
NAACCR XML in ACTION

— SEER*Abs & SEER*DMS —

Fabian Depry, IMS

NAACCR 2016, St. Louis, Missouri

Goals of this Presentation

- Demonstrate supporting NAACCR XML in real, complex applications.
- Provide some feedback to the NAACCR community.
- Identify tips/gotchas for other people going through the process.

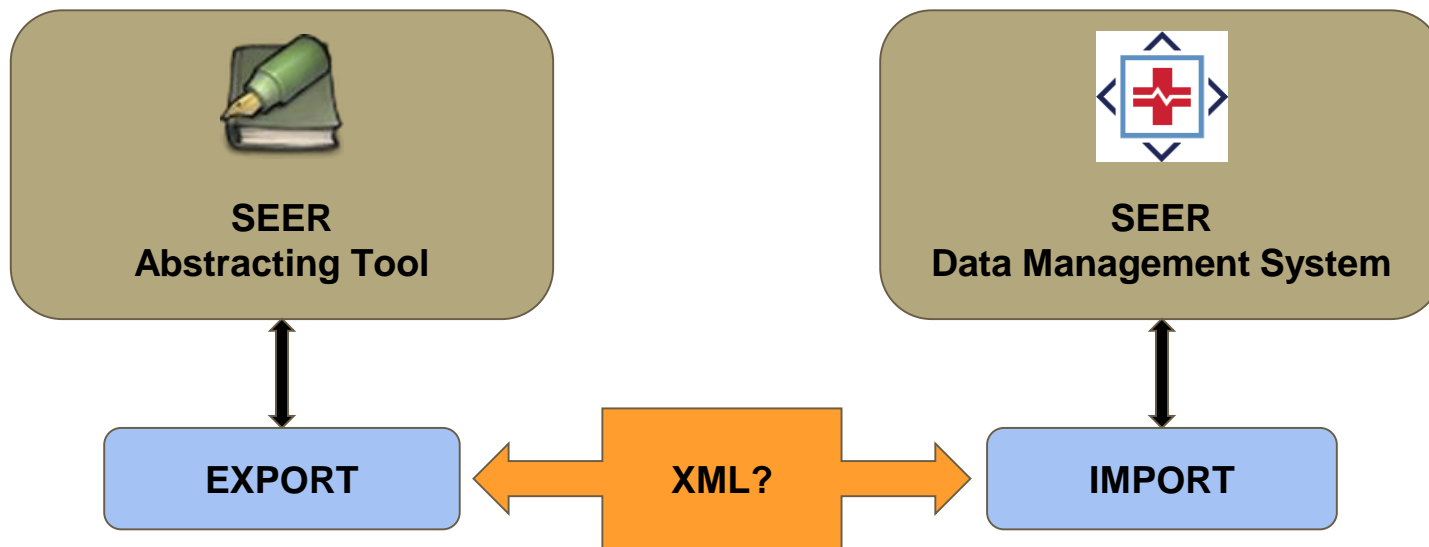
But keep in mind...

- My focus is on data transfer between two specific Java applications.
- Just one use case among MANY of them.

NAACCR XML Overview

- NAACCR FFlat File is a very successful format but has limitations.
- XML addresses many of those limitations, but is more complex.
- Version 1.0 of the XML standard was approved and released in 2015.
- The standard has not been adopted by vendors and might take a while.
- Checkout <https://naaccrxml.org>

Introducing SEER*Abs and SEER*DMS



Creating XML Data in SEER*Abs

- SEER*Abs is highly configurable via XML definition files and Scripts.
- Extract is a (Groovy) script customized by the registries.
- Core application provides context utility methods for the scripts.

This means...

- Changes to the core application only involve new context methods.
- No disruption to registries who are currently using SEER*Abs.

Changes in SEER*Abs

- Two new context functions were added in the core application:
 - **writeNaaccrXmlFile**(records, targetFile, registryDictionary)
 - **writeNaaccrFlatFile**(records, targetFile, registryDictionary)
- Default extract script was modified to handle XML:
 - Translates SEER*Abs field names into NAACCR item IDs.
 - Calls Java library via context function.
 - Can switch between XML and Flat-file mode via one global configuration variable.

Conclusions for SEER*Abs

- Adding support was quick and simple (about a day of work).
- Task was greatly simplified because
 - All XML operations were handled by the NAACCR XML Java library.
 - Internal (database) representation of the SEER*Abs records didn't change.
- Changes to the core application went quickly to production.
- Changes to the customized scripts didn't go to production.
 - Mainly because current XML format brings no advantages over the Flat-file one.

We are halfway there!

First part of the task:

PRODUCE XML Data using SEER*Abs

Second part:

CONSUME XML Data using SEER*DMS

Reading XML Data in SEER*DMS

- SEER*DMS supports many data types via its import module.
- Each data type is processed by an “import algorithm”.
- Algorithms save data in a “RECORD” table containing all SEER*DMS fields.

This means...

- Changes to the application only involve creating a new algorithm.
- Not disruptive to the registries, they don't use new algorithm yet.

Changes in SEER*DMS

- Added new import algorithm to handle NAACCR XML Data files.
- Used NAACCR XML Java library to “read next patient” from the data file.
- Used target DB fields already defined in Flat-file algorithms.
- Used data conversion rules already defined in Flat-file algorithms.
- Did NOT change anything in the persistence layer or record data model.

Conclusions for SEER*DMS

- Adding support was a bit challenging (about a week of work).
- Task was greatly simplified because
 - All XML operations were handled by the NAACCR XML Java library
 - Internal (database) representation of the SEER*DMS records didn't change.
- But it was still challenging because
 - All processes applied by Flat-file imports had to be applied by XML import as well.
 - Java library had technical limitations that had to be worked around.
- New algorithm went quickly to production.
- No registry is currently using it.

Testing Changes Using Real Data

- Asked Utah for their SEER*Abs installation and converted it to XML.
 - Defined a registry-specific NAACCR Dictionary for their non-standard variables.
- Extracted 500 records using “Flat-file” and the same records using “XML”
 - Flat file was created in 7 seconds and was 11MB (uncompressed).
 - XML file was created in 4 seconds and was 4MB (uncompressed).
- Imported both data files in SEER*DMS and compared the RECORD table.
 - Made sure records from both files were completely identical.

Conclusions

- Switching from Flat-File to NAACCR XML
 - Only affects data transmission (import and export).
 - No impact on other system components.
 - No impact on database structure of either system.
- Overall a very positive experience
 - No “deal breaker” issue found in the specifications or the Java library.
 - Minor bugs in the Java library were identified and fixed.
 - Limitations in the XML format specifications were highlighted, mainly
 - Text fields still capped at 1,000 characters.
 - Registry variables still tied to a location in the State Requestor Items.

Conclusions

- Because of the those limitations, there is very little incentive for the Registries to spend any effort on switching to the new XML format.
- Those limitations are being addressed by the NAACCR XML Work Group.
- Hope to use the next version of the XML standard in August 2016 (New Mexico Tumor Registry - SEER*Abs and SEER*DMS)
 - Abstract layout requires additional (non-standard) text fields that exceed 1000 char
 - May use a beta-release, if production version is not available

Conclusions

- Hopefully this work will be a **Gateway** to technical advances in cancer registry operations.
- Come visit us at the IMS booth!
- THANK YOU!

