Dealing with multiple annotated data – format and application

Maik Stührenberg
Daniela Goecke
Nils Diewald
Daniel Jettka
Bielefeld University
Overview

● SGF – The Sekimo Generic Format
● Real World Application Scenarios
  • Analyzing anaphoric relations in the Sekimo project
  • Exchange format for the lexical chaining tool Scientific Workplace
  • Exchange format for the annotation tool Serengeti
● Conclusion
Overview

- SGF – The Sekimo Generic Format
- Real World Application Scenarios
  - Analyzing anaphoric relations in the Sekimo project
  - Exchange format for the lexical chaining tool Scientific Workplace
  - Exchange format for the annotation tool Serengeti
- Conclusion
SGF – The Sekimo Generic Format

Some considerations:
Some considerations:

- Multiple annotations (in separate files)

```xml
<s>
  <np>
    <pron>This</pron>
  </np>
  <vp>
    <v>is</v>
    <np>
      <det>a</det>
      <n>sentence</n>
    </np>
  </vp>
.</s>

<syllables>
  <syll>This</syll>
  <syll>is</syll>
  <syll>a</syll>
  <syll>sen</syll>
  <syll>ten</syll>
.</syllables>
```
Some considerations:

- Multiple annotations (in separate files)
- Primary data is stored redundantly

```xml
<s>
  <np>
    <pron>This</pron>
  </np>
  <vp>
    <v>is</v>
    <np>
      <det>a</det>
      <n>sentence</n>
    </np>
  </vp>
  ...
</s>
```

```xml
<syllables>
  <syll>This</syll>
  <syll>is</syll>
  <syll>a</syll>
  <syll>sen</syll>
  <syll>tence</syll>
  ...
</syllables>
```
Some considerations:

- Let's start with deleting this redundancy by using offsets...

```
This is a sentence.
```

00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19
SGF – The Sekimo Generic Format

Some considerations:

- Let's start with deleting this redundancy by using offsets...

```xml
<s start="0" end="19">
  <np start="0" end="4">
    <pron start="0" end="4"/>
  </np>
  <vp start="5" end="18">
    <v start="5" end="7"/>
    <np start="8" end="18">
      <det start="8" end="9"/>
      <n start="10" end="18"/>
    </np>
  </vp>
</s>
```

```xml
<syllables start="0" end="19">
  <syll start="0" end="4"/>
  <syll start="5" end="7"/>
  <syll start="8" end="9"/>
  <syll start="10" end="13"/>
  <syll start="13" end="18"/>
</syllables>
```
SGF – The Sekimo Generic Format

Some considerations:

- ...but there is still some redundancy left...

```
<s start="0" end="19">
  <np start="0" end="4">
    <pron start="0" end="4"/>
  </np>
  <vp start="5" end="18">
    <v start="5" end="7"/>
    <np start="8" end="18">
      <det start="8" end="9"/>
      <n start="10" end="18"/>
    </np>
  </vp>
</s>
```

```
<syllables start="0" end="19">
  <syll start="0" end="4"/>
  <syll start="5" end="7"/>
  <syll start="8" end="9"/>
  <syll start="10" end="13"/>
  <syll start="13" end="18"/>
</syllables>
```
SGF – The Sekimo Generic Format

Some considerations:

• getting closer...

<s segment="s0">
  <np segment="s0">
    <pron segment="s1"/>
  </np>
  <vp segment="s2">
    <v segment="s3"/>
    <np segment="s4">
      <det segment="s5"/>
      <n segment="s6"/>
    </np>
  </vp>
</s>

<segment id="s0" start="0" end="19"/>
<segment id="s1" start="0" end="4"/>
<segment id="s2" start="5" end="18"/>
<segment id="s3" start="5" end="7"/>
<segment id="s4" start="8" end="18"/>
<segment id="s5" start="8" end="9"/>
<segment id="s6" start="10" end="18"/>
<segment id="s7" start="10" end="13"/>
<segment id="s8" start="13" end="18"/>

<syllables segment="s0">
  <syll segment="s1"/>
  <syll segment="s3"/>
  <syll segment="s5"/>
  <syll segment="s7"/>
  <syll segment="s8"/>
</syllables>
Final step:

- Putting it all together in a single file
  - Use XML ID/IDREF for connecting segments of the primary data with the respective annotation(s)
- Eases import & export

```xml
<segment id="s0" start="0" end="19"/>
<segment id="s1" start="0" end="4"/>
<segment id="s2" start="5" end="18"/>
<segment id="s3" start="5" end="7"/>
<segment id="s4" start="8" end="18"/>
<segment id="s5" start="8" end="9"/>
<segment id="s6" start="10" end="18"/>
<segment id="s7" start="10" end="13"/>
<segment id="s8" start="13" end="18"/>
<s segment="s0">
  <np segment="s1">
    <pron segment="s1"/>
  </np>
  <vp segment="s2">
    <v segment="s3"/>
    <np segment="s4">
      <det segment="s5"/>
      <n segment="s6"/>
    </np>
  </vp>
</s>
<syllables segment="s0">
  <syll segment="s1"/>
  <syll segment="s3"/>
  <syll segment="s5"/>
  <syll segment="s7"/>
  <syll segment="s8"/>
</syllables>
```
Final step:

• Putting it all together in a single file
  
  • Use XML ID/IDREF for connecting segments of the primary data with the respective annotation(s)

• Eases import & export

```xml
<segment id="s0" start="0" end="19"/>
<segment id="s1" start="0" end="4"/>
<segment id="s2" start="5" end="18"/>
<segment id="s3" start="5" end="7"/>
<segment id="s4" start="8" end="18"/>
<segment id="s5" start="8" end="9"/>
<segment id="s6" start="10" end="18"/>
<segment id="s7" start="10" end="13"/>
<segment id="s8" start="13" end="18"/>
<s segment="s0">
  <np segment="s0">
    <pron segment="s1"/>
  </np>
  <vp segment="s2">
    <v segment="s3"/>
    <np segment="s4">
      <det segment="s5"/>
      <n segment="s6"/>
    </np>
  </vp>
</s>
</syllables segment="s0">
<syll segment="s1"/>
<syll segment="s3"/>
<syll segment="s5"/>
<syll segment="s7"/>
<syll segment="s8"/>
</syllables>
```
Final step:

- Putting it all together in a single file
  - Use XML ID/IDREF for connecting segments of the primary data with the respective annotation(s)
  - Eases import & export
- Reuse as much available standards and formats as possible – try not to reinvent the wheel
  - Existing metadata specifications
  - Logical document structure specifications
  - Global XML attributes such as xml:lang or xml:id

```xml
<segment id="s0" start="0" end="19"/>
<segment id="s1" start="0" end="4"/>
<segment id="s2" start="5" end="18"/>
<segment id="s3" start="5" end="7"/>
<segment id="s4" start="8" end="18"/>
<segment id="s5" start="8" end="9"/>
<segment id="s6" start="10" end="18"/>
<segment id="s7" start="10" end="13"/>
<segment id="s8" start="13" end="18"/>

<s segment="s0">
  <np segment="s0">
    <pron segment="s1"/>
  </np>
  <vp segment="s2">
    <v segment="s3"/>
    <np segment="s4">
      <det segment="s5"/>
      <n segment="s6"/>
    </np>
  </vp>
</s>

<syllables segment="s0">
  <syll segment="s1"/>
  <syll segment="s3"/>
  <syll segment="s5"/>
  <syll segment="s7"/>
  <syll segment="s8"/>
</syllables>
```
SGF – The Sekimo Generic Format

SGF is a logical descendant of the Prolog fact base format by Witt et al., 2005 – which itself is an multi layer markup extension of the format introduced by Sperberg-McQueen et al., 2000:

- It uses the Annotation Graph model, i.e. annotations are aligned via character positions (if textual primary data is used) or positions in time
- Multiple annotations can be saved in a single XML file, a native XML database, a relational database or a hybrid database
- In addition SGF can be used for storing whole corpora or single corpusData entries
- Standard (i.e. unmodified) XPath and XQuery is used for analyzing multiple annotations
SGF defines two types of layers:

- The base layer
- One or more annotation layer(s)
SGF – The base layer

The base layer contains:

- Primary data
  - Textual primary data can be referenced via the uri attribute of the location element or stored directly as element content of the textualContent child element of the primaryData element
  - Multimodal primary data is referenced via the uri attribute of the location element – multiple primary data is possible, however, there has to be one master primary data (depicted by the role attribute) providing the master time line

```xml
<sgf:corpus xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.text-technology.de/sekimo sgf.xsd"
  xmlns="http://www.text-technology.de/sekimo"
  xmlns:sgf="http://www.text-technology.de/sekimo">
  <sgf:corpusData id="c1" type="text">
    <sgf:primaryData start="0" end="1036" xml:lang="de">
      <sgf:location uri="ex1.txt" encoding="utf-8"/>
    </sgf:primaryData>
    <sgf:checksum algorithm="md5">15a244f76923</sgf:checksum>
  </sgf:corpusData>
</sgf:corpus>
```
SGF – The base layer

The base layer contains:

- An optional checksum for verifying primary data
- Optional metadata which is specified by an external document grammar (OLAC in our case) – metadata can be applied to the whole corpus, as child of the corpusData element or underneath an annotation level

```xml
    <sgf:corpusData id="c1" type="text">
        <sgf:primaryData start="0" end="1036" xml:lang="de">
            <sgf:location uri="ex1.txt" encoding="utf-8"/>
        </sgf:primaryData>
        <sgf:checksum algorithm="md5">15a244f76923</sgf:checksum>
        <sgf:meta>
            <olac:olac xmlns:olac="http://www.language-archives.org/OLAC/1.0/"
                        xmlns="http://purl.org/dc/elements/1.1/"
                        xmlns:dcterms="http://purl.org/dc/terms/"
                        xsi:schemaLocation="http://www.language-archives.org/OLAC/1.0/ olac.xsd">
                </olac:olac>
        </sgf:meta>
    </sgf:corpusData>
</sgf:corpus>
```
We start with the primary data file...
At first, the segments which delimit the annotated data are defined. Segments can be established by start and end positions...
SGF – Adding annotation layers

...or by referring to already defined segments
SGF – Adding annotation layers

Afterwards, an XML namespace and the annotation layer is added (if there is none in the original data – in case the SGF annotation level is the result of a conversion)
The added `sgf:segment` attribute connect segments and annotation elements. If a segment takes part in different annotation elements it will be defined once only.
For each annotation layer the following holds:

- Declaration is done external (an XSD must be present) – the original document grammar is converted according to the following rules:
  - Mixed content elements are converted to container elements
  - PCDATA elements are converted to empty elements
  - The `sgf:segment` attribute is added to former non-empty elements
- Hierarchy of elements remains intact, no need for additional files – as long as tree structures are adequate
- All attributes remain intact
SGF – Sekimo Generic Format

As already shown SGF allows for two ways of defining segments:

1. By providing start and end positions

   `<sgf:segment id="seg1" start="0" end="100"/>`

2. By referring to already defined segments (in this case the otherwise optional type attribute has to be set to the value 'seg')

   `<sgf:segment id="seg1" start="0" end="100"/>`
   `<sgf:segment id="seg2" start="200" end="250"/>`
   `<sgf:segment id="seg3" type="seg" segments="seg1 seg2"/>`

The mode attribute can be used to define continuous segments (default) or disjoint segments

   `<sgf:segment id="seg3" type="seg" segments="seg1 seg2" mode="continuous"/>`
   `<sgf:segment id="seg4" type="seg" segments="seg1 seg2" mode="disjoint"/>`
Some further remarks:

- A `sgf:corpusData` element contains at least one `sgf:primaryData` element and can store several `sgf:annotation` elements.
Some further remarks:

- A `sgf:corpusData` element contains at least one `sgf:primaryData` element and can store several `sgf:annotation` elements.

- A `sgf:annotation` element can consist of metadata and several `sgf:level` elements. This mechanism can be used for subsuming annotation layers.
SGF – Sekimo Generic Format

Some further remarks:

- A `sgf:level` element contains of optional metadata, a `sgf:layer` element and an optional `sgf:log` (later in this talk).

- The optional `priority` attribute can be used in case of virtually resulting overlapping annotation layers (when doing an inline markup unification) – the highest value determines the innermost annotation layer.

- The optional `import` attribute can be used for layers the actual layer relies on.

- More information regarding the `annotator` attribute later in this talk.
Some further remarks:

- A `sgf:layer` element serves as a wrapper for annotations of a different namespace
- Validation of annotation layers via XSD is mandatory
- Apart from XSD validation, embedded Schematron asserts are available as additional constraints
  - Waiting for XML Schema 1.1

```xml
<xsd:element name="layer">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
        processContents="strict"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

- Cross-grammar validation is possible in principle using Schematron/XSLT/XQuery – but would require a lot of work
SGF – Sekimo Generic Format

Some further remarks:

- There are two implementations for converting inline and standoff annotation into SGF (XSLT 2.0 and Java – shown below)

- Both implementations allow for converting a single inline annotation into an SGF file and merging multiple SGF files afterwards into a single SGF instance

- In addition the Java implementation allows the conversion of the underlying XSD
SGF – Sekimo Generic Format

Some further remarks:

- The examples shown before use the XML inherent tree structure – however, as only the base layer is pre-declared, representing graph structures is possible as well.

- The following DBMS were tested with test sets:
  - eXist, Qizx/db 2.0, Oracle Berkeley DB XML (native)
  - IBM DB2 9.x/9.5 (hybrid)
  - MySQL (relational)

- For evaluation purposes Saxon-SA on a per-file-base was used, evaluation results are available in Stührenberg & Goecke, 2008.
So...

...why did we develop SGF?
There is a variety of already established formats and specifications for dealing with complex document structures:

- The already mentioned Prolog fact base format
- Goddag/TexMECS
- LMNL
- Multi-colored trees
- XML delay nodes
- XConcur
- Graph based formats such as LAF/GrAF
SGF – Sekimo Generic Format

The short answer to this question:

- We wanted a format that is pure XML (without any non-standard extensions)
- We wanted the expressiveness of graph based formats without the disadvantages in terms of processing costs (compared to tree based formats)
- We wanted a format which can be used for analyzing annotated data, not only an interchange format
- We wanted an application-driven approach (since the time a project runs is limited)
Overview

- SGF – The Sekimo Generic Format
- Real World Application Scenarios
  - Analyzing anaphoric relations in the Sekimo project
  - Exchange format for the lexical chaining tool Scientific Workplace
  - Exchange format for the annotation tool Serengeti
- Conclusion
Analyzing anaphoric relations in the Sekimo project

A few words about the Sekimo project:

- SEKundäre InformationsMOdellierung
- Part of the distributed research group 437 „Text-Technological Modelling of Information“ funded by the Deutsche Forschungsgemeinschaft (DFG)
- Domain: Anaphoric relations, specifically the resolution of bridging phenomena between definite descriptions
Analyzing anaphoric relations in the Sekimo project

A few words about the application domain:

- Anaphora resolution is the process of identifying the antecedent(s) in the previous context.
- Anaphora resolution requires a variety of knowledge:
  - POS, morphology, grammatical function, semantic relatedness, distance, discourse structure among others.
- The anaphora resolution process consists of several steps:
  - Detection of anaphoric elements.
  - Creation of a candidate list for each anaphor.
  - Filtering of candidate list.
  - Identification of correct antecedent.
Analyzing anaphoric relations in the Sekimo project
Analyzing anaphoric relations in the Sekimo project

The logical document annotation
Analyzing anaphoric relations in the Sekimo project

The morpho-syntactic annotation (including dependency information)
Analyzing anaphoric relations in the Sekimo project

The discourse entity annotation
Analyzing anaphoric relations in the Sekimo project

Analyzing multiple annotated data with XQuery

- Extracting candidate lists containing possible antecedent candidates for anaphors
- Examining relations between elements derived from different annotation levels
  - Can antecedent candidates for anaphors be found embedded in specific parts of the logical document structure?
  - How long is the typical distance between anaphor and antecedent?
  - Are there elements of the logical document structure that influence the typical distance?
  - Are there differences in terms of anaphora-antecedent-distance between scientific articles and newspaper articles?
- ...

Dealing with multiple annotated data – format and application
Goddag Workshop, Amsterdam, 1.-5. December 2008
Stührenberg, Goecke, Diewald, Jettka
Overview

- SGF – The Sekimo Generic Format
- Real World Application Scenarios
  - Analyzing anaphoric relations in the Sekimo project
  - Exchange format for the lexical chaining tool *Scientific Workplace*
  - Exchange format for the annotation tool *Serengeti*
- Conclusion
Exchange format for the lexical chaining tool *Scientific Workplace*

A few words about the Indogram project:

- **INduction of web genre DOcument GRAMmars**
- Part of the distributed research group 437 „Text-Technological Modelling of Information“ funded by the Deutsche Forschungsgemeinschaft (DFG)
- Project aim: Automatic induction of probabilistic document grammars as models of web genres
- Project method (amongst others): The web based tool *Scientific Workplace/Scientific Desktop* for processing different types of texts
Tennis

Von Marcus Stenhausen

Der Wimbledon ist tot!

Im längsten Finale der Wimbldon - Geschichtschlägt Rafael Nadal - Dauer - Champion

Roger Federer - Die Nachbarin für den Tennis - haben sich endgültig umgekehrt. Eine Analyse

?? Das war das beste Match aller Zeiten ??, schwärme John McEnroe, wiederer Wimbledon - Sieger

und zwischen TV - Kommentator. Der Sieger, Rafael Nadal, sprach vom ???? größten, schönsten Moment

?? seines Lebens ??, In fünf Sätzen hatte er, der kräftige Malloriker den Dauer - Champion Roger Federer


Nadal hatte einen Bissere hingelaufen, sicherte sich die ersten beiden Sätze. Als es im dritten Satz eng wurde, kam

Federer, ein Regenwurf zu Hilfe, das Match wurde für eine Stunde unterbrochen. Danach glich Federer aus

Die Partie musste in den fünften Satz. Es regnete wieder, und um 21:15 Uhr britischer Zeit, die Sonne an der

Kurts Road war schon fast untergegangen, schlug Nadal zu seinem vierten in der Vorhand Matchball auf Federer, hatt die


Der Tennis pur ein unvergesslicher Moment - war unglaublich.

Ein neuer 

Tennis - Legende wirkte erleichtert. Sein Rekord von fünf aufeinanderfolgenden Wimbledon - Titeln bleibt

bestehen. Mit einem Sieg hatte Federer ihn überholen können.

Stattdessen siegte Nadal. Für die Nummer zwei der Welt ist es der größte Erfolg in einer perfekten Saison. Beim

Grand Slam in Paris holte er seinen vierten Titel in Serie. Im Londoner Queens Club gewann er sein erstes

Rasenturnier. In Wimbledon konnten er im Halbfinale ?

Doktor, Schütze ??

Tennis - Tomasota Themla Thronosaus Title

Uhrzeit von Verblüffung Vorfeld Welt Wimbledon York http www
SGF and the *Scientific Workplace/Scientific Desktop*:

- Together with other members of the research group an annotation format for storing lexical chains was established.
- This format is used as annotation layer as part of an SGF instance (called SGF-LC).
- In addition SGF-LC supports the storage of linguistic resources (either included in a single file with other SGF data or as reference to external resources).
<sgf:corpusData id="c1" type="text">
    [...]
    <sgf:segments>
        <sgf:segment id="i1" type="char" start="1" end="5"/>
        <sgf:segment id="i2" type="char" start="6" end="10"/>
    </sgf:segments>
    <sgf:annotation>
        <sgf:level xml:id="lc" resourcesUsed="gn wiki">
            <sgf:layer xmlns:lc="http://www.text-technology.de/lc">
                <lc:link from="i5" to="i1" relation="hyperonymy" relationDirection="l"
                    distance="0.848754899874578" distanceType="WordnetVektor-UMN"
                    resourceUsed="gn"/>
                <lc:link from="i5" to="i3" relation="transitive" relationDirection="r"
                    distance="-0.434" distanceType="NSS-Gwikipedia" resourceUsed="wiki"/>
            </sgf:layer>
        </sgf:level>
    </sgf:annotation>
</sgf:corpusData>
Overview

- SGF – The Sekimo Generic Format
- Real World Application Scenarios
  - Analyzing anaphoric relations in the Sekimo project
  - Exchange format for the lexical chaining tool *Scientific Workplace*
  - Exchange format for the annotation tool *Serengeti*
- Conclusion
Exchange format for the annotation tool *Serengeti*

A few words about the AnaWiki/AnaphoricBank project:

- Started 2008 at University of Essex (cf. Poesio & Kruschwitz, 2008)
- Project aim: Development of tools to allow and encourage large numbers of volunteers over the Web to collaborate in the creation of semantically annotated corpora
- Project method: Game Interface (*Phrase Detectives*) & Expert Annotation Interface (*Serengeti*)
Serengeti Annotator V0.8.11 (Beta)
Semantic Relations Annotation Tool

The Serengeti Annotator is a web based client-server-application used for annotating semantic relations in text documents.

It is developed in the SEKIMO (A2) project, part of the distributed research group "TEXT TECHNOLOGICAL MODELLING OF INFORMATION" funded by the DFG.

Optimized for Mozilla Firefox > V2.0

Last Update: 2008-11-20 by Nils Diewald

Exchange format for the annotation tool Serengeti
Exchange format for the annotation tool Serengeti

b. 1642 Paris, d. 1732 Paris; master before 1666 French ebeniste

Christened by his contemporaries as "the most skillful artisan in Paris", Andre-Charles Boulle's name is synonymous with the practice of veneering furniture with marquetry of tortoiseshell, pewter, and brass. Although he did not invent the technique, Boulle was its greatest practitioner and lent his name to its common name: boule work. Boule also specialized in floral marquetry in both stained and naturally colored wood. Many of his designs are illustrated in a book of engravings published around 1720.

Before 1666, Boulle was awarded the title of master cabinetmaker in 1672, the king granted him the royal privilege of lodging in the Palais du Louvre. In the same year, he achieved the title of cabinetmaker and sculptor to Louis XIV, king of France. This new title allowed him to produce furniture as well as works in gilt bronze, such as chandeliers, wall lights, and mounts. Although strict guild rules usually prevented craftsmen from practicing two professions simultaneously, Boulle's favored position allowed him protected status and exempted him from these statutes.

```
<markable segment="seg452" id="mark_ne65" type="nom" reference="unmarked" gender="neut" number="sing" person="per3" />
<markable segment="seg464" id="mark_ne67" type="nom" reference="unmarked" gender="masc" number="sing" person="per3" />
<markable segment="seg466" id="mark_ne68" type="nom" reference="unmarked" gender="neut" number="sing" person="per3" />
<markable segment="seg476" id="mark_ne69" type="nom" reference="unmarked" gender="masc" number="sing" person="per3" />
<markable segment="seg480" id="mark_ne70" type="nom" reference="unmarked" gender="neut" number="plur" person="per3" />
<newRelation id="sr2" />
```
Exchange format for the annotation tool Serengeti
A few words about *Serengeti*:

- *Serengeti* is a web-based annotation tool developed in the Sekimo project at Bielefeld University.
- It supports multi-user annotation, central file and user management and import & export capabilities.
- While the first version was used exclusively in our project (cf. Stührenberg et al., 2007) the current version is employed as Expert Annotation Interface in the AnaWiki/AnaphoricBank project.
- The current version supports different annotation schemas, and manipulation of markables (i.e. add/edit/delete markables).
Exchange format for the annotation tool *Serengeti*

**SGF and Serengeti:**

- *Serengeti* uses a relational database (MySQL) and SGF as import/export format
- The SGF API for storing SGF instances and accessing segments and single annotation layers can be used inside and outside *Serengeti*
Exchange format for the annotation tool Serengeti

The SGF MySQL database schema
The SGF MySQL database schema
Exchange format for the annotation tool Serengeti

The SGF MySQL database schema
Exchange format for the annotation tool Serengeti

The SGF MySQL database schema
Exchange format for the annotation tool Serengeti

The SGF MySQL database schema
Exchange format for the annotation tool *Serengeti*

Multiple annotated data – second reading

- *Serengeti* allows several users to annotate the same text – and so does SGF.
- Each `sgf:level` element is linked to an annotator (either human or machine) via the `annotator` attribute.
- SGF's log functionality provides a means to save the history of actions regarding a specific SGF file.
Exchange format for the annotation tool Serengeti

SGF's log functionality

- The `sgf:log` (which is defined in a separate XSD file) contains at least one `sgf:entry` element
- Each entry consists of one or more add/modify/delete action(s)
SGF's log functionality

- While the `add` element is a simple wrapper for elements derived from another namespace, the `modify` element makes use of XML Schema's possibility to skip the processing of the wrapper element's content.
- The `delete` element uses the ID reference of the node to be deleted.

```xml
<xs:element name="modify">
  <xs:complexType>
    <xs:sequence>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="skip"/>
    </xs:sequence>
    <xs:attribute ref="log:affectedItem" use="required"/>
  </xs:complexType>
</xs:element>

<xs:element name="delete">
  <xs:complexType>
    <xs:attribute ref="log:affectedItem" use="required"/>
  </xs:complexType>
</xs:element>
```
The SGF API

- Apart from the representation of the structure of SGF's base layer the SGF API provides filters (written as Perl classes) for SGF annotation levels
- Each import/export filter is triggered by the annotation level's namespace
- The current version of Serengeti which was developed as expert annotation interface for the AnaWiki/AnaphoricBank project uses the following filters:
  - Metadata – for general information about the text (author, title, etc.)
  - Logical document structure – for rendering of the text in the text frame
  - Markables – for allowing spans of the text to be enriched with various types of information and to be selected as parts of anaphoric relations
  - Anaphoric relations – for adding/modifying/deleting anaphoric relations
- Other import/export filters can be easily developed – all filter methods are derived from a base filter class and Serengeti will go Open Source after the project is finished (Q4 2008)
Let's start a live example...
Overview

- SGF – The Sekimo Generic Format
- Real World Application Scenarios
  - Analyzing anaphoric relations in the Sekimo project
  - Exchange format for the lexical chaining tool *Scientific Workplace*
  - Exchange format for the annotation tool *Serengeti*
- Conclusion
To sum up:

- We've presented SGF – The Sekimo Generic Format – capable of storing multiple annotated primary data (in terms of multiple resources and multiple users).
- SGF is XML-based on the Annotation Graph model and is a logical descendant to the Prolog fact base format.
- SGF combines inline and standoff annotation and allows for validating multiple annotated data.
- SGF can be queried and processed by using standard XPath or XQuery – performance depends on the imported annotation layers.
- Apart from the Sekimo project SGF is used in other projects as well – and will be made available to other interested parties.
- Together with the SGF-API developed in conjunction with Serengeti SGF is a powerful tool.
Conclusion

So...

...in what way is SGF concerned with Goddag (or – the other way round – why am I here)?
Conclusion

- We are dealing with very similar data structures
- We have the same linguistic background
- We'd like to collect and access experience of other applications and methods
- We'd like to take part in the ongoing process of further development
Last but not least...

Thank you for your attention!

Visit us at http://www.text-technology.de/sekimo

{maik.stuehrenberg|daniela.goecke|nils.diewald|daniel.jettka}@uni-bielefeld.de
Bibliography


