STS Infrastructural considerations

Christian Chiarcos
chiarcos@uni-potsdam.de
Infrastructure

• Requirements
• Candidates
  – standoff-based architecture (Stede et al. 2006, 2010)
  – UiMA (Ferrucci and Lally 2004)
  – RDF-based architecture (Hellmann 2010, Hellmann et al. 2012)
• Comparison
Requirements

• Flexibility
  – support all necessary data structures, hierarchical, and relational

• Interoperability
  – structural („syntactic“)
    • common exchange format for all modules
  – conceptual („semantic“)
    • well-defined data categories
    • clearly specified means to address them
Requirements

• Availability
  – Can we build upon an existing architecture?

• Web Services
  – Semantic modules using large knowledge bases should operate on their own servers

• Efficient interchange format
  – Easy to parse, merge and write

• Performance
1. Standoff-based architecture

- e.g., SuMMAR/MOTS (Stede et al. 2006, 2010)
  - pipeline architecture for high-quality text summarization
    - syntax, coreference, text structure, causal markers, etc.
  - standoff
    - output of different modules to be combined
    - these may also run in parallel
  - exchange format PAULA
    - standoff XML, derived from early (2004) drafts for the LAF
1. Architecture

flexible modules can be arranged in any order in the pipeline or be processed non-sequentially
⇒ standoff XML as common interchange format
1. Summarization pipeline

- Preprocessing Modules
  - Layout Structure and Metadata Extraction
  - Text Structure Extraction
  - Tokenization and Sentence Boundary Detection

- Flexible Modules
  - Coreference Analysis (Rosana)
  - Syntactic Analysis (Connexor)
  - Robust Morphosyntactic Analysis (TreeTagger)
  - Term Weight Calculation
  - Topic Segmentation

- Final Modules
  - Graphical Representation
  - Summary Calculation
  - Merging

Flexible Modules (selection)
1. A fragment

- Coreference Analysis (Rosana)
- Syntactic Analysis (Connexor)
- Transforming Rosana output to PAULA
- Merging multiple annotation layers in one PAULA project
- one single PAULA project comprising annotations from different modules
- Transforming relevant PAULA annotations to Connexor input format
1. Standoff XML

• advantages
  – modularization
  – trivial merge and split operations for annotations of the same document
    • add another file to the annotation project
  – clear conceptual separation of annotations

• disadvantages
  – modules exchange information through XML
    • relatively slow
2. UiMA (Ferruci and Lallas 2004)

- Unstructured Information Management Architecture
- Industry-scale architecture for NLP pipelines
  - active community, good support
- Relatively generic data model with different realizations
  - JAVA Objects, XML, others
2. UiMA

- Wrappers for various NLP tools available
- input and output representations of modules ("CAS consumers") defined by annotation types
  - e.g., a part-of-speech tag inventory
  - different annotation type systems may not be compatible with each other
  
  => limited interoperability
2. UiMA

- disadvantages
  - limited interoperability only
  - how to implement a distributed architecture?

- advantages
  - maturity
    - rich technological ecosystem, active community
  - efficiency
    - supports, e.g., information exchange through JAVA objects
2. UiMA extensions

- Egner et al. (2007)
  - UiMA Grid, distributed large-scale text analysis

- Verspoor et al. (2009)
  - Abstracting the types away from a UiMA type system
  - Ontologies instead of annotation types
    - improved conceptual (‘semantic’) interoperability
    - less efficient indexing

- These extensions would have to be reimplemented for an STS pipeline
  - AFAIK, not publicly available
3. RDF-based architecture

- Hellmann (2010), Hellmann et al. (2012)
  - NLP Interchange Format (NIF)
    - http://nlp2rdf.org/nif-1-0
  - NLP2RDF: RDF wrappers for various tools
    - http://nlp2rdf.org
    - provides NLP analyses for processing with Semantic Web tools
  - applied in a large-scale European research project (LOD2)
    - adopted by several external research groups
3. RDF

• Resource Description Framework
  – W3C standard
  – formalizes labeled directed multigraphs
    (like XML standoff formats)
  – sublanguages define specialized vocabularies
    • RDF Schema: concept hierarchies
    • SKOS: semi-structured terminology bases
    • OWL: ontologies
3. RDF

• different linearizations
  – XML (verbose), Turtle (compact), others

• rich technological ecosystem
  – data bases („triple stores“)
  – APIs and (syntactic) validators
  – query language SPARQL

• OWL/DL
  – description logics
  – defining and checking constraints (axioms)
    => formally defined user-specific data types
3. NLP2RDF

NLP Tools & Services

Structural Interoperability
- URI Scheme
  - Offset
  - Context-hash
  - Content-specific
- String Ontology
- Structured Sentence Ontology

Conceptual Interoperability
- Ontologies
  - OLiA, NERD, SCMS, Topic, Opinion / Sentiment
- Language Resources
  - Powla, Wiktionary, Lemon, Wordnet, DBpedia

Access Interoperability
- REST Interface
  - HTTP
  - Stateless
  - NIF RDF result format
- Triple Store
  - SPARQL
- OWL/DL Reasoner

Cross-Linking
Background Knowledge
Query federation

Data Web
3. RDF

• advantages
  – rich ecosystem, large and active community
  – native support for distributed processing
  – direct integration with LOD resources
    • may be relevant for STS
  – conceptual interoperability through linking with terminology repositories
<table>
<thead>
<tr>
<th></th>
<th>standoff XML</th>
<th>UiMA</th>
<th>NLP2RDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>flexibility</td>
<td>+</td>
<td>(+)</td>
<td>+</td>
</tr>
</tbody>
</table>

flexibility:
+ support for all necessary data structures
(+) UiMA: multiple ways to represent trees
Comparison

<table>
<thead>
<tr>
<th></th>
<th>standoff XML</th>
<th>UiMA</th>
<th>NLP2RDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>flexibility</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>structural interoperability</td>
<td>+</td>
<td>(+)</td>
<td>+</td>
</tr>
</tbody>
</table>

structural („syntactic“) interoperability:
+ same format for all modules
(+) UiMA: multiple ways to define trees
Comparison

<table>
<thead>
<tr>
<th></th>
<th>standoff XML</th>
<th>UiMA</th>
<th>NLP2RDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>flexibility</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>structural interoperability</td>
<td>+</td>
<td>(+)</td>
<td>+</td>
</tr>
<tr>
<td>conceptual interoperability</td>
<td>(-)</td>
<td>(+)</td>
<td>+</td>
</tr>
</tbody>
</table>

Conceptual ("semantic") interoperability:
+ Interoperability through reference to a terminology repository
(+) UiMA: interoperability if the same annotation type system is used
(-) standoff: links to terminology repositories can be provided, but no standard has been established to do so
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>standoff XML</th>
<th>UiMA</th>
<th>NLP2RDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>flexibility</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>structural</td>
<td>+</td>
<td>(+)</td>
<td>+</td>
</tr>
<tr>
<td>interoperability</td>
<td>(+)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>conceptual</td>
<td>(-)</td>
<td>(+)</td>
<td>+</td>
</tr>
<tr>
<td>interoperability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>availability</td>
<td>- (SuMMAR)</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

### availability:
- unknown/restricted licence
+ open license
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>standoff XML</th>
<th>UiMA</th>
<th>NLP2RDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>flexibility</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>structural interoperability</td>
<td>+</td>
<td>(+)</td>
<td>+</td>
</tr>
<tr>
<td>conceptual interoperability</td>
<td>(-)</td>
<td>(+)</td>
<td>+</td>
</tr>
<tr>
<td>availability</td>
<td>- (SuMMAR)</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>maturity</td>
<td>(-)</td>
<td>++</td>
<td>+</td>
</tr>
</tbody>
</table>

**maturity:**

- ++ industry-scale
- + used in multiple research groups
- (-) used in one research group
## Comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>standoff XML</th>
<th>UiMA</th>
<th>NLP2RDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>flexibility</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>structural interoperability</td>
<td>+</td>
<td>(+)</td>
<td>+</td>
</tr>
<tr>
<td>conceptual interoperability</td>
<td>(-)</td>
<td>(+)</td>
<td>+</td>
</tr>
<tr>
<td>availability</td>
<td>- (SuMMAR)</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>maturity</td>
<td>(-)</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>web services</td>
<td>(+)</td>
<td>(+)</td>
<td>+</td>
</tr>
</tbody>
</table>

**support for distributed processing (web services):**

+ available

(+) possible
Comparison

<table>
<thead>
<tr>
<th></th>
<th>standoff XML</th>
<th>UiMA</th>
<th>NLP2RDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>flexibility</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>structural interoperability</td>
<td>+</td>
<td>(+)</td>
<td>+</td>
</tr>
<tr>
<td>conceptual interoperability</td>
<td>(-)</td>
<td>(+)</td>
<td>+</td>
</tr>
<tr>
<td>availability</td>
<td>- (SuMMAR)</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>maturity</td>
<td>(-)</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>web services</td>
<td>(+)</td>
<td>(+)</td>
<td>+</td>
</tr>
<tr>
<td>performance/efficiency</td>
<td>-</td>
<td>+/(+)</td>
<td>(+)</td>
</tr>
</tbody>
</table>

- performance/efficiency
  + direct exchange of objects (without serialization) possible
  (+) compact serialization
  - verbose serialization
## Todo: Rank criteria

<table>
<thead>
<tr>
<th></th>
<th>standoff XML</th>
<th>UiMA</th>
<th>NLP2RDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>flexibility</td>
<td>+</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>structural interoperability</td>
<td>+</td>
<td>(+)</td>
<td>+</td>
</tr>
<tr>
<td>conceptual interoperability</td>
<td>(-)</td>
<td>(+)</td>
<td>+</td>
</tr>
<tr>
<td>availability</td>
<td>- (SuMMAR)</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>maturity</td>
<td>(-)</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>web services</td>
<td>(+)</td>
<td>(+)</td>
<td>+</td>
</tr>
<tr>
<td>performance/efficiency</td>
<td>-</td>
<td>+/(+)</td>
<td>(+)</td>
</tr>
</tbody>
</table>

Which to chose?
Combination of multiple architectures?