Releasing multimodal data as Linguistic Linked Open Data: An experience report

Peter Menke
SFB 673, Project X1
University of Bielefeld

John McCrae
CIT-EC
University of Bielefeld

Philipp Cimiano
SFB 673, CIT-EC
University of Bielefeld

pmenke@techfak.uni-bielefeld.de
{jmccrae,cimiano}@cit-ec.uni-bielefeld.de

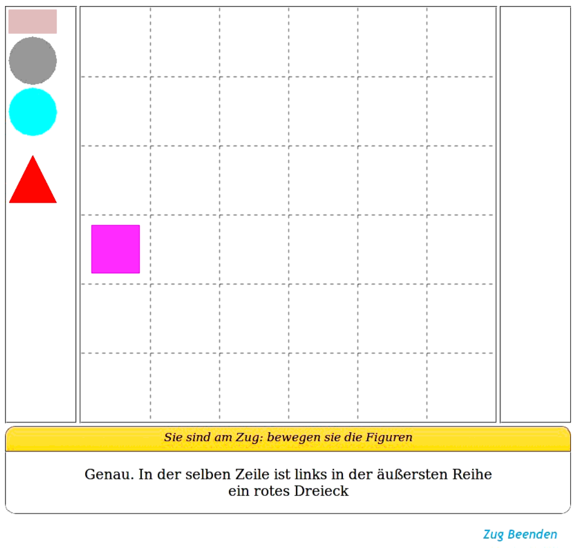
Abstract

In this paper we describe an implemented framework for releasing multimodal corpora as Linked Data. In particular, we describe our experiences in releasing a multimodal corpus based on an online chat game as Linked Data. Building on an internal multimodal data model we call FiESTA, we have implemented a library that enhances existing libraries and classes by functionality allowing to convert the data to RDF. Our framework is implemented on the Rails web application framework. We argue that this work can be highly useful for further contributions to the Linked Data community, especially from the fields of spoken dialogue and multimodal communication.

1 Introduction

In recent years, many linguistic resources have been released as Linked Data (Chiarcos et al., 2011). Most of the datasets that are part of the so called Linguistic Linked Open Data (LLOD) cloud consist of dictionaries, written corpora or lexica. However, multimodal dataset are currently heavily underrepresented. In order to address this gap, we describe a framework supporting the easy publication of multimodal data as RDF / Linked Data which is based on an existing multimodal data model and on the Rails framework. In this paper we describe our approach and summarize our experiences. In particular, we describe our experiences in releasing a multimodal corpus based on an online chat game as Linked Data. The corpus consists of chats and related actions in an object arrangement game using a computer-mediated setting. It contains multiple forms of annotation, including primary material such as text transcripts and information about object movements as well as secondary analysis such as phrase structure analysis of the text. Due to the challenging nature of the data, in particular that it contains annotations on multiple timelines, we developed a new model for the representation of this data, which we call FiESTA.

In order to express both established and new data categories and properties, from linguistics as well as from nonlinguistic communication, we developed a new data category registry, which contains links to other resources in the LLOD cloud, in particular to the ISOcat data category repository (Windhouwer and Wright, 2012), but also serves as a place where categories from novel research fields (mainly multimodal communication) can be collected, discussed, until they have settled down and are stable enough for an integration into more authoritative category registries, such as ISOcat. By means of this we aim to make the re-



Genau. In der selben Zeile ist links in der äußersten Reihe ein rotes Dreieck

Figure 1: A screen configuration as seen by the *slider*, who can see the last chat message (bottom part) and move objects with a mouse. Unused objects are stored in an area on the left.

source more widely available and to enable a long and successful lifecycle for the resource.

Furthermore, we describe a software toolchain for easy extraction of RDF data from existing information structures, such as classes or database records, and delivery of this data via web applications and services based on the popular framework Rails (Ruby et al., 2011). This tool chain is designed to be easy to integrate with existing libraries in a plugin-like fashion, in order to reduce the effort of integrating existing systems into Linked Data networks and infrastructures.

In Section 2 we describe the data collection, its provenance, its experimental setup and its levels of annotations. Then, Section 3 summarizes the steps from the internal representation of this (and other) multimodal data collections to a RDF representation served to the public web via HTTP. Some thoughts and prospects on how this system could be improved and distributed conclude the article.

2 The chat game experiment

2.1 About the chat game corpus

As a pilot test for the generation of RDF data in a large linguistic research project we selected a corpus resulting from a chat game experiment. This choice was motivated by several reasons:

1. The data set is compact and manageable, yet it contains data types and structures (e.g., multimodal and nonlinguistic interaction) that are still underrepresented in the Linked Data context.

2. It is heterogeneous, containing both language data and representations of actions and spatial entities.

3. The consent forms of the experiment contained clauses that permit a publication of the complete anonymized data sets. Without such explicit permissions, the publication even of anonymized derived data sets (such as transcriptions and annotations) is highly problematic especially in Germany. The chat game corpus is one of the few data sets with unproblematic consent forms. In addition, no video and audio recordings were created in this study, which regularly cause further problems considering anonymisation and protection of privacy for participants.

2.2 Participants and setup

28 adults (all native speakers of German) participated in pairs in the study (20 female, 8 male, mean age: 26). Data from several additional participants needed to be excluded due to various reasons. The players received course credit and/or a payment for their participation.

The chat game setup involves an object arrangement game paradigm with two players realised by a computer-mediated situation. Each participant sits at a computer terminal. The first participant (called the "chatter") has to describe target positions of objects on her screen with distinct colors and shapes to the second participant (the "slider") via chat messages. This second participant does not have access to the target configuration, resulting in the chatter's messages being the slider's only input. The slider is also not able to send messages. Their only mode of interaction is to move the game pieces onto the board, and into the correct positions.

The goal of the game is to reach the full target configuration of all objects by the technique described above. In each trial, eight rounds were played, with role switches between rounds.

2.3 Data structures

Primary data[1] essentially consists of an electronic log file of the activities performed by the participants. In particular, two types of actions were used: *chat messages* (including a time stamp and a string containing the message), and *movements of objects* (including a time stamp, an identifier of the object, and two pairs of coordinates, indicating the origin and the destination position on the board). The log file uses a custom XML format suited to the needs of the game (cf. Figure 2).

For each round, additional information about the respective target configuration was added to the log. A header contains further information about participants and a timestamp indicating the begin of the current trial.

Based on this automatically generated data, several annotations have been created:

[1]Terms like *primary* and *secondary data* are problematic when we go beyond classical face-to-face dialogues preserved in audio and video recordings. We use these terms in Lehmann's reading: "Primary linguistic data are [...] representations of [...] speech events with their spatio-temporal coordinates" (Lehmann, 2005, p. 187). However, his distinction between raw (=non-symbolic) and processed (=symbolic) data (Lehmann, 2005, pp. 205ff.) does not work for the data described here, because our raw data is in fact symbolic.

```
<match startTime="16.11.11 11:22">
  <round timeStarted="16.11.11 11:22" roundId="1">
    <chat time="+105" message="grauer kreis linke haelfte obere haelfte">
      <sentence value="fragment w/o verb" type="instruction" lok="spatial" id="s1">
        <parsetree id="parsetree1" tiefe="2" verzweigung="3.0" hoeflichkeit="2">
          <CNP>
            <NP>
              <ADJA lemma="grau">Grauer</ADJA>
              <NN lemma="Kreis">Kreis</NN>
            </NP>
            ...
          </CNP>
        </parsetree>
      </sentence>
    </chat>
    <move shape="gray_circle" from="-1,0" to="215,215" time="+133"/>
    <move shape="gray_circle" from="215,215" to="215,15" time="+136"/>
    ...
  </round>
  ...
</match>
```

Figure 2: A simplified example of the custom XML file format, containing one instruction and two subsequent moves (the second one being a correction).

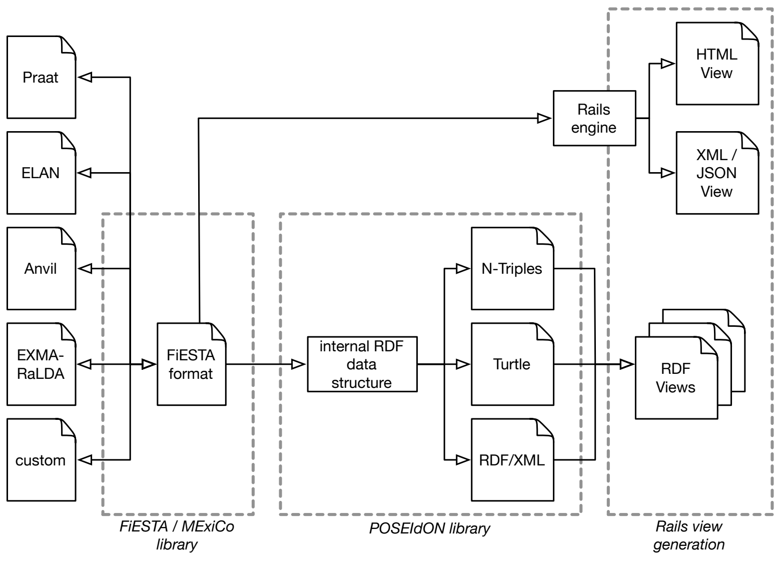


Figure 3: Architecture of the corpus management web application, grouped into scopes of responsibility of the respective libraries (FiESTA and POSEIdON).

1. A transformation of the written messages into orthographically and syntactically correct utterances. This was necessary for the parser (see below) to perform with an adequate accuracy.

2. Utterances were segmented into sentences and then parsed with the Stanford Parser (Klein and Manning, 2002; Klein and Manning, 2003), using the German version trained on the Negra corpus (Rafferty and Manning, 2008).

3. Syntactic and semantic properties of sentences were annotated, among them elaborateness (e.g., fragments and full sentences), speech acts (e.g., greetings, instructions, corrections, feedback) and localisation strategies – for instance, whether positions were described in relation to present objects ("to the right of the circle"), by describing absolute locations of the board itself ("into the bottom-left corner"), or by using metaphors (such as points of the compass, floors of buildings for rows: "south of the circle").

4. The parse trees were further annotated with basic tree measures (depth, breadth), and with an automatically generated quantitative measure of politeness, based on the occur-

rence of certain keywords, sentence types, and syntactic features.

Two annotators annotated the data. Some game instances were annotated by one of the annotators only, some by both of them. Differences were discussed with the experimenters, which lead to repeated correction and refinement of both annotations and annotation guidelines. This additional data was added to the XML files, as additional attributes or descendant elements to those already generated during experimentation.

Overall, the corpus contains 666 chat messages and 1,243 object moves. The parser created a total of 11,812 constituents (including terminal nodes) from the orthographically corrected chat messages (resulting in a total average of 17.75 constituent nodes per chat message).

3 From internal representations to RDF

3.1 Internal representation

We developed FiESTA (an acronym for "**f**ormat for **e**xtensive **s**patio**t**emporal **a**nnotations"), which takes into account various approaches, among them, the annotation graph approach (Bird and Liberman, 2001), the NITE object model (Evert et al., 2003), the speech transcription facilities of the TEI P5 specification (TEI Consortium, 2008), and the (X)CES standard (Ide et al., 2000). There were shortcomings in all these approaches that made it very difficult to express complex multimodal data structures. These shortcomings can also be found in theories and models that are more established in the Linked Data community, such as POWLA (Chiarcos, 2012) or LAF (Ide et al., 2003).

One of the most pressing problems is the restriction to a *single, flat stream or sequence of primary data* (called "text" in some approaches), or a *single, flat timeline*. In several data collections we need to support *multiple timelines*, especially in cases where multiple novel recording and tracking devices are used whose temporal synchronisation is nontrivial (because of irregular tracking intervals, computational delay, etc.). However, when working in a project with a limited duration, researchers are under time pressure, as a consequence, it can become necessary to perform analyses of data sets even before a working mechanism for complete, error-free synchronisation has been built by others. As an example, annotators might want to start the time-consuming transcription of speech as soon as possible, while others might make efforts to perform a categorization of automatically detected head gestures based on raw data generated by a novel tracker device. If it turns out that the time stamps in the tracker data are erroneous and cannot be aligned to the other ones using a simple linear transformation, there might be not enough time for their correction *before* annotators can start creating secondary data. Therefore, both groups need to start their work using their respective, isolated timelines if they do not want to put the project at risk. Simultaneously, the timeline of the tracking data must be aligned to that of the transcriptions in the background without modifying either of them.

The result are data sets that are based on different sets of time stamps, but belong to the same situation under investigation. A synchronisation of those different time stamps should be optional, and the original time stamps must be preserved as primary reference points at all times, even when a complete synchronisation can be achieved. With most of the given models, such an undertaking is either impossible, or it involves the alienation of model components (e.g., creation of phantom annotations being used as fake time points), which both inflates the resulting data structure and makes it less comprehensible. For instance, the annotation tool EXMARaLDA provides a mechanism for creating *time forks* (Schmidt and Wörner, 2005), but this is useful only for shorter stretches of simultaneous events surrounded by synchronised time points (e. g., for shorter segments of simultaneous speech), and not for timelines that might be completely independent from each other in the beginning and need to be merged and aligned later. Also, there are various potential reasons in a scientific workflow that call for the use of an annotation tool different from EXMARaLDA.

Also, in some cases there is need for the expression of spatial information parallel to temporal information. While this could be done by adding additional tiers with annotations, we consider it a cleaner and more logical solution to provide support for spatial (and other) axes on the same structural level as for timelines. This entails a modification of the present concept of the timeline towards a more general *scale* that also enables users to create spatial and abstract axes to which events and annotations can be aligned. There can be one or multiple scales, and each scale is given a unit, a dimension (e. g., time, or a spatial axis),

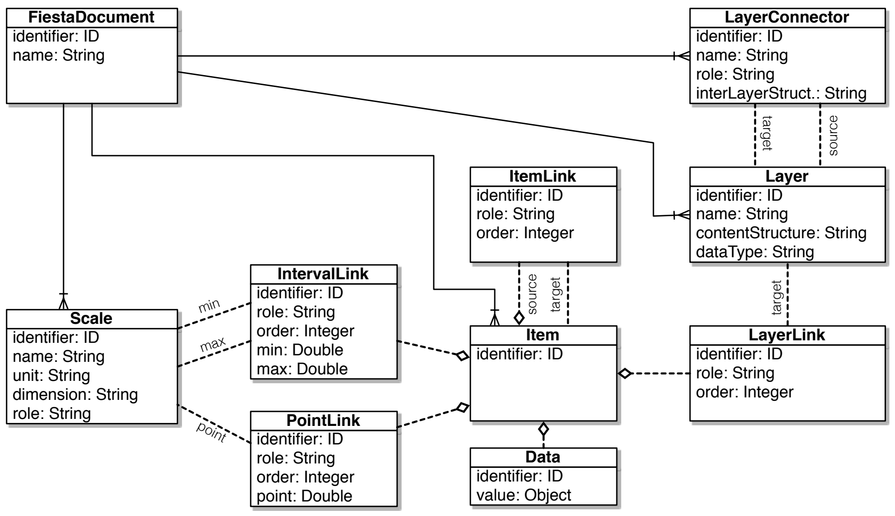


Figure 4: UML class diagram (simplified) of the FiESTA data model.

and a different level of measurement, following (Stevens, 1946). Scales can be left independent, or a synchronisation betweeen them can be expressed (e. g., a linear transformation between a video-frame-based scale and a millisecond-based one, or a manual alignment using explicit alignment points). A simplified version of the scale, and the other FiESTA classes and their relations is shown in the UML class diagram in Figure 4.

For the chat game data, three scales are used, one as a classic timeline, and two as a basis for coordinates on the two-dimensional game board. Chat messages, moves, and subsequent data sets are then imported as annotation items that are linked to points on these scales, and, in some cases, with a reference to other items.

3.2 A simple category registry

We established a simple web application serving as a minimalist concept registry. There, we collect and discuss concepts and categories for our data models as well as the multimodal phenomena that are (or are to be) modelled and described at our institution.[2]

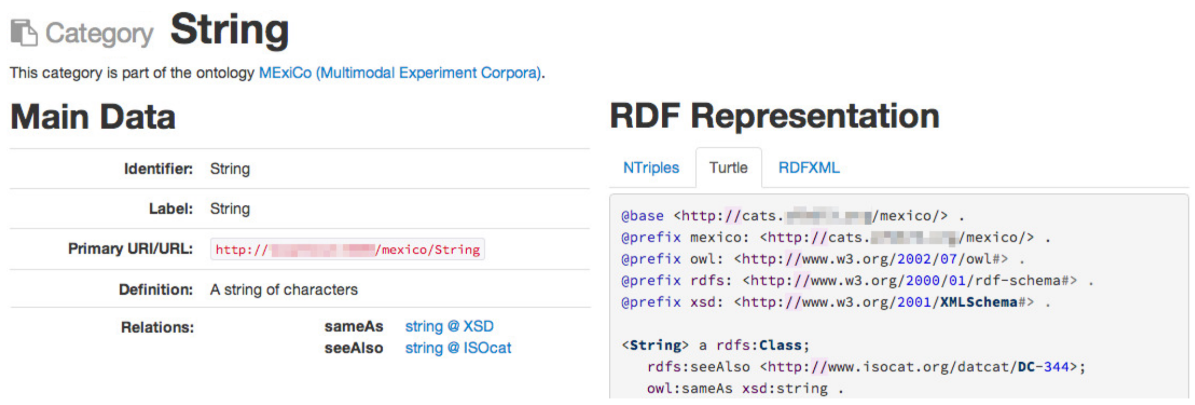
The granularity of the modeling of these concepts (and also of properties) is roughly on the level of the components used in RDF Schema. A *category* consists of *(1)* an *identifier* (which automatically is suffixed to the ontology URI to create an URI for the category), *(2)* a human-readable *label*, *(3)* a human-readable *definition* (typically consisting of one or two sentences), *(4)* information about the *class hierarchy*, *(5)* information about possible *domains* and *ranges*, and *(6)* a number of *relations*, which express equivalence and similarity relations to other categories already existing outside the system (using appropriate vocabulary, such as `rdfs:seeAlso` or `owl:sameAs`).

We added some convenience methods for easy linking to some vocabularies or concept registries, among them, ISOcat (Windhouwer and Wright, 2012), XML Schema, Dublin Core, FOAF, and others.

At the moment, the ontology describing the FiESTA data model (cf. Subsection 3.1) contains 23 categories and 19 properties, resulting in 148 triples. The main part of which uses terms from the RDFS vocabulary for a description and definition of classes and properties. Links to appropriate ISOcat entries were created, as well as to the structuring components in the POWLA ontology. However, most of these links use a weak `rdfs:seeAlso` predicate rather than asserting a strict equivalence, mainly because of slightly deviating definitions, or because of different domain or range specifications.

At the moment, the main purpose of this concept registry is to provide an URL for each concept, and to serve a snippet of information when an HTTP request is sent to such an URL. Depending

[2]This registry is not meant to be a replacement for established solutions such as ISOcat, but rather as an antecedent tool for very early collection and discussion of concepts and terms within projects and groups. We believe that this tool, including additional mechanisms such as discussion boards, is a better place for early concept development. As soon as the first results emerge, categories can be transferred to systems such as ISOcat for presentation and discussion.


Category **String**

This category is part of the ontology MExiCo (Multimodal Experiment Corpora).

Main Data

Identifier:	String
Label:	String
Primary URI/URL:	`http://[redacted]/mexico/String`
Definition:	A string of characters
Relations:	**sameAs** string @ XSD **seeAlso** string @ ISOcat

RDF Representation

NTriples Turtle RDFXML

```
@base <http://cats.[redacted]/mexico/> .
@prefix mexico: <http://cats.[redacted]/mexico/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<String> a rdfs:Class;
   rdfs:seeAlso <http://www.isocat.org/datcat/DC-344>;
   owl:sameAs xsd:string .
```

Figure 5: Screenshot of the simple category registry.

on the type of request, it delivers either a human-readable HTML document containing information about the concept (see Figure 5), or an RDF representation.

3.3 An RDF utility library

Within our systems, all transcription and annotation files are available in the pivotal representation format described above (see Section 3.1). They can be exported into all formats (a) for which an export routine is available and (b) that does not raise irresolvable format conversion errors. However, for the generation of RDF a different solution was chosen. We developed the POSEIdON library, containing modules that can be integrated into existing classes[3] in order to provide these classes and their instances with basic RDF information by using only a small set of configuring methods (see Figure 6 for an example of some POSEIdON directives and the resulting RDF). This can be useful if an existing library should be augmented with RDF information without modifying the existing source code.

For the representation of types and categories, the separate category registry described in 3.3 is used.

Typical use cases for POSEIdON directives are

- The definition of a URI for a class (used for type declarations of its instances).
- The definition of a URI scheme for instances of a class, based on a unique instance property.
- A mapping between instance variables and RDF snippets.
- Rules for a recursive RDF serialisation of member objects.

The low-level basis of POSEIdON is the established `rdf` library[4] which, in combination with various implementations of RDF writers, is used for collecting triples and exporting them to the respective variants of RDF documents. POSEIdON, by providing such a high-level interface, spares the user the creation and management of single RDF triples and graphs.

Several POSEIdON directives are added to the implementation of the FIESTA model. As a result, the RDF representation of a FiESTA document contains its complete contents represented as RDF triples (especially by using the recursive includes provided by POSEIdON).

There are already Ruby libraries that provide high-level support for RDF, such as the ActiveRDF library[5]. However, this library pursues a slightly different strategy by providing Ruby accessor methods to a data collection internally represented in RDF. In contrary, POSEIdON provides a simple way of getting an additional representation (in RDF) from an already existing library or data source in a *read-only* fashion, without modifying the source code of existing classes. Such data interfaces are typically based on XML documents or relational databases which are accessed with standard libraries (e.g., Nokogiri[6] for XML or ActiveRecord[7] for SQL databases). A modifi-

[3]We use Ruby's concept of *mixins*, which basically means the integration of source code contained in a module into an already existing class, without the need to alter the actual source code files of these classes.

[4]`https://github.com/ruby-rdf/rdf`.
[5]`http://activerdf.org/`
[6]`http://nokogiri.org`.
[7]`http://rubygems.org/gems/activerecord`

(a)

```
 1  class Scale
 2    include Poseidon
 3    self_uri 'http://cats.acme.org/Scale'
 4    rdf_property :identifier, 'http://cats.acme.org/identifier'
 5    rdf_property :name, 'http://cats.acme.org/name'
 6    rdf_property :unit, 'http://cats.acme.org/unit'
 7    rdf_property :dimension, 'http://cats.acme.org/dimension'
 8    rdf_property :mode, 'http://cats.acme.org/mode'
 9    ...
10  end
```

(b)

```
 1  @base <http://repo.acme.org/> .
 2  @prefix cats: <http://cats.acme.org/> .
 3  @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
 4
 5  cats:Scale a rdfs:Class .
 6
 7  <resources/1#timeline> a cats:Scale;
 8    cats:identifier "timeline";
 9    cats:name "Timeline";
10    cats:unit "s" ;
11    cats:dimension "time";
12    cats:mode "ratio".
```

Figure 6: Example of how POSEIdON works. *(a)* Usage of the POSEIdON library in a Ruby class to markup URIs (line 3) or to express rules for the export of instance properties (lines 4-8). — *(b)* The RDF resulting from these POSEIdON instructions. Some URLs are anonymized for review.

cation of such standard libraries just for an additional RDF representation would be out of proportion. POSEIdON's separate mixin strategy is a far cleaner approach.

3.4 Characteristics of the RDF representation

The resulting RDF representation (cf. the snippet in Figure 7) of the chat game corpus consists of approx. 300,000 triples (approx. 76,000 of these are data category annotations). A large number of those triples are necessary for the representation of the heavily interconnected phrase structure analyses of the chat messages. The category registry (cf. Subsection 3.2) is used for defining types of the entities contained in the corpus (as can be seen in the last lines of the code example in Figure 6b), where the predicates for the attributes come from the simple category registry described in Subsection 3.2.

3.5 Rails RDF integration

A web-based corpus management system is being developed in our project, which is based on Rails[8], a framework that uses the model-view-controller paradigm. In this system, RDF representations can easily be installed parallel to the standard HTML views and XML/JSON data representations by two rather simple steps:

[8] `http://rubyonrails.org/`

1. Model classes need to be augmented with POSEIdON directives,
2. and additional routes and controller actions need to be defined for the paths and objects for which RDF should be delivered.

RDF data can be obtained by content negotiation either by adding a corresponding file suffix to the URI (if omitted HTML is returned by default), or by setting an appropriate `Accept` field in the HTTP request header. The actual generation of the RDF representation is done entirely by the strategy described in the previous section. The corresponding Rails controller then retrieves the RDF representation generated by POSEIdON, and generates a HTTP response (with the appropriate metadata, such as the content type).

For larger resources, Rails' built-in caching mechanisms can be used to further reduce the response time, in addition to the basic caching implemented in POSEIdON.

4 Conclusion

In this article, we present two main contributions: a chat game corpus that is not easily expressable in terms of classic corpus and annotation models that require a flat sequence of primary data elements (timeline items or tokens); and a toolchain that obtains RDF representations from data sets by attaching a modular interface to existing libraries

```
<1> a cats:FiestaDocument;
  cats:hasItem <1#chat-5>,
    <1#round-1-move-7>,
    <1#round-1-move-8>;
  cats:hasLayer <1#chats>,
    <1#moves>,
    <1#sentences>,
    <1#parsedTrees>,
    <1#parsedPhrases>;
  cats:hasScale <1#timeline01>,
    <1#spatial_x>,
    <1#spatial_y>;
  cats:identifier "1" .

<1#timeline01> a cats:Scale;
  cats:identifier "timeline01";
  cats:name "Timeline";
  cats:unit "s" .

<1#moves> a cats:Layer;
  cats:identifier "moves";
  cats:name "Moves" .

<1#chat-5> a cats:Item;
  cats:hasData <1#chat-5-data>;
  cats:hasLayerLink <1#chat-5-layer>;
  cats:hasPointLink <1#chat-5-t>;
  cats:identifier "round-1-chat-5" .

<1#chat-5-data> a cats:Data;
  cats:stringValue "grauer kreis..." .

<1#chat-5-layer> a cats:LayerLink;
  cats:identifier "chat-5-layer";
  cats:target <1#chats> .

<1#chat-5-t> a cats:PointLink;
  cats:identifier "chat-5-t";
  cats:point 105,
  cats:target <1#timeline01> .
```

Figure 7: A snippet of the RDF representation generated by POSEIdON (corresponding to the chat message from Figure 2), with some context.

without modifying their actual source code. The principles of this toolchain have then been exemplified by taking the chat corpus data as a pilot data set. While our corpus and annotation data models have been under development for some years, the RDF publishing framework is still at an early stage. We believe that this data is a highly useful contributions to the linguistic and Linked Data community and that the resource is easier to use in a RDF form.

One of the more interesting aspects of the data is user-assigned data types, categories and structures used in singular annotation layers, especially when they go beyond the classic linguistic levels. While large vocabularies and ontologies for those have already been collected (for example, see the large number of syntactic and semantic concepts in ISOcat), there are hardly any entries for annotation schemes for gestures, eye movements, or other data coming from non-linguistic modalities. One of the main reasons is that morphosyntactic categories are far more established, and they are mainly agreed upon, at least to a degree sufficient for their integration into category registries. Research on non-linguistic modalities, on the other hand, is still at an early stage, and researchers have much more diverging sets of categories and definitions. As an example, the term *gesture* is used differently depending on the body limbs involved (especially, whether movements of the head and legs, knees, and feet should be subsumed unter this term, or whether there should be separate categories for them), so a premature nomenclature of categories based on only one of these definitions is not advisable.

Although the consequences of such novel research areas make it more difficult to create reliable concepts (and hence stable RDF), we are collaborating with researchers in these fields to collect first sets of categories for these modalities, which then are to be integrated into our category registry, and, when they are sufficiently agreed upon, also into the ISOcat system.

We believe that RDF-based representations especially of non-standard linguistic and multimodal resources (such as the chat game corpus, and other corpora, involving gestures, eye movements, and annotations of facial expressions) are a valuable gain for the Linguistic Linked Data community, even at such an early stage as described in this article.

The data set

The simple category registry can be browsed at `http://cats.sfb673.org`. The pilot data set of the chat game in the draft format as described above will be made available at `http://phoibos.sfb673.org/corpora/ChatStudy`. However, the data set as well as the tools and libraries described above are under active development, so the data set is subject to change during the next months until a stable status of all related systems and tools is achieved.

Acknowledgments

This research has been supported by the German Research Foundation (DFG) in the project X1 Multimodal Alignment Corpora: Statistical Modelling and Information Management of the Collaborative Research Centre (CRC) 673 Alignment in Communication at Bielefeld University, Germany.

References

Steven Bird and Mark Liberman. 2001. A formal framework for linguistic annotation. *Speech communication*, 33(1-2):23–60.

Christian Chiarcos, Sebastian Hellmann, and Sebastian Nordhoff. 2011. Towards a Linguistic Linked Open Data cloud: The Open Linguistics Working Group. *TAL*, 52(3):245–275.

Christian Chiarcos. 2012. Powla: Modeling linguistic corpora in OWL/DL. In Elena Simperl, Philipp Cimiano, Axel Polleres, Oscar Corcho, and Valentina Presutti, editors, *The Semantic Web: Research and Applications*, volume 7295 of *Lecture Notes in Computer Science*, pages 225–239. Springer, Berlin/Heidelberg.

Stefan Evert, Jean Carletta, Timothy J. O'Donnell, Jonathan Kilgour, Andreas Vögele, and Holger Voormann. 2003. The NITE Object Model. In *Proceedings of the EACL Workshop on Language Technology and the Semantic Web*, pages 1–17.

Nancy Ide, Patrice Bonhomme, and Laurent Romary. 2000. XCES : An XML-based Encoding Standard for Linguistic Corpora. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 825–830. ELRA.

Nancy Ide, Laurent Romary, and Eric de la Clergerie. 2003. International standard for a linguistic annotation framework. In *Proceedings of the HLT-NAACL 2003 workshop on Software engineering and architecture of language technology systems - Volume 8*, SEALTS '03, pages 25–30, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2002. Fast exact inference with a factored model for natural language parsing. *Advances in neural information processing systems*, 15:3—-10.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics ACL 03*, 1(July):423–430.

Christian Lehmann. 2005. Data in linguistics. *The Linguistic Review*, 21(3-4):175–210.

Anna N. Rafferty and Christopher D. Manning. 2008. Parsing three German treebanks: Lexicalized and unlexicalized baselines. *Proceedings of the Workshop on Parsing German*, pages 40–46.

Sam Ruby, Dave Thomas, and David Heinemeier Hansson. 2011. *Agile Web Development with Rails 3.2*. Pragmatic Bookshelf, 4th edition.

Thomas Schmidt and K Wörner. 2005. Erstellen und Analysieren von Gespr{ä}chskorpora mit EXMARaLDA. *Gespr{ä}chsforschung*, 6:171–195.

S. S. Stevens. 1946. On the Theory of Scales of Measurement. *Science*, 103(2684):677–680.

TEI Consortium. 2008. *TEI P5: Guidelines for electronic text encoding and interchange*. TEI Consortium.

Menzo Windhouwer and Sue Ellen Wright. 2012. Linking to Linguistic Data Categories in ISOcat. In Christian Chiarcos, Sebastian Nordhoff, and Sebastian Hellmann, editors, *Linked Data in Linguistics*, pages 99–107. Springer Berlin Heidelberg.