

# QRDA Schematron Project Notes

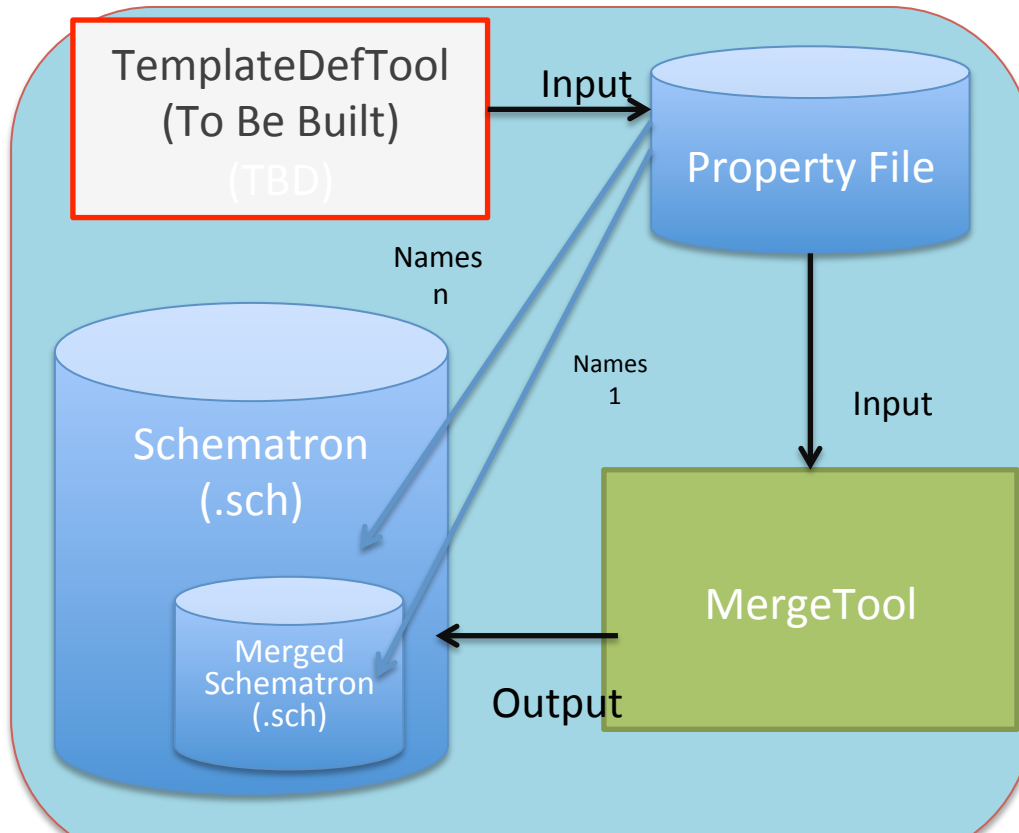
*SDV*

6/6/16 12:54 PM

Friday, June 3, 16

# **ARCHITECTURE DIAGRAMS (HIGH LEVEL)**

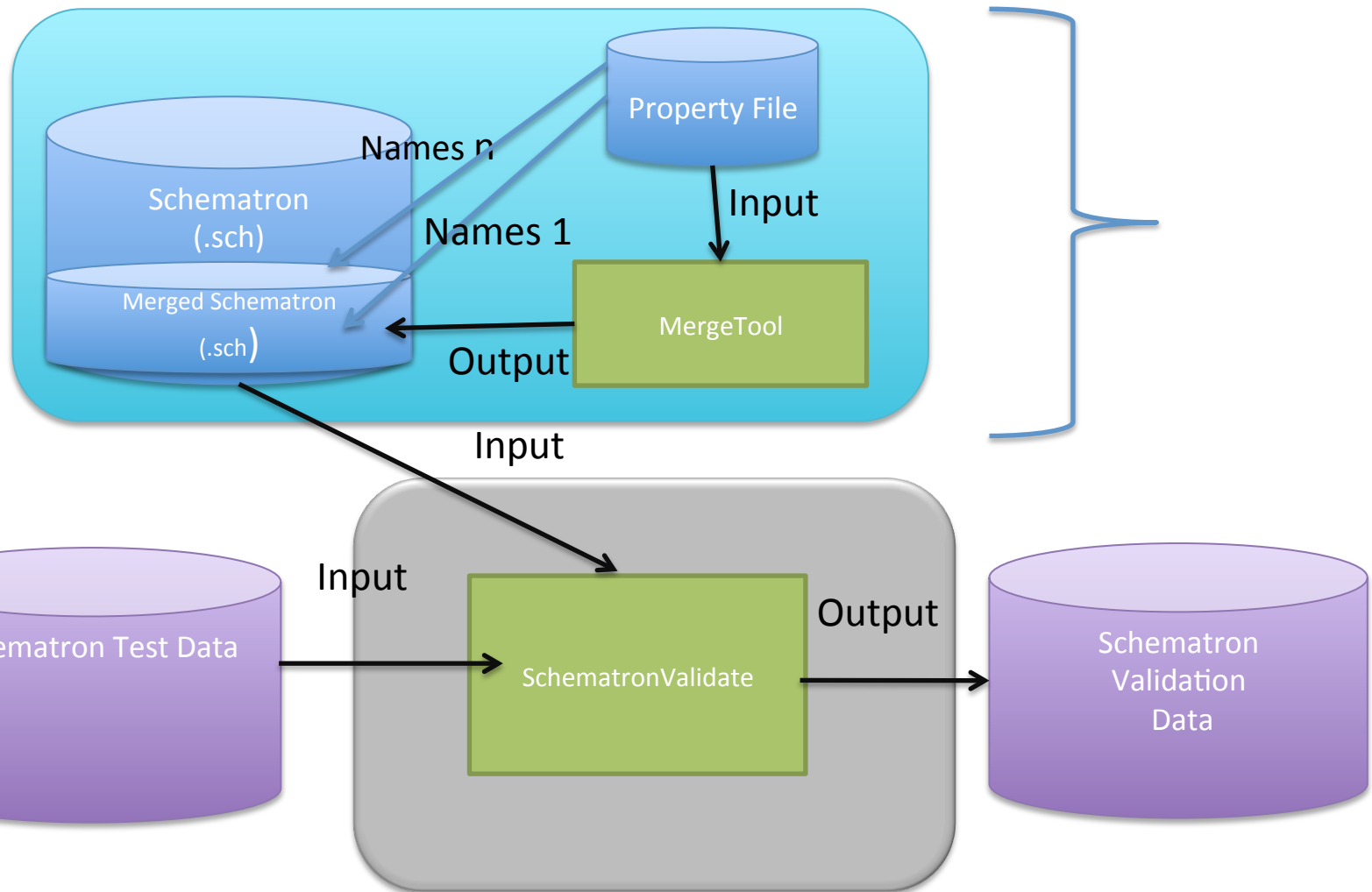
# Schematron Merge Architecture



- Each Property File
  - Names N Schematron Files to Merge
  - Specifies the name of the merged file
  - (See sample file below)

```
1 MergeConfigure.java
2 HL7_core_STU3.1_schematron.properties
3
4 1 NumberOfFiles=3
5 2 Schematron_1=/Users/shonvick/Documents/qdraSchematron/schemaVerification/Communication_From_Patient_To_Provider_V3_Entry.sch
6 3 Schematron_2=/Users/shonvick/Documents/qdraSchematron/schemaVerification/Communication_From_Provider_To_Provider_V3_Entry.sch
7 4 Schematron_3=/Users/shonvick/Documents/qdraSchematron/schemaVerification/Communication_From_Provider_To_Patient_V3_Entry.sch
8 5 MergedFileName=/Users/shonvick/Documents/HL7 QRDA Category\ I\ STU3.1.sch
```

# Schematron Validation Architecture



Friday, June 3, 16

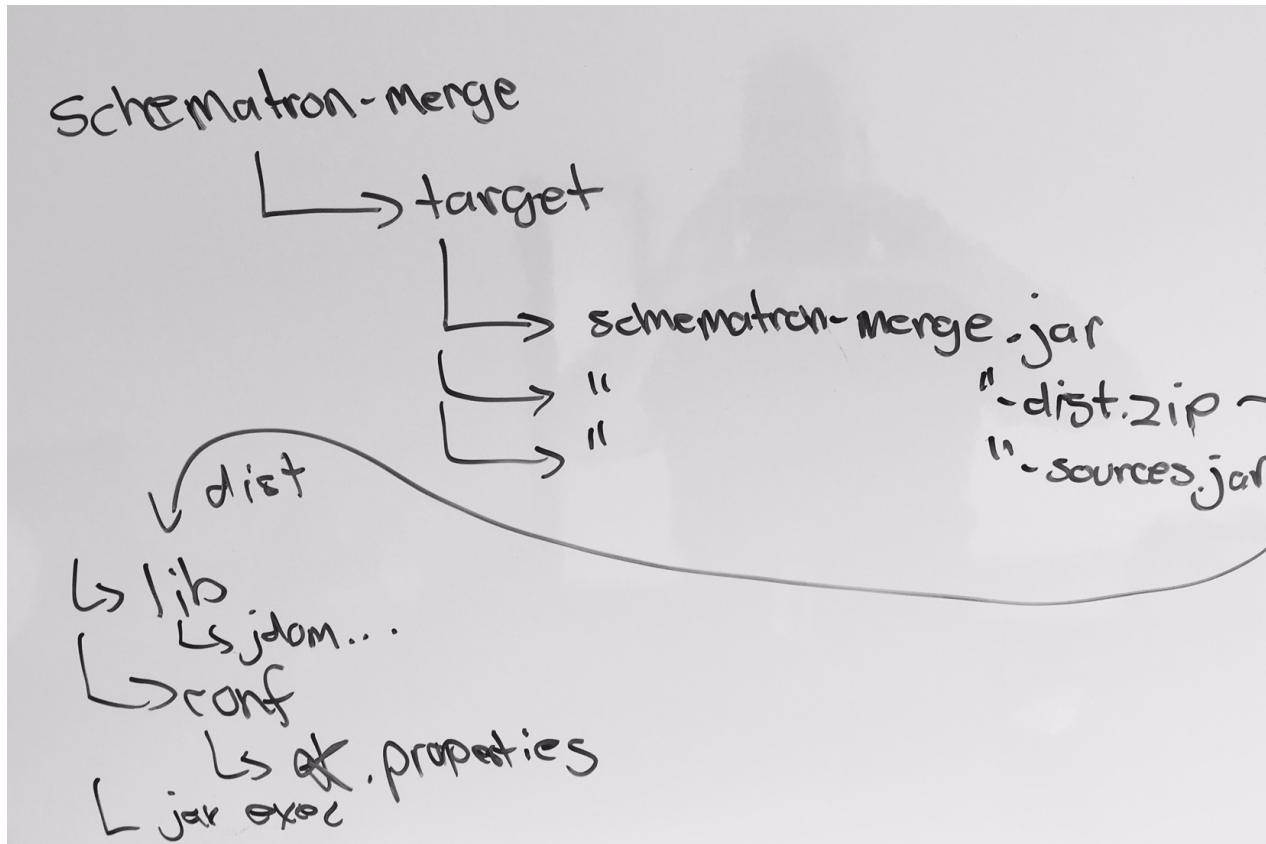
# **NOTES ON CAPTURING POM FILE DEPENDENCIES**

# Initial thoughts

## Phases of a Maven Build Lifecycle

- **Validate**
  - Make sure that there are Properties files
  - Files to compile are there (see Compile phase)
  - Unit tests to compile (see Test Phase)
  - Make sure there is a spreadsheet entry for the three property file cases (see MH Note)
- **Compile**
  - Write Code for
    1. MergeConfigure.java ✓
    2. Schematron.java ✓
    3. SchematronMerge.java ✓
- **Test**
  - Write Junit tests for
    1. MergeConfigureTest.java
    2. SchematronTest.java
    3. SchematronMergeTest.java
- **Package**
  1. MergeConfigure.java
  2. Schematron.java
  3. SchematronMerge.java
  4. MergeConfigureTest.java
  5. SchematronTest.java
  6. SchematronMergeTest.java
  7. Jdom 2.6 or higher
- **Verify**
- **Install**
- **Deploy**

# Initial Design (*Whiteboarding*)



# A Note about an Alternate Rep (Non Property file Based)

Shon,

I have attached a new version of the schematron merge project. After you extract the attached archive, run Maven as you normally would from the command line:

`mvn clean install`

POM file Dependency example  
(Note from Matt Part 1)

After the build completes, the contents of the target directory should look like this:

```
.
├── archive-tmp
├── classes
├── generated-sources
├── maven-archiver
├── maven-status
├── schematron-merge-0.0.1-distribution.zip
├── schematron-merge-0.0.1.jar
├── schematron-merge-0.0.1-sources.jar
└── test-classes
```

The executable is in the schematron-merge-0.0.1-distribution.zip archive. To run the application, extract the contents of this archive to your machine, and inside the schematron-merge folder, run this command:

`java -jar schematron-merge.jar`



# A Note for an Alternate Rep (Non Property file Based)

POM  
file  
Depend  
ency  
example  
(Note  
from  
Matt  
Part 2)

If all goes well, you should see output indicating that the merge ran and the number

----

You'll more than likely notice that I made some changes in addition to the Maven co  
Changed how the schematron input and output files are specified for merging. Inste  
The actual configuration is in the file schematron-merge/src/main/resources/referen  
This configuration will be used by default, and can be "supplemented" with addition

```
java -jar schematron-merge.jar /path/to/supplemental/config/reference.conf
```

Note that using this behavior does not replace the default configuration, but merges  
Added a class named MergeDefinition that is used to represent each merge definitio  
Added a class named MergeConfig which replaces the original MergeConfigure class  
Modified the SchematronMerge class to use the new configuration classes  
Added some other utility classes  
For reference, here is the layout of the new project

schematron-merge

# POM versions

1. Jun 1 13:02 ./Documents/projects/mattHelp/qrda/QRDAValidator/pom.xml

First pom that builds dependencies in a phase

2. Jun 1 13:02 ./projects/mattHelp/qrda/schematronmerge/pom.xml

1. *Δ git Repo*

3. Jun 1 06:27 ./projects/qrda/schematronmerge/target/classes/META-INF/maven/com.esacinc.schematronmerge/Schematron-Merger/pom.xml

*Δ from git Repo*

4. May 31 13:53 ./projects/qrda/schematronmerge/pom.xml

*Cloned from git Repo*

1. May 19 18:01 ./qdraSchmatron/schematronmerge2/pom.xml

# References

- Maven
  - Introduction to Build Lifecycle
  - See <https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html>

# **PROGRESS/TO-DO LISTS**

# Progress and Tasks

- ShonVicks-MBP:Documents shonvick\$ find . -name "\*.java" -print | grep -i merge
- 2 versions (*Repos*)
  - Version 2 (Newer One)
  - adds
    1. comments
    2. property file validation
    3. logic to support coherent property file validation

./qdraSchmatron/schematronmerge2/src/main/java/schemaVerification/MergeConfigure.java  
./qdraSchmatron/schematronmerge2/src/main/java/schemaVerification/Schematron.java  
./qdraSchmatron/schematronmerge2/src/main/java/schemaVerification/SchematronMerge.java

- Version 1 (Older One)

./qdraSchmatron/schemaVerification/src/schemaVerification/MergeConfigure.java  
./qdraSchmatron/schemaVerification/src/schemaVerification/SchematronMerge.java  
./qdraSchmatronNew/SchematronMerge/src/Schematron.java

# Progress and Tasks

- Download maven (homebrew)
- Get m2eclipse up and running
- Generate POM
- Push to remote Repo
- Compare to Dan's
- Understand Dan's
- Read Maven tutorial
- Get Git / Eclipse Integrated
  - (EGit downloaded — to be installed)
    - Use command line for now
- Find example file — looked for examples
  - Got example from Matt
- Implement extended properties as per Michael spec
  - Research existing implementations
    - Eproperties and Apache Common Extensions evaluated
    - Eproperties chosen b/c its closest to current implementation
    - Implement using tool choice
- Implement Use Matt's example

# REFERENCES

# Maven References

- Introduction to Build Lifecycle

- See <https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html>



# Merge Tool References

- Location of the Repo
  - <https://github.com/esacinc/qrda/tree/master/schematronmerge>
- How to merge two XML's with JDOM?
  - <https://community.oracle.com/thread/1626390?start=0&tstart=0>
  - <https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=jdom%20copying%20an%20xml%20file>
- JDOM2 A Primer
  - <https://github.com/hunterhacker/jdom/wiki/JDOM2-A-Primer>
- Testing framework using Maven, TestNG and Webdriver
  - <https://www.leaseweb.com/labs/2014/03/testing-framework-using-maven-testng-webdriver/>

# Builder,Maven References

- Testing framework using Maven, TestNG and Webdriver
  - <https://www.leaseweb.com/labs/2014/03/testing-framework-using-maven-testng-webdriver>
- POM Reference
  - [http://maven.apache.org/pom.html#The Basics](http://maven.apache.org/pom.html#The_Basics)

# Progress and Tasks

# Gradle (Building) References

- Gradle xtext plugin
  - <https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=gradle%20xtext%20plugin>

# Git References

- Atlassian Git Tutorial
  - <https://www.atlassian.com/git/tutorials/rewriting-history>
- Github Repo for Validation
  - <https://github.com/esacinc/qrda>
- Use Git or checkout with SVN using the web URL
  - <https://github.com/esacinc/qrda.git>