SIKB0102: SYNCHRONISING EXCAVATION DATA FOR PRESERVATION AND RE-USE

“We need references”

Wouter Boasson MSc & Ronald Visser Ma
About us

• Wouter Boasson
  • Physical Geography (MSc)
  • GIS & spatial database specialist, including teaching and training
  • Working in archaeology for > 15 year (RAAP)

• Ronald M. Visser
  • Archaeology (MSc)
  • GIS & spatial database specialist, including teaching and training
  • Lecturer in digital archaeology (Saxion University of Applied Sciences)

Both members of the drafting group for the SIKB0102 specification
SIKB0102: SYNCHRONISING EXCAVATION DATA FOR PRESERVATION AND RE-USE

• SIKB0102: Dutch data exchange standard for archaeological (excavation) data.

• Synchronising <-> Exchanging

• Standard created by experienced drafting group
  • archaeologists
  • database specialists
  • Xml expert
SIKB0102 evolution

• Developed from an Xml metadata document
  • containing basic lists of finds, boxes, drawings, files, ...
  • SIKB0102 version 1

• Time-out

• A full excavation documentation standard including
  • features, structures, relations, project specific attributes, ...
  • full geo-information
  • SIKB0102 versions 2 and 3
Why a full exchange standard

- Research perspective:
  - (re)use excavation data

- Business perspective
  - One exchange format
  - Cheaper (in the long run)

- Archives
  - One format
  - Long time storage
Typical solution for an exchange standard

- Define a standard data structure (e.g. in Xml)
  - Attributes
  - Relations
- Require use of codelists.
- Can't cope with specialized data.

- Use: reduce your dataset until it fits.
From a business perspective...

• Good enough for management information
  • Location of surveys
  • Results summary

• Easy to implement
  • Standard structure

• Doesn’t offer a full solution
  • Need to deliver the full dataset with all the details to archives
    (government/requirement)
and from the research point of view

• Too much loss of detail, examples:
  • Utopia: a complete thesaurus
  • Additional properties registered for specific research projects can by definition not be part of the specification.
  • No provision for unknown relations between objects (e.g. finds and samples).

• **Research requires the option to add new insights!**

• **An exchange standard for research purposes**
  • *traditionally considered impossible*
Update of SIKB0102 version 1

- Basic metadata -> full excavation data
- Implement full geo-data support
- Simply allow adding original data in some free fields is not a solution
  - Meaning?
  - Validation?
  - Relation with the standardized code lists?
- From a technical point of view, there are even more difficulties.
Reference framework

• Metadata
• Information on
  • meaning of objects
  • attributes
  • codes
Concept of useless data

• Nice piece of?
Add a reference

• Reference for its size.
And another one

• Reference for its size.
• Something in a hydro power station.
May be we got to a reference 'framework'? 

• Reference for its size.
• Something in a hydro power station.
• Has to do with Pelton turbines.
• **Adjustable tip of injector nozzle.** (Tyssedalen, Norway)
Excavation data exchange: enhancing SIKB0102, reference framework

• Allow inserting any relation between objects.
• Include the original level of detail.
• Enable adding attributes to pre-defined standard objects.

• All with a reference framework for the meaning of the data.
Providing a reference framework for custom additions

• **In Xml**
  • Create a new Xsd (schema) building on top of the existing.
  • Pro: standard technology.
  • Con: very difficult (create and read/convert), depends on Xml.

• **Generic**
  • Include the data objects needed for the reference framework in the standard
  • Pro: much easier, other than the Xml vehicle could also be used.
  • Con: nobody does it, uncharted territory.
A typical XML data structure

• Hierarchical
  • Duplicates when referencing objects multiple times

• No unique identifiers
  • Needed if you want to make relations without duplicates
  • Needed when you want to extend objects in a predictable way (without extending the base model using a new Xsd)

• Both considered showstoppers
  • Document size
  • Research requires the option to add detailed data, including relations
Example: typical Xml structure

<drawing id="1">
    <scale>10</scale>
    <find id="3">
        <feature>10</feature>
        <artefactType>XYZ</artefactType>
        <count>3</count>
    </find>
</drawing>

<image id="1">
    <file>dsc012345.jpg</file>
    <find id="3">
        <feature>10</feature>
        <artefactType>XYZ</artefactType>
        <count>3</count>
    </find>
</image>
Action: new structure

• Add explicit object types for every type of data
  • Eliminate duplicates
  • Make it possible to extend individual object types

• Add unique identifiers
  • Enable linking of objects
  • Make it possible to extend individual object types

• Design the structure for our reference framework
The new SIKB0102 structure (version >= 2)

- Relational ‘database like’ exchange format, including geo-information
- Eliminates duplicates of data
- Use of UUIDs
  - eliminates relying on all kinds of different keys assigned by users
  - combining of multiple datasets possible without key clashes
  - meaningless administrative keys are removed
SIKB0102 structure

• Basic data objects
  • reference framework included in specification

• Required relations
  • reference framework included in specification

• Additional sections for
  • custom relations
  • custom codes
  • custom attributes
Additional sections consist of

• Reference framework
  • data objects describing the additional attributes, relations and codes

• Data objects
  • example: user attributes
Ready for the interesting additions

• Custom relations
• Custom codes
• Custom attributes
Custom relations: object relation section

- Link any object to any other object
- Basically a many-to-many 'table'
- Meta-information included
  - type of relation
  - related object classes
- Use
  - Stratigraphy/Harris matrix
  - Structures
  - Find concentrations
  - ...

<table>
<thead>
<tr>
<th>Id1</th>
<th>Class1</th>
<th>Id2</th>
<th>Class2</th>
<th>RelType</th>
</tr>
</thead>
<tbody>
<tr>
<td>bach</td>
<td>Feature</td>
<td>purce</td>
<td>Feature</td>
<td>OlderThen</td>
</tr>
<tr>
<td>bach</td>
<td>Feature</td>
<td>hande</td>
<td>Structure</td>
<td>BelongsTo</td>
</tr>
<tr>
<td>faure</td>
<td>Feature</td>
<td>hande</td>
<td>Structure</td>
<td>BelongsTo</td>
</tr>
<tr>
<td>shost</td>
<td>Feature</td>
<td>hande</td>
<td>Structure</td>
<td>BelongsTo</td>
</tr>
</tbody>
</table>
Custom codes 1: codereference section

• Provide specialized terms where thesaurus lacks detail
  • Includes description
  • Link with thesaurus term

• Implementation
  • Optional forward reference to original codes

• Advantages
  • Thesaurus terms always available
  • More detail includes description
## Custom codes 2

### Object Id | ArtefactType | ArchDatedBegin | ArchDatedEnd
--- | --- | --- | ---
brahm | ANDKOM [haydn] | MEVD | NTV
mende | ANDKOM [haydn] | MEVD | NTV
mozar | ANDKOM [beeth] | MEVD | NTV

<table>
<thead>
<tr>
<th>CodeId</th>
<th>SrcCode</th>
<th>SrcDescription</th>
<th>SrcCodeList</th>
<th>ThesCode</th>
<th>ThesList</th>
</tr>
</thead>
</table>
haydn | wm.kom | witbakkend Maaslands: kan | art_ceramtype | ANDKOM | ArtefactType |
beeth | wm.stk | witbakkend Maaslands: steelkom | art_ceramtype | ANDKOM | ArtefactType |
Custom attributes: user attribute section

• Add attributes to objects
• Metadata: describes attribute, valuetype, link with object
• Data: values

<table>
<thead>
<tr>
<th>AttId</th>
<th>Attribute</th>
<th>Class</th>
<th>ValueType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dvora</td>
<td>EVE</td>
<td>Find</td>
<td>Number</td>
<td>Eve (fraction 0-1)</td>
</tr>
<tr>
<td>grieg</td>
<td>C14CALY</td>
<td>Sample</td>
<td>Number</td>
<td>C14 calibrated years</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AttId</th>
<th>ObjectId</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dvora</td>
<td>brahm</td>
<td>0.4</td>
</tr>
<tr>
<td>dvora</td>
<td>schub</td>
<td>0.1</td>
</tr>
<tr>
<td>grieg</td>
<td>schum</td>
<td>3200</td>
</tr>
</tbody>
</table>
SIKB0102: *Synchronising* excavation data ‘understanding’

• No-brainer: send and receive data, read the characters.
• Speak the same language!
SIKB0102: *Synchronising* excavation data ‘technical’

- Datasets can change over time (typically grow or get updated).
- Meaning of codings (descriptions) could change.
- As a result: multiple versions of the same dataset could exist.
Managing versions

• Use of globally unique identifiers allows for updating user data.
  • Individual records (by using globally unique identifiers, no dependency on user keys).

• Code-lists are versioned as to be able to deal with updates.
  • Metadata of code books (even with references to previously used codes and/or meanings).
Implementation drawbacks

• Xml based
• Some flaws in design
  • Archaeological complexes not separated from find concentrations
  • Not clear how to deal with some relations
• A lot of overhead in storage of custom attributes, relations and codes
• No structure present to understand the original keys
Take aways

• Biggest issue: willingness
  • Large-scale use is still a point on the horizon.

• References are the key
  • Makes your data understandable, and future proof.

• Pay attention to future expansion
  • Avoid having to change the fundamentals

• Size matters
  • Be aware of serious differences in system behaviour depending on dataset size

• Don’t push your luck too far

• Be prepared: resistance

• Be patient: workflows change

• Yes, exchanging data with research quality and integrated geo-information can really work!
Real world. SIKB0102 is an Xml standard

**Why: it’s the default option.**
- Well constrained
- Many (standard) tools available
- SIKB0102 another ‘soil-archive’ standard, just like SIKB0101 (Xml)
- Started of as a pure metadata standard (little data).
- “Everybody does it”

**The challenges.**
- Typical implementation uses hierarchical structure
- Not suited for extremely large code list constraints
- Blows up with large datasets
- Really large datasets cannot be handled by Xml document parsers
Xml has serious disadvantages

• Xsd (Xml Schema Document) validation does not work against super large documents.
• Document parsers cannot handle 100s of MBs.
• Streaming parser needed: difficult

• Still, SIKB0102 is an Xml format, we never discussed that.