Evaluating User Behavior as They Create Mappings in a Web Development System Using Local Radiance

Scott Britell[*,a], Lois M.L. Delcambre[a]

[a] Computer Science Department, Portland State University, Portland, OR, USA

Abstract. *Information integration with local radiance (IILR) is a system designed for use in web development frameworks that allows for the creation of polymorphic widgets based on small schema fragments and mappings to local schema that allow non-expert users to instantiate these widgets in their site. Here, we present results of a user study using IILR. We show that non-expert users can, and for the most part enjoy, creating the mappings required for our system. We describe the different behaviors observed of our participants and relate these behaviors to the survey data from our users.*

Keywords. Conceptual Modeling • Web Modeling • User Study

1 Introduction

The goal of our work is to facilitate and empower data creators and domain experts to perform complex tasks in web-based information systems that heretofore needed a database or web developer to accomplish. Modern web-based content management systems have enabled non-technical, domain-savvy users to store and publish data in rich structures with content types that can reference other content types. This effectively allows non-technical, domain users to define and populate their own conceptual models. While most domain users can publish their data in complex structures, configuring sites to use widgets typically requires expert developers. These widgets are often limited in their flexibility: the user's data must either fit the existing schema of the widget or the widget must be rewritten to accommodate the user's schema. Our approach provides more complex widgets that are written

generically against widget-specific schemas (*canonical structures*). These widgets can then be configured by non-technical, domain experts by creating simple mappings between their data in the schemas that they have created (*local schemas*) and domain-specific schemas (*domain structures*) that are isomorphic to the canonical structures.

Our previous work (Britell et al. 2014, 2016) presented the formal basis for our system and its implementations. Our IILR technology with generic widgets has been implemented in an operational, production website[1] with 6,500+ documents and 4,000+ users for over six years but without the ability for domain users to provide the mappings. In this work we present the results of our first user study where domain users with a range of technical expertise are asked to provide mappings. Subjects were provided a short training session and then required to use our system to create mappings for a widget in a website that they had not seen before. We show that all subjects of the study were able to successfully use the system. Subjects used the system in a number of different ways and we report on these different behaviors. The overriding goal of the study was to evaluate the feasibility of

---

---

[1] http://stemrobotics.cs.pdx.edu

our approach; we show that users generally were confident of their choice of mappings and enjoyed using our system.

This paper provides a brief overview of the IILR system in Section 2. Section 3 describes the design of our user study, including the participants, the test structure, the websites used, and the mapping interface used. We explore the different ways users interact with our system in Section 4. Results of the user surveys and their relationship to user behaviors are shown in Section 5. The paper concludes with a short discussion in Section 6.

2 Background and Related Work

Traditional structured information integration scenarios require that local schemas be mapped to a global schema; developers (and users) can then issue queries against the global schema to retrieve information from all participating local data sources. In our work, we share the goal of allowing multiple local schemas because we have seen that even within a single website, different groups of users may define distinct content types/relationships to describe semantically similar content. Thus our work supports information integration from multiple user-defined schemas (what we call local schemas). Note that one additional feature of our work is that we are able to extract the local schema names (based on the mappings in place) for display in our widgets - rather than providing query answers (against the global schema) that use only the global schema names. This feature that allows the local schema names to "shine through" to the global or integrated level is why we call our work Information Integration with Local Radiance.

Our work differs from traditional information integration approaches in that we use three levels of schemas, as shown in Figure 1. A generic schema (canonical structure) used by the widget developer, a domain-specific schema (domain structure) that is presented to the user defining the mappings, and the local schemas as defined by the domain users. We describe each of these in turn. We use a simple conceptual model where content

types/entities are shown as labeled rectangles and uni- and bidirectional references from one content type/entity type to another are shown as uni- and bidirectional arrows, respectively.

2.1 Canonical/Domain Structures and Local Schemas

An example of a canonical structure is shown at the bottom of Figure 1 to support a navigation widget, a preview of which is shown in Figure 2. The canonical structure represents only the essential structure required to write the widget code. For this example, the navigation widget presents the initial entity (i.e., Parent) with a nested display of all subordinate entities (i.e., Part). The navigation widget recursively displays further subordinate entities using the Parent - Part relationship, for all mappings that have been provided. Note that the canonical structure uses names related to the functionality of the widget and not to the domain of the application (library collections, in this example). The widget is thus domain-independent and can be reused in other domains.

A domain structure for use in this website is shown in the middle of Figure 1. This domain structure is isomorphic to the canonical structure shown at the bottom of the figure. A domain structure provides domain-meaningful names for the schema elements in the corresponding canonical structure.

A local schema for a library is shown at the top of Figure 1. Here we see that each Library references any number of Collections containing (i.e., referencing) any number of Books. Each Book may contain/reference Chapters which then may contain/reference Sections.

2.2 Mappings

Our system requires mappings at two levels as shown in Figure 1. One level of mappings is between the canonical and domain structures— shown as dotted black lines between the domain structure in the middle and the canonical structure at the bottom of the figure. We envision that the widget developer or a person charged with configuring widgets for a particular application
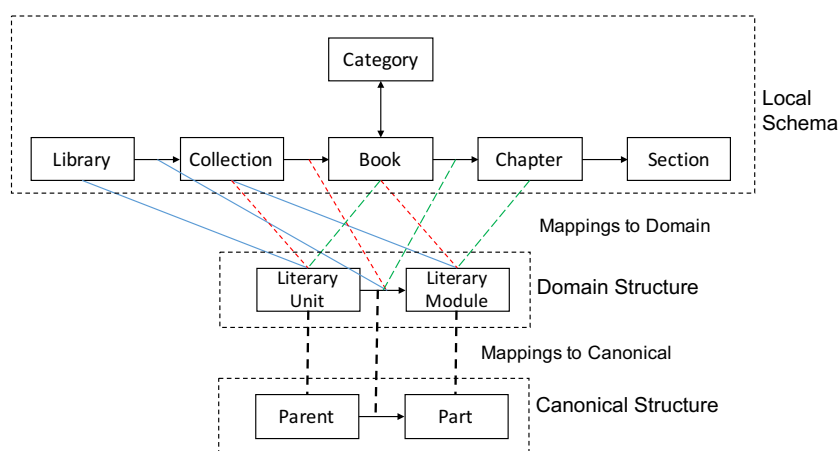
Figure 1: A local library schema (top), the library domain structure (middle), the parent-part canonical structure (bottom), and mappings between the three.

domain would define domain structures and also provide the mappings from the canonical structure used by a widget and the corresponding domain structure.

The focus of this paper is on the mappings from each local schema to the relevant domain structure (that has already been mapped to the canonical structure for the widget). An example of how the local schema for the library (at the top of the figure) could be mapped to the domain structure (in the middle of the figure) is shown with colored lines. The blue solid lines show that the Library-Collection portion of the local schema has been mapped to the Literary Unit-Literary Module portion of the domain structure. Similarly, the red dotted lines show that the Collection-Book portion of the local schema has also been mapped to the Literary Unit-Literary Module portion of the domain structure. And, the green dotted lines show the Book-Chapter portion of the local schema mapped to the Literary Unit-Literary Module portion of the domain structure. The navigation widget accordingly displays a library (The Metropolitan Library) with two collections (Astrology and Family Studies) where the Calendars books in the Astrology Collection has two chapters shown, see Figure 2. Note that the navigation widget displays the +/− symbol

that allows the end-user to expand/contract the navigational display, as desired.

2.3 Widgets

To summarize, the navigational widget, as shown in Figure 2 is written to query the canonical structure but is able to display the local data and schema because of the two levels of mapping provided, as shown in Figure 1. The widget always displays the data as mapped; if a mapping is added or removed, the widget immediately displays more or less information to the user.
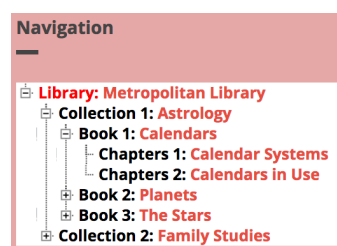

Figure 2: Widget using mappings from Figure 1

2.4 Related Work

Our work has been strongly influenced by work in schema mapping (Dessloch et al. 2008) and ETL (Miller et al. 2001) where users can draw simple lines between source and target schemas. While these tools are automated to facilitate the

schema mapping process, the mappings themselves and the tools to use them are targeted at expert database developers. We adopt this approach to allow non-expert users to do similar simple mappings.

We are also inspired by the field of end-user programming (Jones 1995; Lieberman et al. 2006) whose goal is to get non-developer users of software to create, modify, and extend that software. Current end-user web programming paradigms (Rode et al. 2005; Wong and Hong 2007) focus on allowing end-users to create "mashups" of existing widgets and data on the internet. We focus instead on letting end-users populate polymorphic widgets with their own data and schema.

We also believe that we can exploit the domain knowledge of our users to facilitate this process much in the same way it can be used to get non-expert users to perform semantic annotation (Hinze et al. 2012; Price et al. 2009). Our work is similar to approaches in conceptual models that introduce intermediate models between end-user specifications and system models (Mayr and Kop 2002; Vöhringer and Mayr 2006).

3 Design of the User Study

We designed our user study to test if subjects could create mappings between local schemas and domain structures. Subjects were asked to complete three tasks; one training task in which participants were guided through the process of creating mappings in a site with a fairly simple schema; and, two tasks where participants worked on their own to create mappings in two different sites (with a simple schema and a more complex schema). Subjects were given a demographic questionnaire at the beginning of the session, evaluation questionnaires at the end of each of the two testing tasks, and an overall evaluation questionnaire at the end of the session. As we expect our tool to be used by domain savvy users our test was limited to sites in a single domain (in this case, an educational domain).

For the training task, subjects used a website built using the library local schema shown in Figure 1. This schema has a simple hierarchy between Library, Collection, Book, Chapter, and Section. There is also a bidirectional relationship between Category and Book so that the subjects could create recursive mappings using the tool. The subjects were shown the structure of the site using only the hypertext links in the webpages within the operational website.

They were then shown the mapping tool (Figure 3) that, for a given domain structure, allows the user to select a content type from a list of all possible content types in the site and then choose a relationship associated with that type. Part of the mapping interface is a preview widget that shows how the navigation widget in the site would look using the given mappings. The tool then allows users to delete a specific mapping, save all of their mappings, or delete all of their mappings (Figure 4).
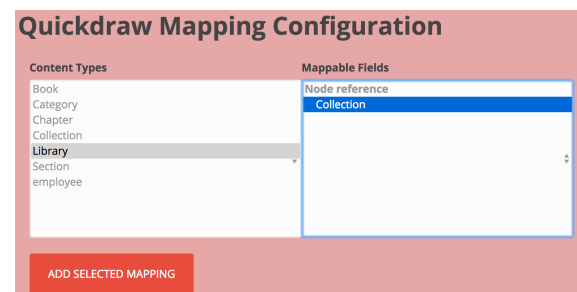

Figure 3: In the mapping interface, users select a content type (on the left) and then are shown all possible relationships to other content types (on the right).

In the training tasks the subjects were asked to create a number of specific mappings and encouraged to make additional mappings, as desired. There may be many different mappings that can be created within any given site for any number of reasons so we explicitly allowed our subjects to create whatever mappings they felt were appropriate. Since the choice of mappings is subjective, we did not test to see if subjects would create any specific mappings. Mappings were only deemed incorrect if the end result produced irregular widget behavior (e.g., duplicate mappings or disjoint
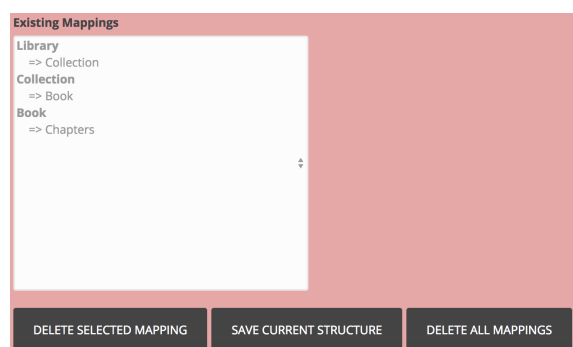
Figure 4: The mapping tool allows users to save their current mappings, delete specific mappings, or delete all mappings.

mappings). After the scripted part of the training session participants were allowed to explore the training site and the mapping tool for as long as they desired.

After the training task, participants were then asked to create mappings they saw as appropriate for a website with the schema shown in Figure 5. This schema is a simple hierarchy of an academic journal using unidirectional relationships only. For the second testing task participants were asked
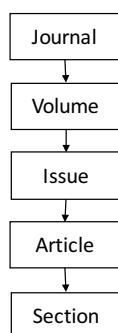

Figure 5: Schema for first task in the study.

to create mappings they saw as appropriate for a website based on a university schema shown in Figure 6. The schema was created with bidirectional relationships and cycles.

Participants for the study were recruited from the pool of departmental administration staff from the university who are in charge of updating the university webpage for their respective departments. All participants had working knowledge
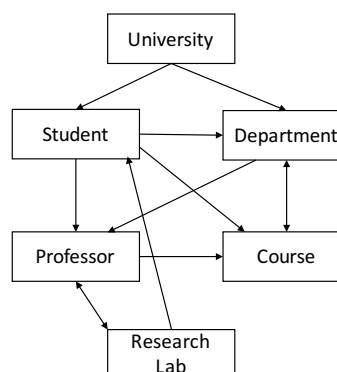

Figure 6: Schema for second task in the study.

of Journals, Libraries, and Universities. The participants had varying degrees of technical expertise ranging from three to more than ten years of website configuration experience and none to more than ten years of database experience.

4 User Behaviors

We showed participants how to browse the site, see a preview of the widget, and create large and small mappings. We emphasized the use of the preview functionality as we believed it would benefit the creation and checking of mappings.

Figure 7 shows an overview of the study subjects' sessions. Each subject's session is represented in a horizontal bar beginning with their anonymous id. The sessions are broken into boxes for each task, the first (pink) box shows the training task, the second (blue) box shows the first testing task, and the last (green) box shows the second testing task. The smaller boxes inside each box represent the various actions performed by the subject within the task (explained below). The longest session lasted a little less than 50 minutes and the shortest was less than 20 minutes. This is unsurprising given the open-ended nature of the tasks. In most sessions, subjects took a longer time with the second task likely due to the more complex nature of the local schema for the site in that task. For the two subjects who completed the second task, one created a single set of mappings for the university, without cycles, and decided they were done while the second appeared to rush
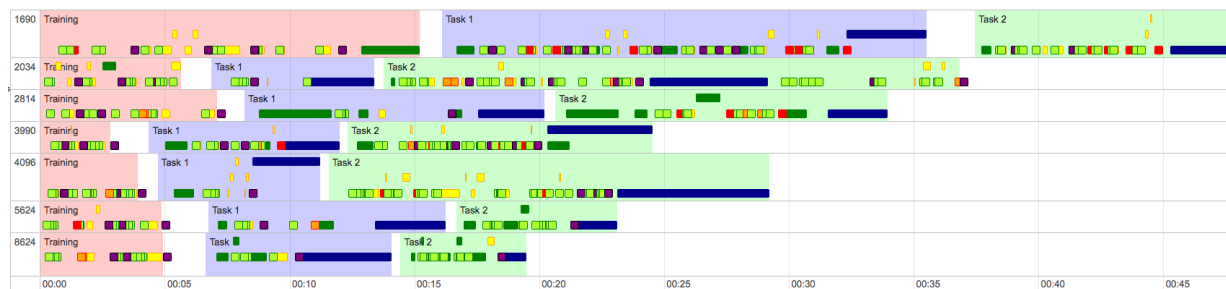
Figure 7: Timelines of user sessions showing length of training, task 1, and task 2.
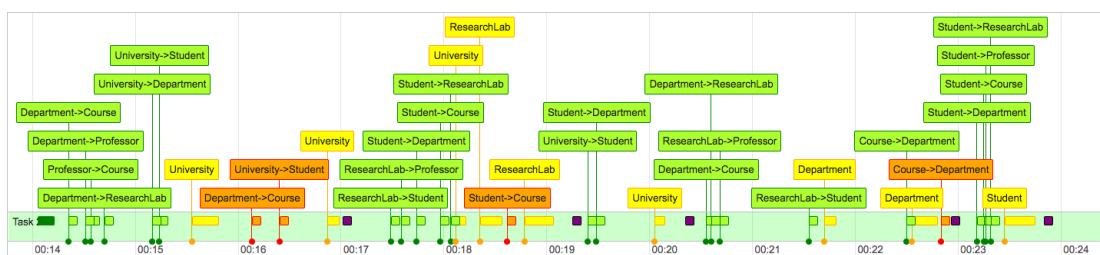

Figure 8: A study session of task 2 demonstrating the preview, test and check, and the enity-centric behaviors.

through the task and saved a set of mappings that included duplicate mappings. This was the only subject to save a set of mappings we deemed to be incorrect; other subjects created structures that had duplicate mappings or contained disjoint parts of the schema but in all cases these subjects discovered and deleted their bad mappings.

As mentioned above, we believed that the preview feature of our tool would be useful for checking and evaluating mappings. We found that some of our participants chose to use the preview feature while others chose to browse through the live site and see the live widget in the context of the actual webpages. Figure 8 shows an example of the use of the preview. In this case the subject starts the task by creating six mappings, uses the preview function to check the mappings, deletes two mappings, uses the preview again, and then saves the set of mappings. This subject continues creating a few mappings and checking with preview. Figure 9 shows another example of a previewer where the user starts by creating a single mapping, previews that mapping, and then saves the set of mappings. This user continues previewing after each new mapping. Contrast that with Figure 10

where the subject creates five mappings, browses the site, creates two more mappings, browses the site again, creates six more mappings, and then saves the set of mappings.

Figures 8 and 9 also demonstrate two other behavior patterns observed in the test. In Figure 8 the subject creates a few mappings, checks them with preview, then deletes a mapping or two before saving the set of mappings. Compare that with Figure 9 where the subject often creates a number of mappings and when they decide that they are incorrect or not to their liking they delete the entire set of mappings and start over.

The three examples shown thus far also demonstrate the two different ways subjects approached the process of mapping. In Figures 8 and 10 the subjects created larger navigational structures (with more mappings). Theses structures were also built in an entity-centric way, where the subject starts at one type, creates all the mappings related to that type and then moves to the next. Contrast that to Figure 9 where the subject creates many small structures and the mappings are often created in what appears to be a random fashion.
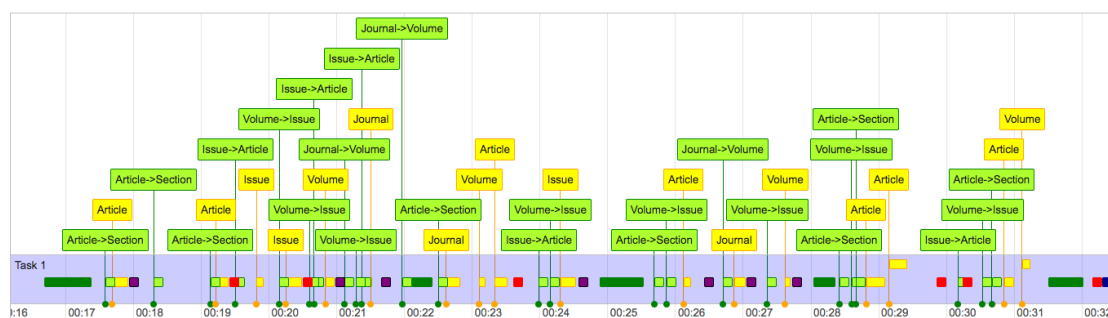
Figure 9: A study session of task 1 demonstrating the preview, delete and start over, and the random behaviors.
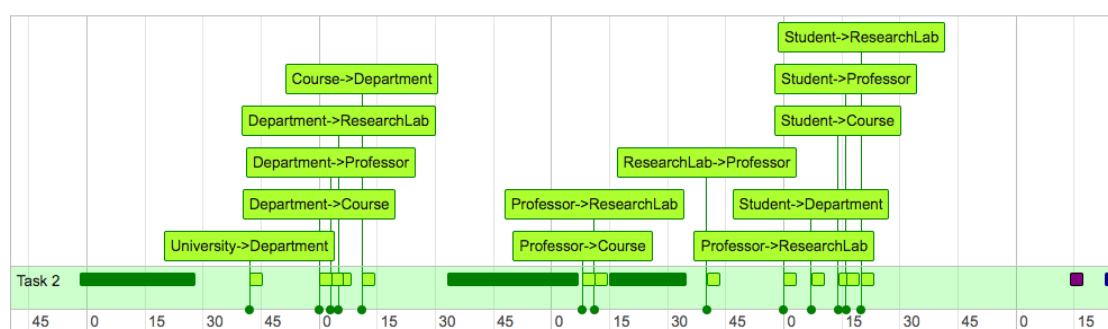

Figure 10: A study session of task 2 demonstrating the browse, large, and entity-centric behaviors. Darker rectangles represent browsing behavior, mappings are displayed above the lighter colored squares, and the dark square at the end saves all mappings.

5 Results

In regards to our main goal of this study—to see if domain savvy users can perform the mapping tasks in our system—the study was a success. All participants were able to complete the given tasks within a reasonable timeframe (we designed the test to take no more than one hour with the caveat that the open-ended nature could have made it take longer) and, although they could leave at any time during the test, no one left the test prematurely. (One participant inadvertently failed to complete the final questionnaire but completed all tasks and task questionnaires.) Participants were asked to rate the overall usefulness of the tool and enjoyment of mapping on a scale of one (Strongly Disagree) to five (Strongly Agree) with the average response for both questions being 3.7.

An interesting aspect of the behaviors listed above is how groups of differing behaviors corresponded to satisfaction results of the tool and mapping. Table 1 shows some of this detail. We found that the group of subjects that used preview was also the group with more than five years of experience. Those with less experience tended to exhibit more browsing behavior.

Table 1: Aggregated user feedback showing satisfaction with the system for each task and overall and a scale of one (Strongly Disagree) to five (Strongly Agree).

	Satisfaction		
Behavior Group	Task1	Task2	Overall
Previewer and Experienced	2.8	3.4	3.6
Non Previewer and Inexperienced	5	2	4
Larger and Entity-centric	4	3.25	4.7
Smaller and Random	2.7	2.7	2.7

Overall those with less technical experience tended to find the system more useful than those with more experience. In the questionnaires related to the tasks, those participants with more experience expressed some frustration that they were limited to what the tool could do when they knew how to edit HTML directly to get the results they wanted. Also of note is that even though inexperienced users preferred the tool overall they were less satisfied during the second more complex task.

Table 1 also shows the difference between the larger/entity-centric mappers and the smaller/random mappers. We see that across the board the larger/entity-centric mappers were more satisfied with the tool on each of the tasks and overall. It is likely that these subjects had a better understanding of the structure of the sites, our tool, and the tasks.

6 Conclusions

We use three levels of schema to facilitate end-users as they enhance their websites by mapping their local schemas to domain-specific schemas while allowing developers to maintain their own widget-specific schemas. Our test successfully demonstrated that non-expert domain users can perform the mapping tasks necessary to use our system. Overall subjects enjoyed using the tool. It is interesting that those with less experience appear to like our tool more than those with more experience, but we had a very small sample size (seven) and the user interface for our mapping tool was rudimentary, at best.

Our production website that implements IILR is written using the Drupal[2] web development framework. We have made our technology, including the mapping tools, available publicly to the Drupal community. Given our experience with our production website over the years and the evolution of our thinking regarding the IILR approach, we look forward to re-implementing the system including new implementations in other web development frameworks. This would likely present a cleaner, more fully featured, and more general implementation of IILR.

References

Britell S., Delcambre L. M. L., Atzeni P. (2014) Flexible Information Integration with Local Dominance. In: Information Modelling and Knowledge Bases XXVI, pp. 21–40

Britell S., Delcambre L. M. L., Atzeni P. (2016) Facilitating Data-Metadata Transformation by Domain Specialists in a Web-Based Information System Using Simple Correspondences. In: Conceptual Modeling: 35th International Conference, ER 2016 Springer International Publishing, pp. 445–459

Dessloch S., Hernandez M. A., Wisnesky R., Radwan A., Zhou J. (2008) Orchid: Integrating Schema Mapping and ETL. In: ICDE '08: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering. IEEE Computer Society, Washington, DC, USA, pp. 1307–1316

Hinze A., Heese R., Luczak-Rösch M., Paschke A. (2012) Semantic Enrichment by Non-Experts: Usability of Manual Annotation Tools. In: The Semantic Web – ISWC 2012: 11th International Semantic Web Conference. Springer International Publishing, pp. 165–181

Jones C. (1995) End-User Programming. In: Computer 28(9), pp. 68–70

Lieberman H., Paternò F., Klann M., Wulf V. (2006) End-User Development: An Emerging Paradigm. In: End User Development. Springer Netherlands, Dordrecht, pp. 1–8

Mayr H. C., Kop C. (2002) A User Centered Approach to Requirements Modeling. In: M. Glinz, G. Müller-Luschnat (Hrsg.): Modellierung 2002 - Modellierung in der Praxis - Modellierung für die Praxis Springer, pp. 75–86

Miller R. J., Hernández M. A., Haas L. M., Yan L., Howard Ho C. T., Fagin R., Popa L. (2001) The Clio project. In: ACM SIGMOD Record 30(1), pp. 78–83

---

[2] http://drupal.org

Price S. L., Lykke Nielsen M., Delcambre L. M., Vedsted P., Steinhauer J. (2009) Using semantic components to search for domain-specific documents: An evaluation from the system perspective and the user perspective. In: Information Systems 34(8), pp. 724–752

Rode J., Bhardwaj Y., Pérez-Quiñones M. A., Rosson M. B., Howarth J. (2005) As Easy as "Click": End-User Web Engineering. In: Web Engineering: 5th International Conference, ICWE. Springer, Berlin, Heidelberg, pp. 478–488

Vöhringer J., Mayr H. C. (2006) Integration of Schemas on the Pre-Design Level Using the KCPM-Approach. In: Advances in Information Systems Development. Springer US, Boston, MA, pp. 623–634

Wong J., Hong J. I. (2007) Making mashups with marmite. In: Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07. ACM Press, New York, New York, USA, p. 1435