

URI-AWARE USER INPUT INTERFACES FOR THE UNOBTRUSIVE REFERENCE TO LINKED DATA

André Langer, Christoph Göpfert, and Martin Gaedke
Chemnitz University of Technology, Computer Science Department, Chemnitz, Germany.

ABSTRACT

Using appropriate entity URIs is a crucial factor for the success of semantic-enabled applications for data management and data retrieval. Especially data applications that collect data to build knowledge graphs rely on correct concept identifiers in favor of ambiguous literals. This collection involves human interaction in the web frontend without annoying the user. But appropriate user interfaces for this task are still a challenge. In this article, we focus on the design of form elements that unobtrusively allow input data both for human and machine interaction from a semantic point of view. Motivated by web-based scholarly document-submission systems, we first present a brief current-state analysis on the support of semantic input operations, investigate how these users input interfaces can be improved for concept linking purposes with an auto-suggestion behavior and finally evaluate with a proof-of concept implementation and user survey the advantages and acceptance of our approach.

KEYWORDS

Linked Data, User Interfaces, Usability, Data Management, Web Components

1. INTRODUCTION

Resources in the World Wide Web are traditionally published in a document-centric fashion. Taking the research community as an example, scientists publish their findings by digitally uploading a paper that contains natural language floating text via a document submission system, and may also add additional meta information such as a list of authors and their affiliation, categorization tags or other publishing information. Other researchers can then find and access this document information in the result list of a search-based environment by entering similar keywords. For the last decades, this approach has been easy, sufficient and convenient. But the amount of information on the World Wide Web increases rapidly and the demand for

new strategies and services requires to better publish, manage and retrieve particular knowledge in a structured fashion.

The Semantic Web community therefore proposed knowledge graphs as a graph-based organization of linked information in order to turn the traditional document-centric web into an entity-centric web of data (Shadbolt, Berners-Lee, & Hall, 2006). Multiple application domains already profited from such an entity-based knowledge representation as content can be found and filtered on a fine granular entity level, even in different representations.

Though, human interaction is often needed to add or modify information in such a knowledge graph. Thereby, 1. each entity related to a particular resource has to be described separately 2. with a persistent identifier that is 3. well-known to represent this particular real-world concept. Simple literal text input is usually not appropriate to accomplish this task as different spellings and interpretations could be used. So, human users would have to deal with the provision of unambiguous persistent identifiers to create business value but which will also be a very tedious activity.

A good user interface experience (UIX) is needed to lower the barrier and to motivate users to provide correct Linked Data references. Assisting the user during this input activity by presenting suitable information and hiding complex or unnecessary data in the current context is one possible approach. In web development, the HyperText Markup Language v5 (HTML5) therefore already provides a set of simple input elements. These input controls are still used in current web forms for simple literal text input although they could be enriched to better support the reference of external Linked Data. The unobtrusive usage together with semantic entity URIs is hardly considered.

In this article, we extend previous work (Langer, Göpfert, & Gaedke, 2018) dealing with the research question how to improve the user motivation and experience in Web-enabled applications with Web Engineering methodologies to let humans provide correct entity URIs without changing their regular interaction habit. Our contributions are:

1. We provide evidence through a comparison of web forms in established document-oriented data collection system, that referencing related entities with literal input or local IDs is still dominating and represents an obstacle for interoperability
2. We coin the term of URI-aware user input interfaces as an extension of traditional HTML text input elements with an auto-suggestion functionality that present human-readable content as well as machine-readable persistent identifiers on client and server side
3. We conduct an experimental user survey to show that URI-aware user input interfaces can considerably increase the quality of provided Linked Data references while maintaining the traditional interaction user satisfaction

The remainder of the article is structured in the following way. In section 2, we use the scenario of research submission systems as a motivating example, check the current support on existing web platforms and infer requirements on suitable user interfaces. Section 3 presents general characteristics of a URI-aware input element and discusses the functional extension of different types of HTML elements to turn them into URI-aware components. In section 4, we implement the presented approach as a proof of concept and evaluate in a usability study the opinion of multiple users. Section 5 contains the Related Work and sets our contribution into relation to other findings within the Linked Data community. In the last section, we summarize the entire topic and outline possible future extensions.

2. USER INTERFACE SUPPORT FOR REFERENCING RELATED LINKED DATA ENTITIES

We first provide a common scenario from the academia as a motivating example to illustrate the problem domain of our research focus, which we will later also use for evaluation purposes.

2.1 A Motivating Use Case

In a traditional research publication scenario, a user *Jane* has written a scientific publication and wants to submit this document so that other researchers can find it. So, she uploads a PDF file and specifies manually further meta data such as the names of contributing *authors*, the publication *title* and *abstract*, *keyword tags*, *language* and *license* information and others. This research publication can then be found on web platforms like *Bibsonomy*, *Easychair*, *Google Scholar*, *Mendeley*, *ResearchGate* or *Zenodo* by other users.

For that purpose, such a document submission system presents the user a web page with an input form where the concrete document can be uploaded and optionally further processed on server-side to extract additional meta data in an automated fashion, and additional meta data can be entered.

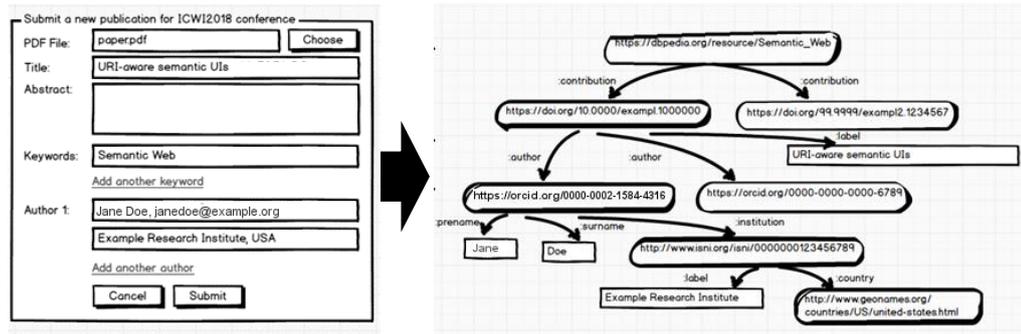


Figure 1. Meta data input in a web form and possible resulting, desired knowledge graph structure

This extracted document data together with implicit document meta data and the manually provided explicit meta data by the user can then be used as the base for search operations performed by other users the find relevant resources such as the provided document file for consumption. Traditionally, these search operations are normally text-based as users enter search terms as strings to find related content that either occurs in the resource itself or in the associated meta data. Normally, this approach returns adequate results, but can lead to false-positive hits that either contain text with ambiguous meanings (e.g. looking for “*operation*” can refer to *mathematical operations* as well as to documents dealing with *activities*) or entities with same strings (e.g. searching for authors with the name *Jones*). In addition, it is difficult to retrieve related contributions on a more detailed level than the publication title or keywords of such a paper (such as the interest in a particular methodology or result artifact type). It would create value if the publication and its content is described in an unambiguous, structured fashion to allow better retrieval operations and to improve visibility and reuse.

An accepted means to accomplish this objective is the construction of knowledge graphs as depicted in Figure 1. Therefore, persistent Uniform Resource Identifiers (PID/URI) are required to represent distinct RDF entities. Identification schemes already exist for common items, such as *ArXivID*, *DOI*, *PMID* or *PURL* for representing data sets or documents, *Dublic Core* vocabulary terms for describing bibliographical elements, or *FOAF* and *ORCID* for symbolizing human entities. To simplify naming, we will refer to this entire group of identifiers in the following as URIs, which also encompasses the term persistent identifiers (PID). These URIs are representing a comprehensive real-world concept with a unique identifier, but although they are readable they are hard to remember and abstract for a human user.

Users are accustomed to enter literal text content in an input form instead of providing URIs. To solve this discrepancy in a semantic-aware web application, users were either explicitly imposed to enter an explicit identifier, which involved tedious Copy&Paste operations and commonly resulted in a bad response rate for optional data, or they were asked to simply enter natural language text, and consecutively a mapping operation was performed in the background to transform the entered word to a representative URI. But as entity recognition and mapping is not always reliably solving ambiguities or able to find a representation for unknown concepts at all, the challenge is to both gather explicit Linked Data references via standardized input interfaces as well as to motivate human users to provide this explicit data. In the following, we will concentrate on the well-grounded research of this UI improvement.

2.2 An Analysis of Current User Input Interfaces

As a starting point, we first provide an analysis and comparison of current-state web user input interfaces from existing document-collection systems in the research domain. We are interested, to which degree established submission platforms already take care of a semantic data input management and how their user interfaces support this activity. Six service providers for research data management were conscientiously chosen for this scenario, in detail *Bibsonomy*, *EasyChair*, *Google Scholar*, *Mendeley*, *ResearchGate* and *Zenodo*. Although these systems differ in their focus area, similarities can be identified when simply concentrating on the submission form user interface and the provided input elements. As shown in figure 2, these user interfaces are still dominated by plain text input elements such as input fields and text areas. They are not only used for string input operations for individual floating text such as a document title or description, but also for gathering entity-related meta information such as author names and affiliation, keywords to represent categories, or license information. It appears odd that platform providers attempt to collect additional, structured data for better retrieval operations from a limited value range but allow an unrestricted input in many cases where different spellings impede consecutive business operations. It remains unclear, if additional effort for more sophisticated user interfaces was simply not invested, or if linking of data was not in the application focus in the past. In contrast to our previous analysis (Langer et al., 2018), the situation improved in such a way, that commercial providers already enhanced their web form interfaces during the last four months. The application providers *Mendeley* and *ResearchGate* started to reference entities such as authors and licenses in a more structured fashion, however, they still use local ID values for their representation or do not provide an ID on frontend side at all.

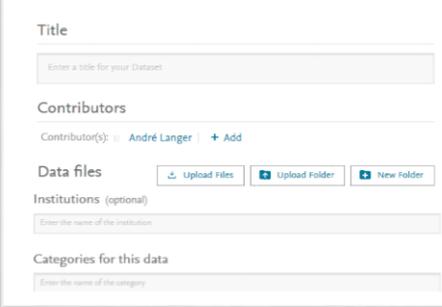
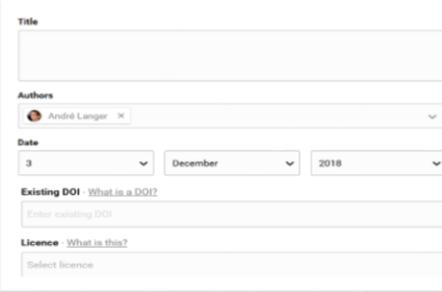
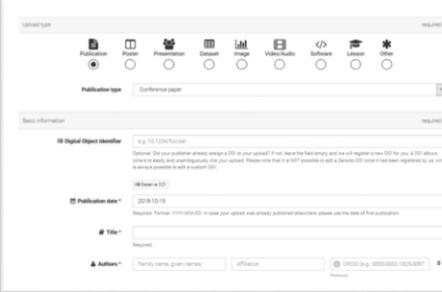
 <p>...ng publication metadata fields.</p> <p>entry type <input type="text" value="article"/></p> <p>The entry type of the publication. Learn more.</p> <p>title <input type="text"/></p> <p>This field is required.</p> <p>author(s) <input type="text"/></p> <p>This field is required.</p> <p><input type="checkbox"/> I am an author</p>	 <p>The title and the abstract should be entered as plain text, they should not contain HTML elements.</p> <p>Title (*) <input type="text"/></p> <p>Abstract (*) <input type="text"/></p> <p>Keywords</p> <p>Type a list of keywords (also known as key phrases or key terms), one per line to characterize your submission. You should specify at least three keywords.</p> <p>Keywords (*) <input type="text"/></p>
<p>Bibsonomy</p>	<p>EasyChair</p>
 <p>Zeitschrift Konferenz Kapitel Buch</p> <p>Titel <input type="text"/></p> <p>Autoren <input type="text"/></p> <p>Zum Beispiel: Patterson, David; Lampert, Leslie</p> <p>Publikationsdatum <input type="text"/></p> <p>Beispiel: 2008, 2008/12 oder 2008/12/31</p> <p>Zeitschrift <input type="text"/></p>	 <p>Title <input type="text"/></p> <p>Enter a title for your Dataset</p> <p>Contributors</p> <p>Contributor(s): <input type="text" value="André Langer"/> + Add</p> <p>Data files <input type="button" value="Upload Files"/> <input type="button" value="Upload Folder"/> <input type="button" value="New Folder"/></p> <p>Institutions (optional) <input type="text"/></p> <p>Enter the name of the institution</p> <p>Categories for this data <input type="text"/></p> <p>Enter the name of the category</p>
<p>Google Scholar</p>	<p>Mendeley</p>
 <p>Title <input type="text"/></p> <p>Authors <input type="text" value="André Langer"/></p> <p>Date <input type="text" value="3"/> <input type="text" value="December"/> <input type="text" value="2018"/></p> <p>Existing DOI <input type="text"/></p> <p>What is a DOI?</p> <p>Enter existing DOI</p> <p>Licence <input type="text"/></p> <p>What is this?</p> <p>Select licence</p>	 <p>Select type <input type="text" value="Publication"/></p> <p>Publication type <input type="text" value="Conference paper"/></p> <p>Basic information</p> <p>DOI <input type="text"/></p> <p>Publication date <input type="text" value="2018-12-18"/></p> <p>Title <input type="text"/></p> <p>Authors <input type="text" value="André Langer"/></p>
<p>ResearchGate</p>	<p>Zenodo</p>

Figure 2. Meta data user input interfaces in current document submission systems (November 2018)

Table 1 provides a feature matrix to compare multiple supported features among the selected application platforms with a focus on Linked Data input management in real-world document submission applications. Relevant assessment criteria are formulated in the first column. Platforms, that do not support this criterion at all, are indicated with (-), with (o) if they partially support it, with (+) if a regular plain text input is possible and with (++) if they support it with a sophisticated UI control.

Table 1. Comparison of document-centric submission web UIs of selected application providers

	Bib- sonomy	Easy- chair	Google Scholar	Men- deley	Research gate	Ze- nodo
File Submission ?	+	+	O	+	+	+
URI/DOI for file?	-	-	-	+	++	+
Author meta data input?	+	+	+	++	++	+
URI for Author?	-	O	-	O	O	+
Title input?	+	+	+	+	+	+
Abstract / Descriptive text?	-	+	-	+	+	+
Keyword/Tag/Category input?	+	+	-	+	-	+
URI for keywords?	-	-	-	-	-	-
Additional meta data?	+	O	+	+	+	+
URI for additional meta data?	-	-	-	-	-	-
Language input?	+	-	-	-	-	+
URI for language?	-	-	-	-	-	-
License input?	-	-	-	++	++	+
URI for license?	-	-	-	-	-	-
Publication detail data?	+	(+)	+	+	-	+
URI for Publication detail data?	O	-	-	-	-	O
Explicit document artifacts?	-	-	-	-	-	-
URI for explicit document artifacts?	-	-	-	-	-	-
Reference data input?	-	-	-	++	++	+
URI for reference data?	-	-	-	+	+	-

As shown, all user interfaces allow the submission of a file artifact together with common bibliographical data such as a title, author list, description, and category. For identifiable real-world entities such as an author, some user interfaces already ask for the selection from a list of known author names, but without any globally valid URI reference. For other entities such as categories, this strategy is not similarly followed where only literal input is commonly expected. Other relevant meta information such as date or language information is often neither requested at all nor referenced in an easy-processable way. At least, further publication information regarding the conference, journal or year is often requested in a literal fashion. It would also make sense to reference the location of the publication in an entity fashion. More fine-granular structured information for document artifacts such as contained definitions, code snippets, figures, tables, or other sections could not be found at all. Two applications at least allowed the specification of related contributions.

The analysis shows, that concealing the management of Linked Data URIs by instead presenting readable labels is not the common case in web forms so far for input interfaces. Although entities such as author names can already be specified as separated entities in favor of plain text, application providers then handle these objects with local, internal IDs. Other meta data information is still requested via plain text input fields and managed as a literal string.

So, the situation for gathering structured, semantic data in a systematic fashion is not ideal in order to build knowledge graphs from this data for semantic web applications, where data can be linked together and set into relation on a more detailed level than simply based on file resource URIs. To improve this situation, we define the following list of requirements on user input interface elements in applications dealing with Linked Data:

REQ1: Human-readable literal text can be entered or selected so that the existing interaction habit of a user is not changed

REQ2: One or multiple corresponding URI value(s) are returned for the provided entity label to the application

REQ3: Key-Value mappings shall be either statically provided or dynamically retrieved from an appropriate remote web service

REQ4: Optionally, an explicit entity URI shall be providable if none of the presented mappings represent the entity of interest

REQ5: Existing standard web UI components shall be used

3. URI-AWARE INPUT ELEMENTS FOR IMPROVED SEMANTIC DATA HANDLING

Based on our research question, how users can be motivated to provide Linked Data references with existing web input elements that transparently manage entity URI values, we present in this section a well-grounded approach on how to enrich standard input elements available in W3C's HTML5 standard with LD URI support capabilities for several conceptual use cases.

3.1 Characteristics of URI-aware Input

In the following, we refer to frontend user input interfaces that are capable of managing URIs/PIDs as input values for human readable entity labels as *URI-aware UI*. They shall allow a better semantic data management starting already with the input operation step while also supporting a human user with assistance functionalities to get correct entities URI for Linked Data relationships. We thereby assume, that entity URIs shall be favored for RDF object definitions instead of simple literals, however it does not imply, that URI-aware UI shall be used for all kind of input data because it makes no sense to represent each individual information piece with a persistent identifier.

URI-aware user input interfaces have the following characteristics:

- UI interaction based on labels instead of URIs
- Auto-suggestion functionality with a particular scope of already known data entities
- Provision of corresponding URIs as a data value to other frontend components
- Fallback option, if none of the known entities matches the desired object

3.2 Enriching Common Input Elements

In this section, we discuss several types of W3Cs HTML5 input elements and the procedures to turn them into URI-aware user input interfaces.

3.2.1 Input Text

In HTML5, the `<input>` element natively enables a user to manually input a text string. Such a string can refer 1. to a plain literal information as well as 2. to a public general URI; in our example scenario e.g. an *author name* or a *topic (keyword)*. Gathering in case 1 literal content is an already well-understood task. In order to better process the provided data in a semantic fashion, web developers are encouraged to consider the explicit specification of corresponding data attributes (such as `datatype` or `content language`). Linking in case 2 to a known entity is interesting for us, as we want to know the identifier of the entity that is referenced. The trivial approach is to explicitly require the user in an input field to insert a URI for the desired entity, selectively decorated with a `type="url"` attribute and other DCterms properties.

Yet, manually dealing with an explicit URI may result in a bad response rate and the user interface experience is cumbersome for a user. Instead, the input process should remain familiar when dealing with Linked Data and simply enhanced with some assisting functionalities. This can include a mapping between an entity label and a URI via *auto-suggestion*, either statically or by using a remote endpoint dynamically. I.e., for person entities, we could use the *ORCID API*, or services such as *Wictionary* or *ConcetNet* for categorization entities. An HTML5 *datalist* that is bound to a text input field can be considered as a solution for enriching a basic text input element with LD URI management capabilities as shown in figure 3. A *datalist* allows the definition of a set of input values together with an alternative label that is considered during text input.

```

01 <!DOCTYPE html>
02 <html><body><form>
03   <input name="keyword" list="keywords"/>
04   <datalist id="keywords">
05     <option value="http://www.wikidata.org/entity/Q54837" label="Semantic
06     Web">
07     <option value="http://www.wikidata.org/entity/Q47146" label="User
08     Interface">
09   </datalist>
10   <input type="submit"/>
11 </form></body></html>

```



Figure 3. Enrich a text input element with an auto-suggestion for managing Concept URIs

One drawback of this solution is, that the presentation and behavior of the displayed suggestion varies between different web browsers. Additionally, the URI value is visibly inserted into the related input element after selecting the intended entity. This can be desired for editing and verification purposes, but an advanced, better solution is needed if the user shall simply interact with literal content. In fact, the user does not even have to interact in any way when the desired entity is already defined in a distinct way.

In particular, the following five cases have to be distinguished in the auto-suggestion process how a Linked Data URI can be determined based on the literal input provided by the user that describes a specific entity:

1. Single bijective match between the entered entity name and the corresponding Linked Data URI is a correct representation of this object: then, the suggestion does not even have to be displayed, as the corresponding URI can be retrieved in a distinct way
2. Single bijective match between the entered entity name and a corresponding Linked Data URI but which represents a different concept: then, the user should get a feedback of the retrieved and likely URI result with a possibility to change the identifier
3. Multiple URI matches for the entered entity name that represent different concepts: then, the user should get the possibility to select the desired concept
4. Multiple URI matches for the entered entity name that represent the same concept: then, the user should get the possibility to select the favored URI
5. No matching URI for the entered entity name found at all: then, we need a fallback as we need a URI in every case to build a knowledge graph:
 - Ask the user manually for an explicit Linked Data URI
 - Auto-generate a URI
 - Map the input data to a pre-defined URI indicating that no corresponding entity URI is known so far

We applied an advanced dropdownlist component to a text input field that can deal with these cases. Figure 4 shows, that the underlying entity URI does not even have to be visible for the user in order to select the correct concept that is meant with the entered input string.

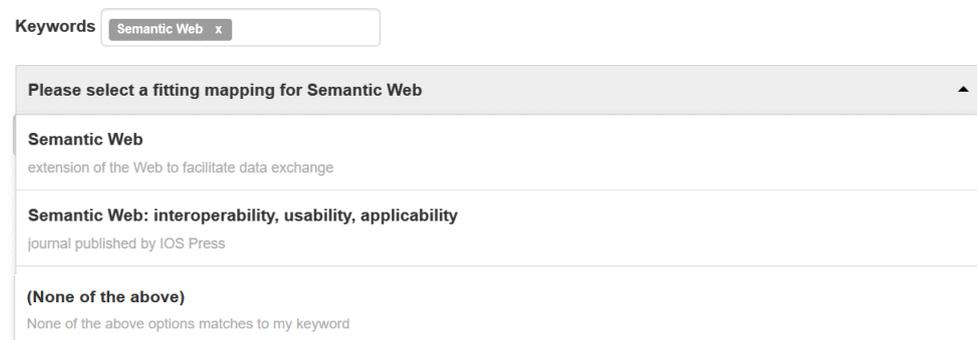


Figure 4. Enrich a text input element with a data list auto-suggestion for managing Concept URIs

3.2.2 Selection Elements

Controls like select lists, radio button groups and checkbox groups are other input elements that allow the selection of predefined options which a user can choose. They can be used to specify additional meta data such as a *document license* or *language*. Values for appropriate Linked Data URIs are e.g. *http://creativecommons.org/licenses/by-** for a license, or *http://lexvo.org/id/** for language information (in favor of IETF language tags).

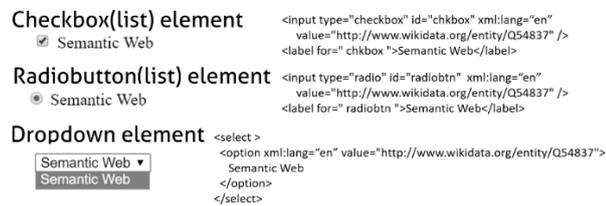


Figure 5. Three types of URI-aware HTML5 selection elements, exemplarily with one option

In HTML5, these tags allow the built-in definition of key/value pairs. In traditional web applications, local IDs are commonly used as a value for a key label. Thus, in order to make these UI input elements URI-aware, developers should instead provide carefully selected Linked Data URIs as a value. They can either be defined statically or retrieved dynamically from a remote endpoint in the Semantic Web. Providing an underlying global persistent identifier already on the frontend side increases interoperability and enhances the content processing by external tools. Additionally, an “Other” option with an explicit URI input field or a “None of the above” option should be provided as already discussed in section 3.2.1 as best practice.

3.2.3 Referencing Entity Parts

There are entities apart from already discussed concepts that can be used as descriptive meta data in our document submission scenario. If we want to set data into relation to other related contributions on a fine granular level, this might also involve the reference to a particular part of the current document such as included *figures*, *tables*, *definitions*, *code snippets* as well as *bibliographical reference data* that is related to the current publication. If the submission is already identified by a persistent identifier, e.g. DOI, a URI for a particular entity within this document can easily be created e.g. by using a fragment URI built from the document URI and a fragment identifier. This applies both to a detailed internal description of the current document content as well as a reference to external sources. In the first case, a URI-aware user input interface can rely on preprocessed structural information of the current object, in our case a document, and provide meaningful labels in suggestions for all identified parts of this object. In the second case, APIs of publication collection platforms can be consumed to refer to referenced literature with correct title/URI key/value pairs in favor of simple literal text for referenced literature.

3.3 Component-oriented Realization

To implement the described characteristics of URI-aware user input interfaces, a straight-forward reusable approach to extend existing HTML5 interfaces is desirable. W3C WebComponents allow the component-based usage of URI-aware user input interfaces. Such a component can be defined as shown in figure 6. *Uriaware-input* is an exemplary realization of a text field that allows the multiple input of data labels (here representing keywords). When entering a string, an auto-suggestion overlay is interactively opened. Elements for this *uriaware-list* can either be defined statically (*static-option*) or retrieved from a remote endpoint (*dynamic-option*). A combination of multiple options is also possible, If none of the presented options matches, an *alternative-option* can be defined and selected, either by specifying a fixed URI or by providing an explicit URI by the user.

```

01 <!DOCTYPE html>
02 <html><head><script src="uri-aware-wc.js"></script></head>
03 <body>
04 <form>
05   <label for="keywords">Keywords</label>
06   <uriaware-input name="keywords" id="keywords" multiple-selection="true"
07     order="ascending" alternative-type="fixed" >
08     <static-option value="http://dbpedia.org/resource/Linked_Data" label="Linked Data" />
09     <dynamic-option src="https://www.wikidata.org/w/api.php?action=wbsearchentities&search=" />
10     <alternative-option value="http://dbpedia.org/resource/other" label="None of the above" />
11   </uriaware-input>
12   <input type="submit"/>
13 </form></body></html>

```

Figure 6. Implementing a URI-aware text input as a WebComponent

4. EVALUATION

In order to evaluate our approach, we first created a prototypical implementation as a proof-of-concept for URI-aware input element. Based on that, we conducted on a website an unsupervised experimental survey. With this study, we tested if the provision rate for entity URIs increased when using improved user interface elements for Linked Data reference input among multiple participants. The prototypical implementation and the user survey can be accessed via <https://purl.org/net/vsr/semanticui>.

We applied the already discussed use case of submitting a *scientific document via an input form on a web page*. Therefore, we asked participants to finish *six* different tasks through interaction with the provided web user input interface. In the first three tasks, they had to enter author information together with an appropriate ORCID persistent identifier. In the other three tasks, we asked to provide categories to classify the document content by providing appropriate keywords and a corresponding entity URI. Each task presented a different user interface to accomplish this task. First, a traditional literal input without any support, then a plain entity URI input element, and finally a traditionally looking form enriched with our URI-aware auto-suggestion approach. After each experiment, all participants were required to answer a set of provided statements to subjectively assess their personal user experience based on a five-level Likert rating scale.

Based on (Nielsen & Landauer, 1993), we focused on a small test user group in favor of a larger set of questions on the interface usability to ask each candidate afterwards. Eight users participated in total. According to Nielsen et. al, this test set is appropriate for identifying usability issues and opinions, as results would not differ heavily with a larger participant group. Statistical distribution: 62% male, 38% female, Age range: 50% between 20-29, 25% between 30-39, 25% between 40-49, Qualification: 50% graduate student, 50% researchers, Expert level: 50% Advanced knowledge, 25% intermediate knowledge, 25% novice.

All of the participants stated to be basically familiar with tradition web form user interfaces and able to provide appropriate literal meta data such as a prename, surname, organization and email correctly. One user filled out the *country* data not correctly. Interestingly, the 88% of all participants simply ignored the optional ORCID input field where it was possible to provide a reference for a persistent author identifier, only 1 out of 8 users was able to provide a correct value for it.

URI-AWARE USER INPUT INTERFACES FOR THE UNOBTRUSIVE REFERENCE TO LINKED DATA

The subjective rating of the statement if *they are confident what to enter as a correct value in such an URI input field* also confirmed this. When requesting them to explicitly provide a Linked Data URI for an *author* ahead in a second form, all participants were able to perform the Retrieve and Paste operation (one participant only provided the ID and not a URI). The action step itself was assessed as quite easy, however, 50% argued and agreed that *it does not feel natural to enter URIs for specifying person data*. After improving the web form with a URI-aware user input behavior for an author name, where the ORCID was fetched and exposed semi-automatically, the provision rate for a provided correct author entity URI increased to 88%, which is a substantial improvement in comparison to the traditional web form input with a response rate of only 12%. Interestingly, users nevertheless assessed it only neutral that it *would be more convenient to not getting asked for a URI input*.

Similar results were obtained in the second, *keyword-oriented* test scenario. On average, a participant provided *four* keywords in the traditional, literal based input form. Spelling, word combinations and acronyms varied widely. When entering explicit category URIs in the second test, 38% were not able or willing to provide this information in a correct way at all, the others participants on average only provided *two* entity URIs from *Wikidata* or *DBpedia*. 72% disagreed that *it is easy and familiar to provide explicit URIs for keywords*, instead they *find it more convenient to work with literal text inputs*. Then in the third test, the web form with implicit URI-retrieval in the background showed again that participants provided the same amount of tags as in the traditional case, this time with corresponding correct URIs, although the subjective benefit of the suggestion was only rated as neutral on average.

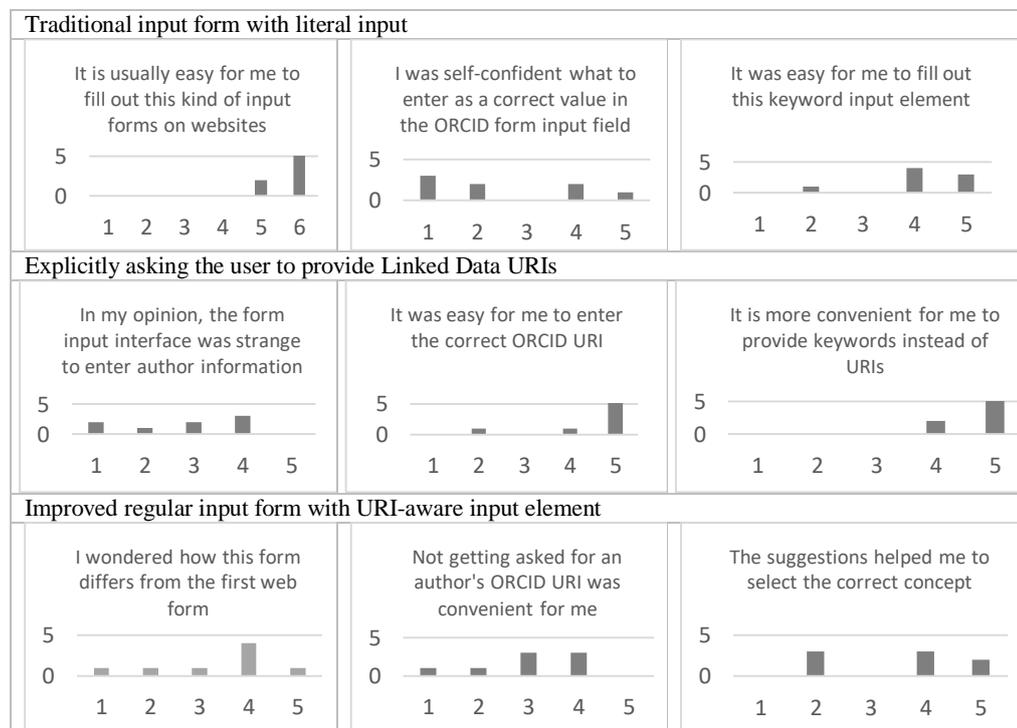


Figure 7. Result overview for a set of related questions from our conducted user survey

5. RELATED WORK

In the past, the management of Linked Data primarily concentrated on backend aspects by building knowledge graphs and storing semantic data in a graph database. The information for these knowledge graphs was assumed to be either published in an explicit way (Heath & Bizer, 2011), derived from existing text data through Entity recognition (Usbeck et al., 2014), or extracted with NLP or AI technologies (Rizzo, Troncy, Hellmann, & Bruemmer, 2012). Frontend aspects focused on the visualization of Linked Data as knowledge graph (Valentina Ivanova, Patrick Lambrix, Steffen Lohmann, & Catia Pesquita, 2017) or dealt with ways to embed structured data in HTML (Microformats or W3Cs' RDFa) in a uniform, extractable way (Bizer et al., 2013) by providing well-known attributes in HTML tags. In HTML5, dedicated attributes as well as generic data-* attributes were standardized (Bostock, Ogievetsky, & Heer, 2011). Input interfaces can obviously also be annotated with these attributes ("hcard-input-examples Microformats Wiki," 2017), yet, the collection of Linked Data references involves also other frontend considerations. Although frontend frameworks like SemanticUI exist, their understanding of semantics focusses primarily on self-explaining element descriptors as class names and less on standardized frontend elements that allow the input and reference of Linked Data entities. Projects like LD-R: Linked Data reactor (Khalili & Graaf, 2017) makes a step in the direction to standardized frontend Flux components to view and edit Linked Data, but favor the presentation of Linked Data instead describing components for input operation. We are convinced that also input interfaces have to be improved, because not all relevant data can automatically be extracted from existing data sources and requires human interaction, at least in a data curation step. Our contribution and LD-R can therefore benefit from each other. Using auto-suggestion for keyword mappings was already suggested by other authors ((Latif, Afzal, Hoefler, Saeed, & Tochtermann, 2009) in a case study, but their approach had a narrower context by only focusing on keywords and concentrated on the construction of SPARQL queries in the backend. Instead, our presented follows a human-centered approach and focuses on assisting the user as the appropriate actor to explicitly enter and distinguish the information that was actually meant to build knowledge graphs with correct data.

6. CONCLUSION AND FUTURE WORK

In this article, we presented URI-aware user input interfaces, that allow the management of data labels together with corresponding Linked Data entity URIs as persistent identifiers in the frontend of semantic-aware web applications in a transparent and inconspicuous way. After discussing characteristics of URI-aware user input interfaces, we described the realization as standardized WebComponents. We implemented a proof-of-concept for URI-aware user input interfaces and used it in a document submission scenario in combination with a user study. The evaluation confirmed, that the provision rate for Linked Data URIs can increase dramatically if users do not have to deal with explicit URIs/PIDs on their own in cases where it is not entirely necessary, even if they are unaware that they currently interact with Linked Data. Our contribution has a relevant impact on the development of new data-driven web applications, as Frontend-related input aspects for Linked Data have not sufficiently been discussed before. We showed that humans are unwilling to working with persistent identifiers if they do not have to, and accurately crafted assisting user interfaces can improve this situation.

In the future, the user experience for our presented solution can still be enhanced. Users reported that typeahead functionalities in combination with contextual constraints on the

suggested data can further improve the UIX by assuring an uninterrupted input operation. Another challenge is the application of these findings to alternative data input interfaces such as voice or gesture on mobile devices.

ACKNOWLEDGEMENT

This work was supported by the grant from the German Federal Ministry of Education and Research (BMBF) for the LEDS Project under grant agreement No 03WKCG11D.

REFERENCES

- Bizer, C., Eckert, K., Meusel, R., Mühleisen, H., Schuhmacher, M., & Völker, J. (2013). Deployment of RDFa, microdata, and microformats on the web - A quantitative analysis. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 8219 LNCS, pp. 17–32). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-41338-4_2
- Bostock, M., Ogievetsky, V., & Heer, J. (2011). D³ Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12), 2301–2309. <https://doi.org/10.1109/TVCG.2011.185>
- hcard-input-examples · Microformats Wiki. (2017). Retrieved July 25, 2018, from <http://microformats.org/wiki/hcard-auto-fill-examples>
- Heath, T., & Bizer, C. (2011). Linked Data: Evolving the Web into a Global Data Space. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 1(1), 1–136. <https://doi.org/10.2200/S00334ED1V01Y201102WBE001>
- Khalili, A., & Graaf, K. A. de. (2017). Linked Data Reactor: Towards Data-aware User Interfaces. In *Semantics2017 Proceedings*. Amsterdam. Retrieved from http://research.lid-r.org/papers/Semantics_2017_dataAwareUIs.pdf
- Langer, A., Göpfert, C., & Gaedke, M. (2018). F.I.E.L.D.S. – Analyzing form input interfaces for explicit Linked Data handling in document submission systems. In *Proceedings of 17th International Conference WWW/Internet (ICWI2018)*. Budapest. Retrieved from <http://www.iadisportal.org/digital-library/fields-analyzing-form-input-interfaces-for-explicit-linked-data-handling-in-document-submission-systems>
- Latif, A., Afzal, M. T., Hoefler, P., Saeed, A. U., & Tochtermann, K. (2009). Turning keywords into URIs: simplified user interfaces for exploring linked data. *ACM International Conference Proceeding Series; Vol. 403*, 76–81. <https://doi.org/10.1145/1655925.1655939>
- Nielsen, J., & Landauer, T. K. (1993). A mathematical model of the finding of usability problems. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '93* (pp. 206–213). New York, New York, USA: ACM Press. <https://doi.org/10.1145/169059.169166>
- Rizzo, G., Troncy, R., Hellmann, S., & Bruemmer, M. (2012). NERD meets NIF: Lifting NLP extraction results to the linked data cloud. In *CEUR Workshop Proceedings* (Vol. 937). Retrieved from http://nerd.eurecom.fr/ui/paper/Rizzo_Troncy_Hellmann_Bruemmer-ldow2012.pdf
- Shadbolt, N., Berners-Lee, T., & Hall, W. (2006). The Semantic Web - The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3), 96. <https://doi.org/10.1109/MIS.2006.62>
- Usbeck, R., Ngonga Ngomo, A.-C., Röder, M., Gerber, D., Coelho, S. A., Auer, S., & Both, A. (2014). AGDISTIS - Graph-Based Disambiguation of Named Entities Using Linked Data (pp. 457–471). Springer, Cham. https://doi.org/10.1007/978-3-319-11964-9_29
- Valentina Ivanova, Patrick Lambrix, Steffen Lohmann, & Catia Pesquita. (2017). CEUR-WS.org/Vol-1947 - Visualization and Interaction for Ontologies and Linked Data (VOILA! 2017). In *VOILA*. Vienna: CEUR. Retrieved from <http://ceur-ws.org/Vol-1947/>