcut is pushed, the AppleScript determines the currently used desktop application and requests the data from SemClip that uses the ontology the application is able to handle. Since this might include an ontology transformation (including a request to the Mapping Library) this operation might take some time. To inform the user that the paste request is being processed, we use the Growl tool[10] to display a message to the user.

The use of ontology mediation and the data wrapper makes SemClip flexible to support new ontologies and target applications. The current implementation supports the following ontologies: FOAF, vCard, SWRC, and W3C geo point[11] and is able to paste the data to the following applications: Apple Address Book, iCal, Google Earth, BibDesk, and Finder. In addition, we extended the application wrapper that not only the Web browser but also other desktop applications, such as the address book or Google Earth, can copy RDF data to SemClip.

Just as the operating system, SemClip should be running on the users desktop and only connect to the Mapping Library to download mapping rules that are not cached locally. For the Semantic Web Challenge, however, we installed the SemClip and its Web Services on our group Web server to relieve the user from installing the Web service environment on his computer. The bookmarklet and AppleScrip that access SemClip running on our server can be downloaded from `http://seal.ifi.unizh.ch/semwebchallenge`

5 Related Work

The email client of the upcoming version of Mac OS X, Leopard, parses email bodies for address and date patterns and allows users to add this information to

---

[9] `http://quicksilver.blacktree.com`

[10] `http://growl.info/`

[11] `http://www.w3.org/2003/01/geo/`

the address book and calendar respectively. The email feature does not rely on semantic annotations of the emails but uses pattern recognition techniques.

Shift[12] follows the opposite approach to SemClip. Shift allows users to produce RDFa code from desktop objects (e.g., contacts, calendar entries) via drag and drop, that can then be used to annotate Web pages such as blog posts. Previous work by the same authors provided similar functionality as part of a blog authoring tool, and also contained a tool that would transfer data back into desktop applications [4]. In comparison to SemClip, however, this tool only operated on RSS feeds, and was therefore much more restricted.

The Live Clipboard [5] is a DHTML application that provides copy and paste functionality for data associated with a Web page. To transfer the data, it is wrapped into an XML document that requires special parsers at the source and client side. The Web Clipboard [9] is a semantically enabled extension of the Live Clipboard that is able to copy and paste RDF data. Instead of having to parse a complete resource description with each copy and paste operation, the Web Clipboard uses a small JSON[13] snippet which only contains an identifier of the resource and information where full information can be obtained from.

6 Lessons Learned and Future Work

In this section we discuss our experiences with SemClip and give an outlook on future work. Many Semantic Web applications confront the end-user with RDF triples. RDF, however, is mainly intended for machine-consumption and not well suited for human users [2]. Therefore, SemClip follows the interaction pattern of the operating system clipboard the users are familiar with and does not expose the user to any Semantic Web technology.

When we were looking for test data to be used in SemClip we browsed the Web for semantically annotated Web pages. The most commonly used ontologies we found are FOAF, vCard, iCal, Dublin Core, and SWRC. However, not all of the annotated Web pages found could be processed by SemClip because the RDF graphs showed syntactical and semantical errors. The most common errors we encountered are the use of wrong namespaces, typos in URIs that reference ontology concepts (*e.g.,* `foaf:person` instead of `foaf:Person`), or missing CURIEs[14] when using namespaces in the `href` attribute in RDFa. It seems that this meta-data has been manually added to the Web pages, but that the data has never been processed by any Semantic Web application.

The question that now arises is: Should Semantic Web applications that process semantically annotated Web pages be built tolerant and be able to cope with the errors, just as the Web browsers handle HTML since the mid-nineties? Or should these applications stick to the W3C recommendations and enforce the correct use of RDF/RDFa syntax, if they want to have their annotations reused? We decided to go for the second solution, even if we miss some data on the Web

---

[12] `http://kantenwerk.org/shift`

[13] `http://www.json.org`

[14] `http://www.w3.org/2001/sw/BestPractices/HTML/CURIE`

that cannot be pasted using SemClip. On the SemClip Web page we maintain a list of Web pages that are syntactically and semantically correct and can be used as data source to test SemClip.

One way to increase the quality of the semantic annotations is the use of tools that free the user from writing RDF/XML or RDFa syntax. E.g., FOAF-a-Matic[15] can be used to create correct FOAF annotations in RDF/XML. Shift can be used to create RDFa code from address book contacts or calendar entries.

The goal of the development of a Semantic Clipboard has to be the integration into the operating system and to enable desktop applications to directly write RDF data to and read from the clipboard, to make semantic wrappers obsolete. To come closer to this goal, we are currently working together with the open source community on integrating a Semantic Clipboard into the KDE Linux desktop. In addition, SemClip will be integrated into the Social Semantic Desktop developed within the NEPOMUK project. Furthermore, we are working on wrappers for MS Windows applications to enable Windows users to take advantage of SemClip.

Acknowledgments

References

1. T. Berners-Lee. Semantic Clipboard, January 2004. `http://www.w3.org/DesignIssues/SemanticClipboard`.
2. D. Kargar and M. M. C. Schraefel. The pathetic fallacy of rdf. In *The 3rd International Semantic Web User Interaction Workshop (SWUI 2006) at the Interantional Semantic Web Conference ISWC 2006*, Athens, Geogria, USA, November 2006.
3. G. M. Laube, G. Reif, and H. Gall. Architectural issues of the semantic clipboard as ontology mediation service. In *SemDeskDesign2007 Workshop at the ESWC2007*, Innsbruck, Austria, June 2006.
4. K. Möller, U. Bojārs, and J. G. Breslin. Using Semantics to Enhance the Blogging Experience. In *ESWC 2006*, pages 679–696, Budva, Montenegro, June 2006.
5. R. Ozzie. Live clipboard technical introduction, Last visited July 2007. `http://spaces.live.com/editorial/rayozzie/demo/liveclip/liveclipsample/techPreview.html`.
6. G. Reif, M. Morger, and H. Gall. Semantic clipboard - semantically enriched data exchange between desktop applications. In *Semantic Desktop Workshopat at the ISWC06*, Athens, Geogria, USA, November 2006.
7. World Wide Web Consortium (W3C) Semantic Web activity homepage. `http://w3c.org/sw`.
8. G. Tummarello, C. Morbidoni, P. Puliti, and F. Piazza. Signing individual fragments of an RDF graph. In *International World Wide Web Conference*, pages 1020–1021, Chiba, Japan, 2005. ACM Press New York, NY, USA.
9. Web clipboard demo, Last visited August 2006. `http://www.sparqlets.org/clipboard/home`.

---

[15] `http://www.ldodds.com/foaf/foaf-a-matic`