TEI and Databases

Øyvind Eide

June 2009
TEI and Databases

Overview

• Introduction
• Short history
• Types of connections between TEI documents and databases
• Examples
• Conclusions
TEI and Databases

Overview

• Introduction
• Short history
  • Types of connections between TEI documents and databases
• Examples
• Conclusions
TEI and Databases

Overview

• Introduction
• Short history
• Types of connections between TEI documents and databases
• Examples
• Conclusions
Overview

- Introduction
- Short history
- Types of connections between TEI documents and databases
- Examples
- Conclusions
TEI and Databases

Overview

- Introduction
- Short history
- Types of connections between TEI documents and databases
- Examples
- Conclusions
Introduction

Ways to use databases with TEI:

- store TEI documents in databases
- store references to TEI documents in databases
- store information in databases that is based on information in TEI documents
Introduction

Ways to use databases with TEI:

- store TEI documents in databases
- store references to TEI documents in databases
- store information in databases that is based on information in TEI documents
Introduction

Ways to use databases with TEI:

- store TEI documents in databases
- store references to TEI documents in databases
- store information in databases that is based on information in TEI documents
Use of databases with TEI

Ways to use databases with TEI:

- store TEI documents in databases
- store references to TEI documents in databases
- store information in databases that is based on information in TEI documents
Historic example: XML in museums

- Cultural heritage information in XML
  - Complex
  - Often not TEI
Historic example: XML in museums

- Cultural heritage information in XML
- Complex
- Often not TEI
Historic example: XML in museums

- Cultural heritage information in XML
- Complex
- Often not TEI
Historic example: Mapping of archaeological data

- Archives: catalogues, reports etc.
- Interpretation and descriptive process
  - according to an archeological practice or theory
- Interpretation of texts by an encoder according to a fixed database ontology (model)
- "Real world" Excavation Surveys
  - according to an archeological practice or theory
  - according to a fixed database ontology (data model)
- Common database
Historic example: Processing archaeological data

- Original text (text witness)
  - step 1: registration
- Bibliographical record
  - step 2: reproduction
- Facsimile
  - step 3: transcription
- Text with XML markup:
  1) Structural markup
  2) Lemmatization etc.
  - step 4: content markup
- Text with XML markup: Information elements identified and marked up according to the data model of the museum database
  - connected to
- Museum database artefacts, excavations, referential information. Event/object oriented model (CIDOC-CRM like)
Historic example: Processing archaeological data

- Original text (text witness)
  ↓ *step 1: registration*

- Bibliographical record
  ↓ *step 2: reproduction*

- Facsimile
  ↓ *step 3: transcription*

- Text with XML markup:
  1) Structural markup
  2) Lemmatization etc.
  ↓ *step 4: content markup*

- Text with XML markup: Information elements identified and marked up according to the data model of the museum database
  ↓ connected to

- Museum database artefacts, excavations, referential information. Event/object oriented model (CIDOC-CRM like)
Historic example: Processing archaeological data

• Original text (text witness)
  ➾ step 1: registration

• Bibliographical record
  ➾ step 2: reproduction

• Facsimile
  ➾ step 3: transcription

• Text with XML markup:
  1) Structural markup
  2) Lemmatization etc.
  ➾ step 4: content markup

• Text with XML markup: Information elements identified and marked up according to the data model of the museum database
  ➾ connected to

• Museum database artefacts, excavations, referential information. Event/object oriented model (CIDOC-CRM like)
Historic example: Processing archaeological data

- Original text (text witness)
  \( \downarrow \text{step 1: registration} \)
- Bibliographical record
  \( \downarrow \text{step 2: reproduction} \)
- Facsimile
  \( \downarrow \text{step 3: transcription} \)
- Text with XML markup:
  1) Structural markup
  2) Lemmatization etc.
  \( \downarrow \text{step 4: content markup} \)
- Text with XML markup: Information elements identified and marked up according to the data model of the museum database
  \( \uparrow \text{connected to} \)
- Museum database artefacts, excavations, referential information. Event/object oriented model (CIDOC-CRM like)
Historic example: Processing archaeological data

- Original text (text witness)  
  ↓*step 1: registration*
- Bibliographical record  
  ↓*step 2: reproduction*
- Facsimile  
  ↓*step 3: transcription*
- Text with XML markup:  
  1) Structural markup 2) Lemmatization etc.  
  ↓*step 4: content markup*
- Text with XML markup: Information elements identified and marked up according to the data model of the museum database  
  ↓*connected to*
- Museum database artefacts, excavations, referential information. Event/object oriented model (CIDOC-CRM like)
Historic example: Processing archaeological data

- Original text (text witness)
  - step 1: registration
- Bibliographical record
  - step 2: reproduction
- Facsimile
  - step 3: transcription
- Text with XML markup:
  1) Structural markup
  2) Lemmatization etc.
  - step 4: content markup
- Text with XML markup: Information elements identified and marked up according to the data model of the museum database
  - connected to
- Museum database artefacts, excavations, referential information. Event/object oriented model (CIDOC-CRM like)
TELification: ODD

- Can store complicated models in TEI
- Can put archaeologically tagged paragraphs into \textit{p} elements
- Archaeological structure included using ODD
- Documents will be quite odd
- Useful?
TEI and Databases

TELifisation: ODD

• Can store complicated models in TEI
• Can put archaeologically tagged paragraphs into p elements
  • Archaeological structure included using ODD
• Documents will be quite odd
• Useful?
TEI and Databases

TEElifesation: ODD

- Can store complicated models in TEI
- Can put archaeologically tagged paragraphs into $p$ elements
- Archaeological structure included using ODD
- Documents will be quite odd
- Useful?
TEI and Databases

TElifisation: ODD

• Can store complicated models in TEI
• Can put archaeologically tagged paragraphs into p elements
• Archaeological structure included using ODD
• Documents will be quite odd
• Useful?
TEI and Databases

TElifisation: ODD

• Can store complicated models in TEI
• Can put archaeologically tagged paragraphs into \texttt{p} elements
• Archaeological structure included using ODD
• Documents will be quite odd
• Useful?
The rest of this presentation will be about elements already existing in TEI

*No more ODDities*
Referring strings are connected to objects representing things in a world. We call these objects *real world object*, although they do not have to represent things in a *real* world.

Such connections are based on the following three building blocks:

- Referring strings: e.g. TEI `placeName`
- Real world objects: e.g. TEI `place`
- Connections: e.g. `ref→xml:id`
Referring strings are connected to objects representing things in a world. We call these objects *real world object*, although they do not have to represent things in a *real* world.

Such connections are based on the following three building blocks:

- Referring strings: e.g. TEI `placeName`
- Real world objects: e.g. TEI `place`
- Connections: e.g. `ref→xml:id`
Connecting TEI to the world

Referring strings are connected to objects representing things in a world. We call these objects *real world object*, although they do not have to represent things in a *real* world.

Such connections are based on the following three building blocks:

- Referring strings: e.g. TEI `placeName`
- Real world objects: e.g. TEI `place`
- Connections: e.g. `ref→xml:id`
How to store real world objects?

- Three different ways of storing real world object sets connected to TEI documents:
  1. Store the objects in external XML-documents, e.g. RDF or CRM-Core
  2. Store the objects in the TEI header using an external XML name space, e.g. RDF or CRM-Core
  3. Store the objects in the TEI header using the existing elements in TEI P5

- From all of these, connection can then be made to databases
- Database objects can also be connected directly to referring strings in TEI documents
How to store real world objects?

- Three different ways of storing real world object sets connected to TEI documents:
  1. Store the objects in external XML-documents, e.g. RDF or CRM-Core
  2. Store the objects in the TEI header using an external XML name space, e.g. RDF or CRM-Core
  3. Store the objects in the TEI header using the existing elements in TEI P5

- From all of these, connection can then be made to databases
- Database objects can also be connected directly to referring strings in TEI documents
How to store real world objects?

• Three different ways of storing real world object sets connected to TEI documents:
  1. Store the objects in external XML-documents, e.g. RDF or CRM-Core
  2. Store the objects in the TEI header using an external XML name space, e.g. RDF or CRM-Core
  3. Store the objects in the TEI header using the existing elements in TEI P5

• From all of these, connection can then be made to databases
• Database objects can also be connected directly to referring strings in TEI documents
How to store real world objects?

- Three different ways of storing real world object sets connected to TEI documents:
  1. Store the objects in external XML-documents, e.g. RDF or CRM-Core
  2. Store the objects in the TEI header using an external XML name space, e.g. RDF or CRM-Core
  3. Store the objects in the TEI header using the existing elements in TEI P5

- From all of these, connection can then be made to databases

- Database objects can also be connected directly to referring strings in TEI documents
How to store real world objects?

- Three different ways of storing real world object sets connected to TEI documents:
  1. Store the objects in external XML-documents, e.g. RDF or CRM-Core
  2. Store the objects in the TEI header using an external XML name space, e.g. RDF or CRM-Core
  3. Store the objects in the TEI header using the existing elements in TEI P5

- From all of these, connection can then be made to databases
- Database objects can also be connected directly to referring strings in TEI documents
Example: Lexicography — database application

aller \( \text{al}^{\text{i}} \text{\`l}^{\text{er}} \) (adv)
1. aldri
2. vel ikke - ho kjæm vel allier hit?

**HemnesBold**

*(Forkortingsliste)*
Example: Lexicography — database model
Example: Lexicography — database records

```
SELECT *
FROM artikkel t
WHERE t.artikkel_id > 83615 AND t.artikkel_id < 83625
```
<entry xml:id="HemnesBold_orig41">
  <form type="simple">
    <orth extent="full">
      <hi rend="bold">aller</hi>
    </orth>
    <pron extent="full">
      <hi rend="norvegia">al9 l9er</hi>
    </pron>
  </form>
  <gramGrp>(<pos>adv</pos>)</gramGrp>
  <def>1. aldri</def>
  <def>2. vel ikke - <hi rend="italic">ho kjæm vel ailler hit?</hi></def>
</entry>
Example: Co-reference — what is it?

“The table by the window was…”

“The beautiful table could be…”
Example: Co-reference — application model
Example: Co-reference — core data model
Example: Co-reference — link model

- The person table acts as a local coref storage for these source documents.
- Even if the connection goes from the person record to the external reference, the link to the source element gives us co-reference at document level.
- The person table acts as a local coref storage for these source documents
  - Even if the connection goes from the person record to the external reference, the link to the source element gives us co-reference at document level.
Example: Co-reference — link model

- The person table acts as a local coref storage for these source documents
- Even if the connection goes from the person record to the external reference, the link to the source element gives us co-reference at document level.
Conclusion

• Not database or TEI, but both
• Best of two worlds:
  • XML for texts
  • Databases for data integration
• Every project must find its balance
Conclusion

• Not database or TEI, but both
• Best of two worlds:
  • XML for texts
  • Databases for data integration
• Every project must find its balance
Conclusion

- Not database or TEI, but both
- Best of two worlds:
  - XML for texts
  - Databases for data integration
- Every project must find its balance
Conclusion

- Not database or TEI, but both
- Best of two worlds:
  - XML for texts
  - Databases for data integration
- Every project must find its balance