Standardization's all very well, but what about the Exabytes of Existing Content and Learner Contributions?

Felix Mödritscher [1)], Victor Manuel García-Barrios [2)]

[1)] Institute for Information Systems and New Media,
Vienna University of Economics and Business Administration,
Augasse 2-6, 1090 Vienna, Austria
felix.moedritscher@wu-wien.ac.at

[2)] Institute for Information Systems and Computer Media,
Graz University of Technology,
Inffeldgasse 16c, 8010 Graz, Austria
vgarcia@iicm.tu-graz.ac.at

1. Problem Definition

Standardization in the field of technology-enhanced learning focuses on structuring and aggregating assets to interoperable educational entities, like a course package. However, available standards and specifications in this area do not include an approach for addressing semantics embedded in existing content. This consideration might be useful for user-centered concepts, like learners tagging or commenting existing material, as well as for automated mechanism, like extracting relations or other meta-information automatically from the resources. In the upcoming section we indicate application areas and explain why available standards and specifications do not support these scenarios. Thereafter, we present a XML-based description language for semantics embedded in web-based content, which might be a solution for these use cases. Finally, we summarize and discuss some experiences on in-content semantics from former projects, particularly AdeLE (Adaptive e-Learning with Eye-tracking, http://adele.fh-joanneum.at) and iCamp (http://icamp.eu), and give an outlook on future work.

2. Application Areas and Shortcomings of Standards

In 2004 we came in the situation that we had to cope with semantic enrichment of existing learning content, precisely to enable facilitators to tag web-based resources. However, we did not find satisfying solutions for this problem. Particularly, standards and specifications did not fulfill our requirements at that time. In the following, four typical application areas are described, and problematic aspects of standard support are outlined.

First of all, existing learning materials include a lot of semantic information embedded in the content itself. Typically, one would ask how this kind of semantics can be described so that this semantics can be processed automatically. Researchers in the iCamp project developed a harvesting approach to extract learning object metadata from web pages by using a GRDDL ('Gleaning Resource Descriptions from Dialect Languages'; cf. http://www.w3.org/TR/grddl) transformation to generate XML-based metadata snippets. However, this solution requires that the metadata is embedded according to a pre-defined xhtml-schema. Furthermore, no TEL standard or specification is currently supporting this kind of semantics, so such a

extraction mechanism is either implemented within a system or, more dynamically, through GRDDL transformations which are pre-defined by experts.

Second, new influences from the Web 2.0 stream foster participatory and learner-driven approaches, like facilitators (AdeLE) or peers (iCamp) tag or comment learning materials. Herby, technological solutions would either require an 'intelligent' repository or authoring tools to modify the content. The first concept can be identified with the 'Knowledge Artefact Repository' in the APOSDLE project which manages the metadata for digital objects stored in external systems. The latter approach was implemented with the Semantic TAGging Editor (STAGE) in the AdeLE project, where facilitators can specify learning goals ('to read', 'to learn', 'to ignore') for elements of web-based instructions.

Third, describing in-content semantics with TEL-related standards would require authors to break down pages into smaller chunks and describe them separately, e.g. with LOM. Consequently, this would not only increase the efforts in authoring, but must be supported by the LMS in terms of some kind of content aggregation engine. Besides, determining the necessary standards and specifications also means that experts have to create a metadata scheme ex ante and users have to fill out metadata fields.

Finally, even if current TEL standards would support this kind of in-content semantics, tools which are brought in by learners would have to provide standard-compliant APIs for interoperability reasons. For instance, if a learner prefers to work with another Wiki system like the one integrated in the learning management system, user-given semantics, like tags or comments, might be isolated if the LMS does not offer, let's say, an ECMAScript-API and the external tool does not use it appropriately. This solution approach might not be generic and practical at all, as different standards and specifications might require an own API.

In any case, in-content semantics is obviously not sufficiently considered by standardization efforts although it might be useful or even necessary, e.g. for interoperability or automatization purposes.

3. Proposed Solution

As a solution approach, we tie up to the idea of microformat and propose a description language for these semantic in-content entities.

3.1 Microformats, The Semantic Arteries of the Web

Over the last years, the O'Reilly's 'Participatory Web' evolved a new generation of tools and systems, enabling users to create content, annotate or tag resources, etc. easily via web-based user interfaces. Behind such web-based applications, 2.0-driven platforms aim at being based on an open, interoperable data formats, either described with own standards like Atom or RSS 2.0 or being embedded into the content, e.g. in the form of an xhtml-snippet. To tribute to these basic data entities of Web 2.0 tools, an initiative named microformats.org (http://microformats.org) was founded in 2005. According to this community's definition two important aspects are addressed here: First, microformats are designed "*for humans first and machines second*". Second, microformats are considered to be a "*set of simple, open data formats built upon existing, widely adopted standards*".

Figure 1 shows that microformats base on XML and, particularly for the Web, on XHTML, two well-known and widely-used W3C standards. In a broader sense, they can be seen as content snippets implementing a special mark-up, like certain values for attributes, a certain tag-structure, or any combination of these characteristics. Moreover, web practitioners differentiates between elemental microformats which can be described with one specific tag (e.g. the rel or the class attribute) and compound microformats which comprise a structure containing one or more elemental microformats.

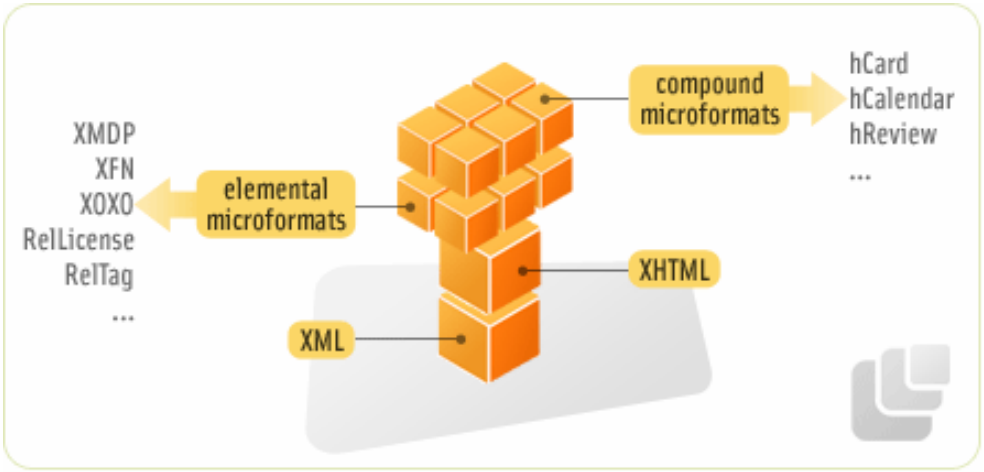

Figure 1: Overview of and examples for microformats (taken from http://microformats.org/about)

Typical examples for an elemental microformat are tags described with the rel-attribute (RelTag) or Xhtml Friend Network (XFN) allowing users to specify their relation to other people. A compound microformat would be hCalendar which stands for a way to manage collections of events within web-based content. In any case, microformats support the idea that web content must not be split up in smaller information junks to be semantically described further, but that the semantic is embedded into the content itself without changing its meaning. As an analogy to a medical view on the human body, we consider microformats to be the 'semantic arteries of the web', as they deliver semantics into web-based corpora.

Nevertheless, we see clear shortcomings of support of microformats, particularly in the field of technology-enhanced learning. In particular, own research activities or learning solutions might require own kinds of microformat, which must be spread and promoted by the designers who are in need of them e.g. to communities like the one behind microformats.org. This process is not only time-intensive and, therefore, costly, but also has a small change of having an impact on e-learning platforms and tools in order to sustainably guarantee data interoperability. Primarily for own prototypical approaches it would be beneficial to have a description language of microformats embedded within learning content. On the long term, this way of specifying in-content semantics might be also provided by standards and specifications aiming at describing (web-based) resources, like LOM or DCMS.

3.2 Data Model for a Microformat-Enabled Resource Description

Methodologically, there are two possibilities to describe microformat-based semantics embedded within the content: (1) Providing a XSL schema and, thus, describing what content junks should be extracted in which particular way, as also done with GRDDL transformations; (2) Developing an own, more simplified way to specify in-content semantics.

In this paper, we decided to follow the later approach, as we want to reduce complexity of the semantic description language and try to avoid describing how to extract the microformat-based semantics. However, a mapping from our data model to XSL can be easily made up.

Practitioners list 12 concrete examples for microformat specifications, beginning from elemental ones, like rel-license, rel-tag or VoteLinks, up to compound microformats, such as hCard, hCalendar or hAtom. Nevertheless, there exist much more specifications; an overview of them can be found at http://microformats.org/wiki/Main_Page. In order to describe all these in-content semantics, a data model has to consider the following aspects in terms of required or optional attribute fields:

- **Type:** The first issue to determine is if one wants to use an elemental or a compound microformat.
- **Identifier:** Second, this kind of specification of in-content semantic requires, like all other resource description standards, some kind of identifier, so that semantic applications or humans can differentiate between the semantic elements.
- **Design pattern:** Third and most important, it is necessary to describe which specific design pattern one wants to address. By definition, microformat design patterns comprise a formalism to "*reuse pieces of code that are generally useful in developing new microformats*". On other words, design patterns determine which XHTML elements and attributes are used to define a certain microformat. Thus, we propose to describe such a design pattern according to these two entities: (1) the element name, and (2) the attribute name. Assuming that a microformat is always based on an attribute, we consider the element as optional and the attribute as required. Furthermore, it should be possible to combine elements according to different attributes.
- **Matching string:** In order to restrict the XHTML attribute of the design pattern, an optional field for string matching is introduced. Values of the specified XHTML attributes are evaluated on basis of string equivalents as well as regular expressions.
- **Scope:** The scope, again, is optional and restricts the scope of the semantics within the web-based content. If given, the in-content semantics is valid within all DOM elements specified by this field.
- **Selector:** Another optional field, the so-called selector, is necessary to define from which source (XHTML element text or attribute) the semantics has to be extracted. If no selector is specified the value of the design pattern is used. Otherwise, an application might retrieve the value of the specified selector which, for instance, could be the title attribute or the text value itself.
- **Reference:** The optional reference field is of use to refer to another, existing microformat. Particularly, such a mechanism is useful for compound microformats, i.e. to avoid multiple definitions of elemental microformats. If referring to another microformat, all other fields except the identifier are ignored.
- **Optional:** Finally, the optional field indicates that an elemental microformat is optional within a compound one, which means that this element is not required to detect the compound microformat.

Figure 2 shows a possible description language for microformat-based, in-content semantics. In this example, seven elemental microformats (lines 2 to 7 and line 14) and one compound one (line 8 to 13) are specified. The first elemental microformat, namely 'xfn_met' (line 2), stands for a particular type of the commonly-known XHTML Friends Network (XFN) specification which can be determined with the rel-attribute having the value 'friend met'. For extracting semantics from such elements, the text-field (the value between opening and closing tag) has to be used via the selector-field. If no selector is given, an information

extractor simply uses the value of the specified pattern. Having such a specification of an elemental microformat, any application, even browser plug-ins like the Operator add-on (cf. http://addons.mozilla.org/de/firefox/addon/4106), can detect and extract this kind of semantic information from web-based content if supporting our data model.

```
1  <microformats>
2    <elemental id="xfn_met" pattern="rel" match="friend met" select="text" />
3    <elemental id="vote" pattern="a:rev" match="vote-*" select="title" />
4    <elemental id="vote_link" pattern="a:rev" match="vote-*" select="href" />
5    <elemental id="all_links" pattern="a:*" select="http://*" scope="/html/body" />
6    <elemental id="fn" pattern="class" match="^fn | fn | fn$" select="text" />
7    <elemental id="url" pattern="a:rel|div:rel" match="url" select="href" />
8    <compound id="vevent" pattern="class" match="vevent">
9      <elemental id="vevurl" ref="url" />
10     <elemental id="vevsummary" pattern="class" match="summary" select="text" />
11     <elemental id="vevstart" pattern="class" match="dtstart" select="title" />
12     <elemental id="vevend" pattern="class" match="dtend" select="title" optional="true" />
13   </compound>
14   <elemental id="goal" pattern="adele" match="to *" scope="/html/body/content"/>
15 </microformats>
```

Figure 2: Construction of the data model according to concrete examples

Line 3 and 4 demonstrate the VoteLinks microformat which are characterized in the way the rev-attribute of the a-tags start of with the value ‘vote-‘, whereby line 3 addresses the title of the voting and line 4 focus on the links. The asterisk represents any string that may follow, which, for the VoteLinks specification, means that there are different options (e.g. vote-for or vote-against). Line 5 describes a rather general microformat pattern to extract any link from web content. Here, all a-tags are examined to have attributes starting with ‘http://’ and being in the scope of the DOM element ‘/html/body’. Finally, line 6 and 7 show how to specify elemental microformats on different values or on the same attribute of different tags. Furthermore, line 6 makes use of a wide-spread microformat pattern, namely the class-design-pattern.

As an example for a compound microformat, we tried to describe the commonly-known hCalendar specification, consisting of the root-element ‘vevent’ and a set of required and optional elemental microformats. The very first elemental microformat is the URL, here given as a reference on a before-specified elemental. The, we included a summary text for the event, a start date and an optional end date. The very last line of the example in figure 2 we devoted to another elemental microformat, describing an own piece of semantics which utilizes the adele-attribute of DOM elements within the scope ‘/html/body/content’ for storing values with the prefix ‘to’. This specification of the adele-microformat was used in the AdeLE project, as pointed out in the following.

4. Experiences, Implications and Future Work

As mentioned before, we already dealt with in-content semantics and realized a couple of technological solutions to support it. In the APOSDLE project, we implemented a semantic layer which allowed us to integrate external repositories and manage their metadata separately from them. Hereby, it is possible to semantically enrich parts of documents. The harvesting approach realized in the iCamp project allowed automatic extraction of learning object metadata from web-based resources. Finally and mostly in line with the Web 2.0 idea, the AdeLE project supports a full life-cycle of in-content semantics. The STAGE tool, a plug-in for Firefox, empowers facilitators to tag existing web-based instructions, while the AdeLE LMS identifies the tagged information and uses it to adapt the learning process. In terms of the description language introduced in the last section, the in-content semantics of AdeLE is

described with line 14 in figure 2. For the LOM-harvesting approach, we would have to describe a compound microformat (the learning object itself) consisting of several elemental microformats, like the identifier, the title, and the forth.

Even though the AdeLE approach already covers authoring and exploitation a more generic solution would be an interpreter translating the description language into concrete actions for semantic technologies. From point of view of software engineering, such a component might consist of a Builder Pattern in combination with a Product Factory, so that the description of the semantics is read in and the targeted system is configured according to its purpose. Hereby, a harvester would use the description file to automatically generate a GRDDL file for information extraction. Furthermore, an authoring tool might fine-tune the storage filter, so that the xhtml-snippets are stored according to the description language. An LMS, on the other hand, would have to configure its content rendering engine, so that microformats embedded in the instructions are detected and used. To conclude this paper, it has to be said that our description language currently is nothing more than a starting point for standardization of in-content semantics.