Bullet Proofing Your Build Chain Using Open Source Tools

Presentation on using SCons, Schematron and Cruise Control in the build pipeline.

By A.E.Bailey, CTO, Tantalus.
Credit – N. McVeity, Lead Programmer, Tantalus.

email:andrew@tantalus.com.au

Overview

- Disclaimer – case study - depth (not tutorial, really overview)
- Background
- Build pipeline overview
- XML / XSLT overview
- Introduce tool set, backwards
  - CruiseControl
  - Schematron
  - SCons
- Conclusion
- Questions

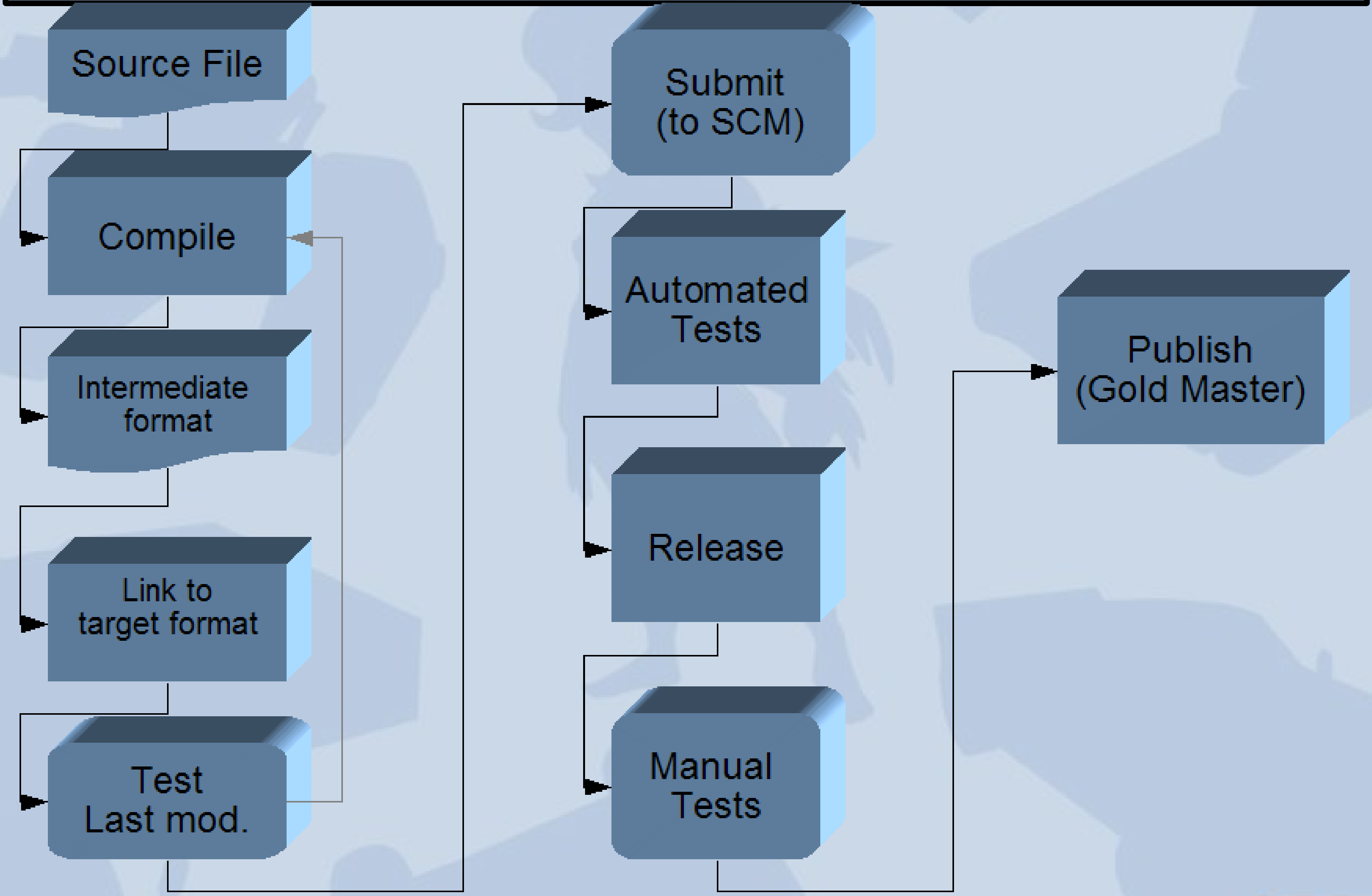
Background

- GCAP 06 round table
- Old Project Post-mortem
- New Project Pre-production

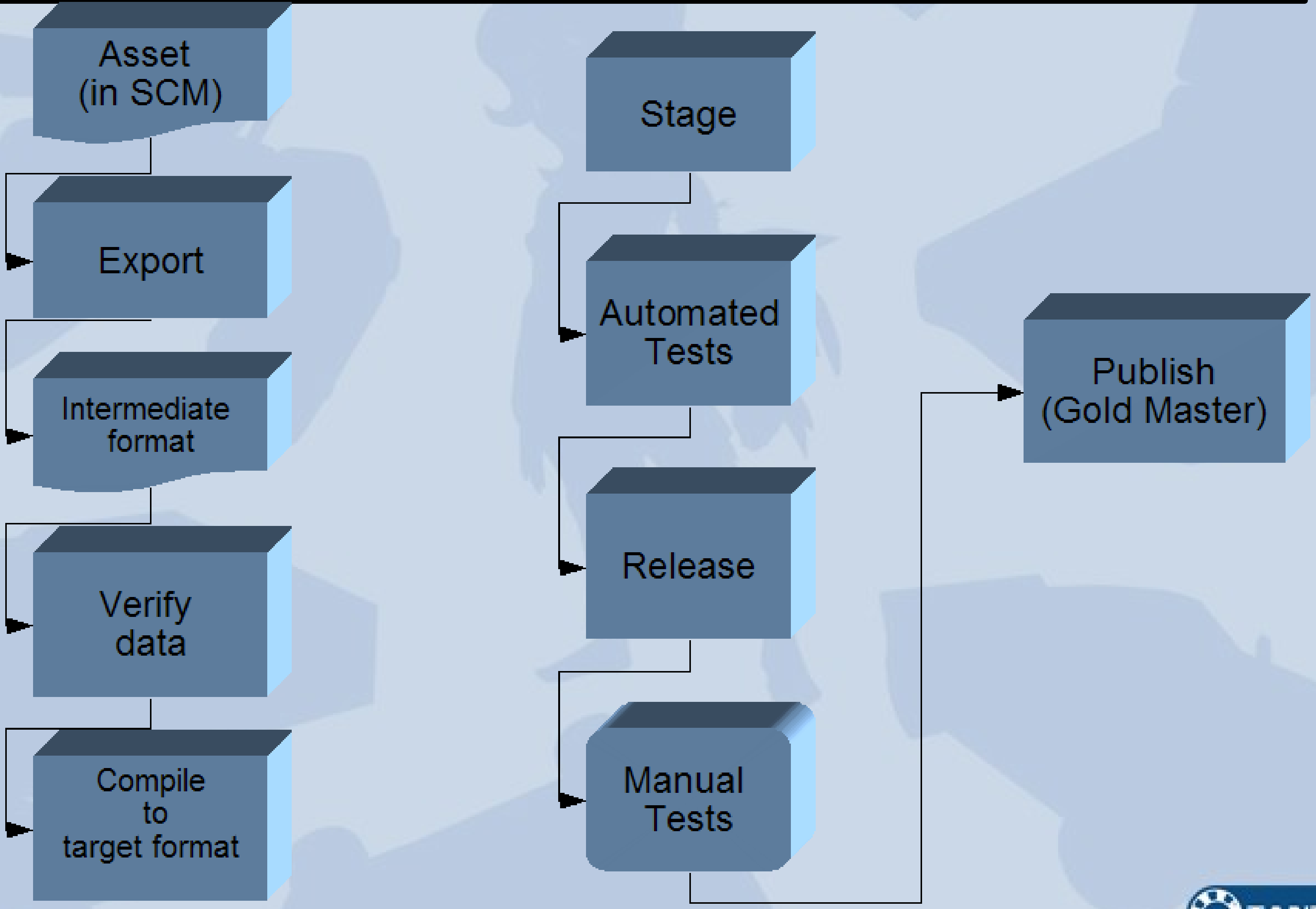
Build pipeline overview

- *The build pipeline is a series of builds, each performing some specific task. The output of one build becomes the input of the next.*

Code Build pipeline example



Art Build pipeline example



XML

- The Extensible Markup Language (XML) is a general-purpose markup language. It is classified as an extensible language because it allows its users to define their own tags. Its primary purpose is to facilitate the sharing of structured data across different information systems, particularly via the Internet.[2] It is used both to encode documents and serialize data. *(http://en.wikipedia.org/wiki/XML)*
- Lots of editing tool support, from MS-IDE to freeware.
- Easy to write basic XML, even without help of a library, using just plain text.
- A number of reader and parsing technologies available.
- Ever growing application usage of XML.

XSLT & XPath

- Extensible Stylesheet Language Transformations (XSLT) is an XML-based language used for the transformation of XML documents into other XML or "human-readable" documents
- XPath (XML Path Language) is an expression language for addressing portions of an XML document, or for computing values (strings, numbers, or boolean values) based on the content of an XML document.

CruiseControl

- URL: http://cruisecontrol.sourceforge.net/
- What is?
- Why do it?
- How does it work?
  - Server Configuration – Build Loop
  - Reporting
  - Client Dashboard
- .Net version URL: http://confluence.public.thoughtworks.org/display/CCNET/Welcome+to+CruiseControl.NET

CruiseControl.Net

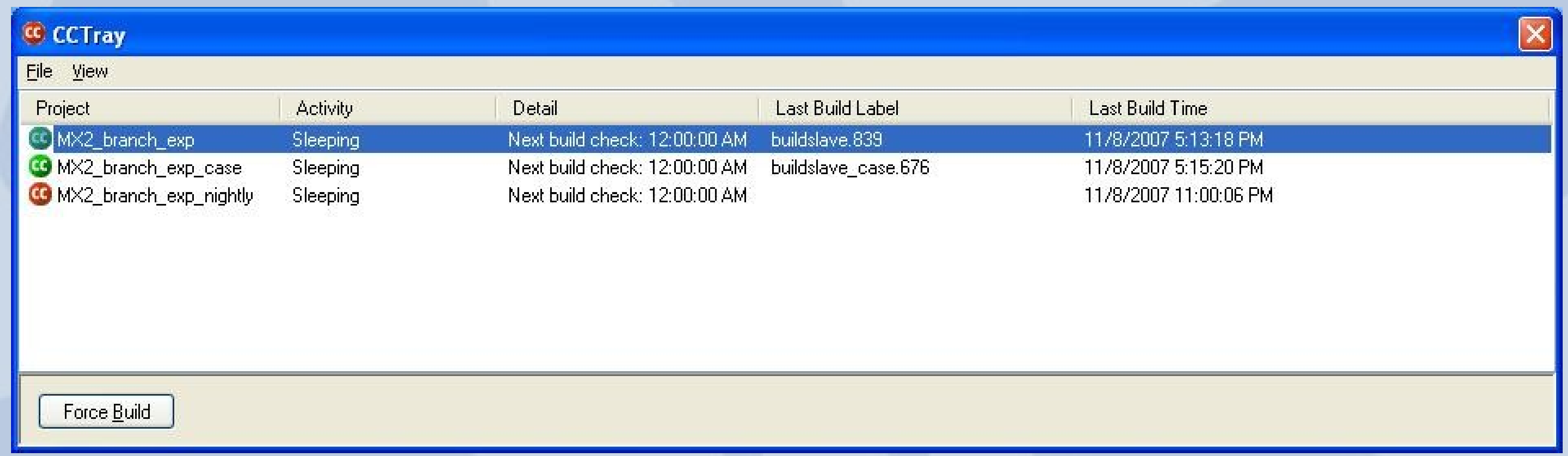
- Features
  - XML configuration.
  - Central repository interface.
    - Sync
    - Labelling
    - Blame game
  - XML reports
  - Chaining. Build to regression tests.

CruiseControl.Net

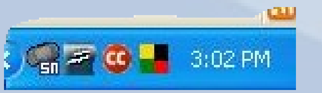
```
<cruisecontrol>
  <project name="cars2 branch-dev" workingDirectory="c:\projects\cars2\branch-dev">
    <webURL>\\buildslave1\logs\cars2\branch-dev</webURL>
    <modificationDelaySeconds>10</modificationDelaySeconds>
    <sourcecontrol type="p4">
      <view>//depot/cars2/branch-dev/...</view>
      <client>buildslave_xp</client>
    </sourcecontrol>
    <tasks>
        <exec>
          <executable>.\tools\bin\make.exe</executable>
          <baseDirectory>c:\projects\cars2\branch-dev</baseDirectory>
          <buildArgs>BUILD=DEBUG</buildArgs>
          <buildTimeoutSeconds>3600</buildTimeoutSeconds>
        </exec>
        <exec>
          <executable>.\tools\bin\make.exe</executable>
          <baseDirectory>c:\projects\cars2\branch-dev</baseDirectory>
          <buildArgs>BUILD=RETAIL unit_tests</buildArgs>
          <buildTimeoutSeconds>3600</buildTimeoutSeconds>
        </exec>
        <exec>
            <executable>p4.exe</executable>
            <buildArgs>labelsync -l cars2-buildslave-success //...</buildArgs>
        </exec>
    </tasks>
    <publishers>
      <xmllogger logDir="c:\logs\cars2\branch-dev" />
    </publishers>
  </project>
</cruisecontrol>
```

CruiseControl.Net

Client experience



Project	Activity	Detail	Last Build Label	Last Build Time
MX2_branch_exp	Sleeping	Next build check: 12:00:00 AM	buildslave.839	11/8/2007 5:13:18 PM
MX2_branch_exp_case	Sleeping	Next build check: 12:00:00 AM	buildslave_case.676	11/8/2007 5:15:20 PM
MX2_branch_exp_nightly	Sleeping	Next build check: 12:00:00 AM		11/8/2007 11:00:06 PM

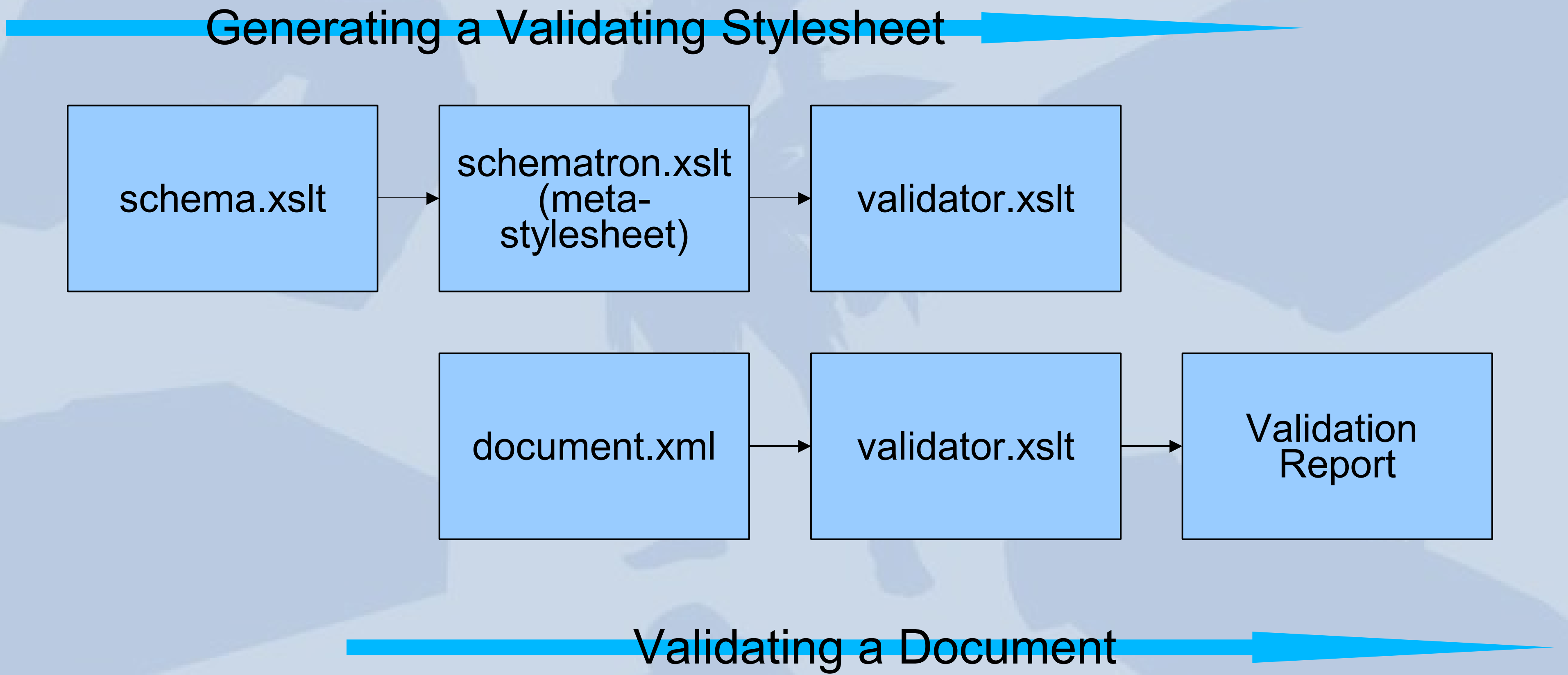


Schematron

- URL: http://www.schematron.com/
- What is it?
- Why do it?
- How does it work?

Schematron

- Data Flow



Generating a Validating Stylesheet

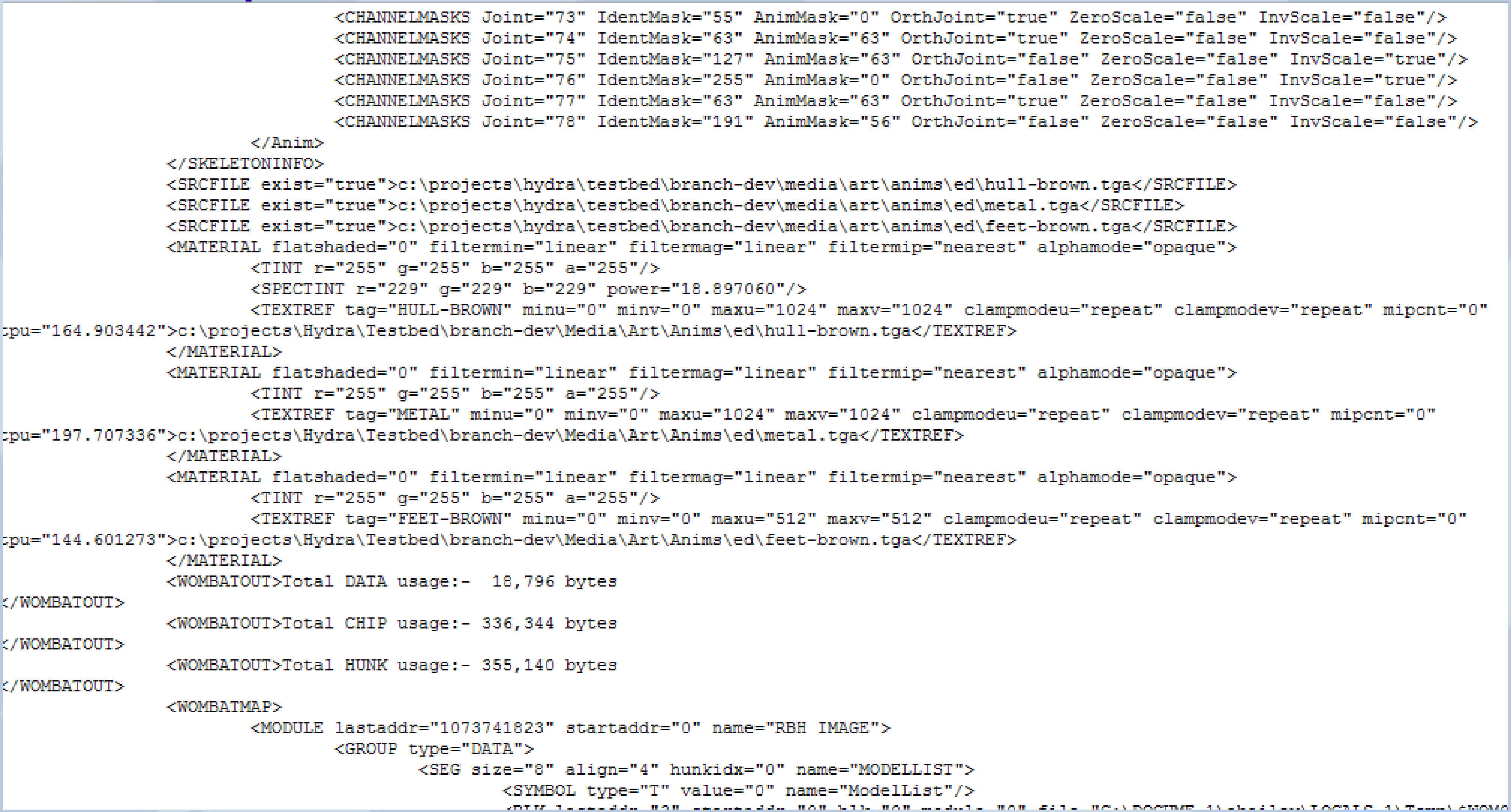
schema.xslt → schematron.xslt (meta-stylesheet) → validator.xslt

document.xml → validator.xslt → Validation Report

Validating a Document

Schematron

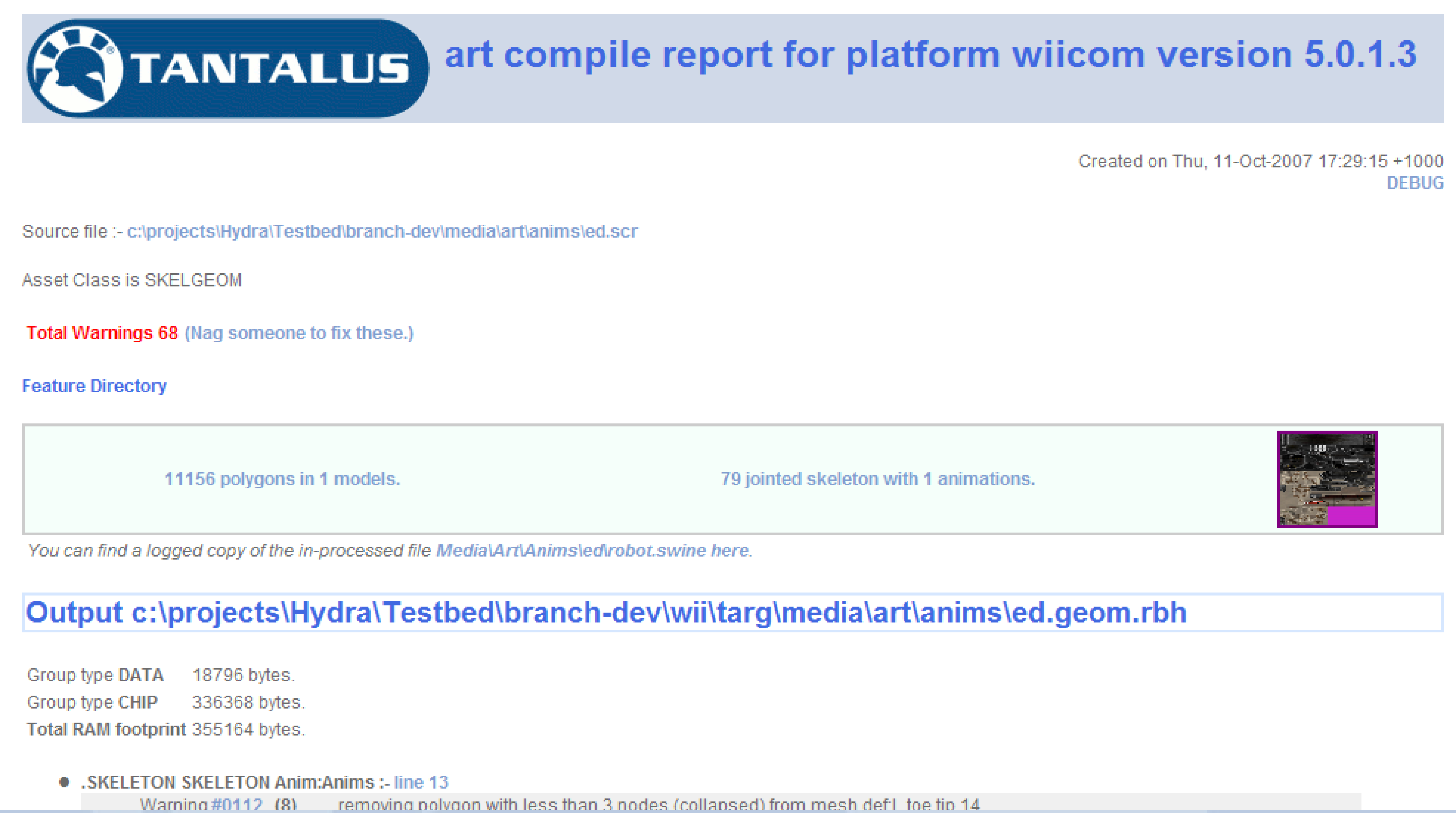
- Example



```
                                <CHANNELMASKS Joint="73" IdentMask="55" AnimMask="0" OrthJoint="true" ZeroScale="false" InvScale="false"/>
                                <CHANNELMASKS Joint="74" IdentMask="63" AnimMask="63" OrthJoint="true" ZeroScale="false" InvScale="false"/>
                                <CHANNELMASKS Joint="75" IdentMask="127" AnimMask="63" OrthJoint="false" ZeroScale="false" InvScale="true"/>
                                <CHANNELMASKS Joint="76" IdentMask="255" AnimMask="0" OrthJoint="false" ZeroScale="false" InvScale="true"/>
                                <CHANNELMASKS Joint="77" IdentMask="63" AnimMask="63" OrthJoint="true" ZeroScale="false" InvScale="false"/>
                                <CHANNELMASKS Joint="78" IdentMask="191" AnimMask="56" OrthJoint="false" ZeroScale="false" InvScale="false"/>
                        </Anim>
                </SKELETONINFO>
                <SRCFILE exist="true">c:\projects\hydra\testbed\branch-dev\media\art\anims\ed\hull-brown.tga</SRCFILE>
                <SRCFILE exist="true">c:\projects\hydra\testbed\branch-dev\media\art\anims\ed\metal.tga</SRCFILE>
                <SRCFILE exist="true">c:\projects\hydra\testbed\branch-dev\media\art\anims\ed\feet-brown.tga</SRCFILE>
                <MATERIAL flatshaded="0" filtermin="linear" filtermag="linear" filtermip="nearest" alphamode="opaque">
                        <TINT r="255" g="255" b="255" a="255"/>
                        <SPECTINT r="229" g="229" b="229" power="18.897060"/>
                        <TEXTREF tag="HULL-BROWN" minu="0" minv="0" maxu="1024" maxv="1024" clampmodeu="repeat" clampmodev="repeat" mipcnt="0"
cpu="164.903442">c:\projects\Hydra\Testbed\branch-dev\Media\Art\Anims\ed\hull-brown.tga</TEXTREF>
                </MATERIAL>
                <MATERIAL flatshaded="0" filtermin="linear" filtermag="linear" filtermip="nearest" alphamode="opaque">
                        <TINT r="255" g="255" b="255" a="255"/>
                        <TEXTREF tag="METAL" minu="0" minv="0" maxu="1024" maxv="1024" clampmodeu="repeat" clampmodev="repeat" mipcnt="0"
cpu="197.707336">c:\projects\Hydra\Testbed\branch-dev\Media\Art\Anims\ed\metal.tga</TEXTREF>
                </MATERIAL>
                <MATERIAL flatshaded="0" filtermin="linear" filtermag="linear" filtermip="nearest" alphamode="opaque">
                        <TINT r="255" g="255" b="255" a="255"/>
                        <TEXTREF tag="FEET-BROWN" minu="0" minv="0" maxu="512" maxv="512" clampmodeu="repeat" clampmodev="repeat" mipcnt="0"
cpu="144.601273">c:\projects\Hydra\Testbed\branch-dev\Media\Art\Anims\ed\feet-brown.tga</TEXTREF>
                </MATERIAL>
                <WOMBATOUT>Total DATA usage:-  18,796 bytes
</WOMBATOUT>
                <WOMBATOUT>Total CHIP usage:- 336,344 bytes
</WOMBATOUT>
                <WOMBATOUT>Total HUNK usage:- 355,140 bytes
</WOMBATOUT>
                <WOMBATMAP>
                        <MODULE lastaddr="1073741823" startaddr="0" name="RBH IMAGE">
                                <GROUP type="DATA">
                                        <SEG size="8" align="4" hunkidx="0" name="MODELLIST">
                                                <SYMBOL type="T" value="0" name="ModelList"/>
```

Schematron

- Example



art compile report for platform wiicom version 5.0.1.3

Created on Thu, 11-Oct-2007 17:29:15 +1000
DEBUG

Source file :- c:\projects\Hydra\Testbed\branch-dev\media\art\anims\ed.scr

Asset Class is SKELGEOM

Total Warnings 68 (Nag someone to fix these.)

Feature Directory

11156 polygons in 1 models. 79 jointed skeleton with 1 animations.

You can find a logged copy of the in-processed file Media\Art\Anims\ed\robot.swine here.

Output c:\projects\Hydra\Testbed\branch-dev\wii\targ\media\art\anims\ed.geom.rbh

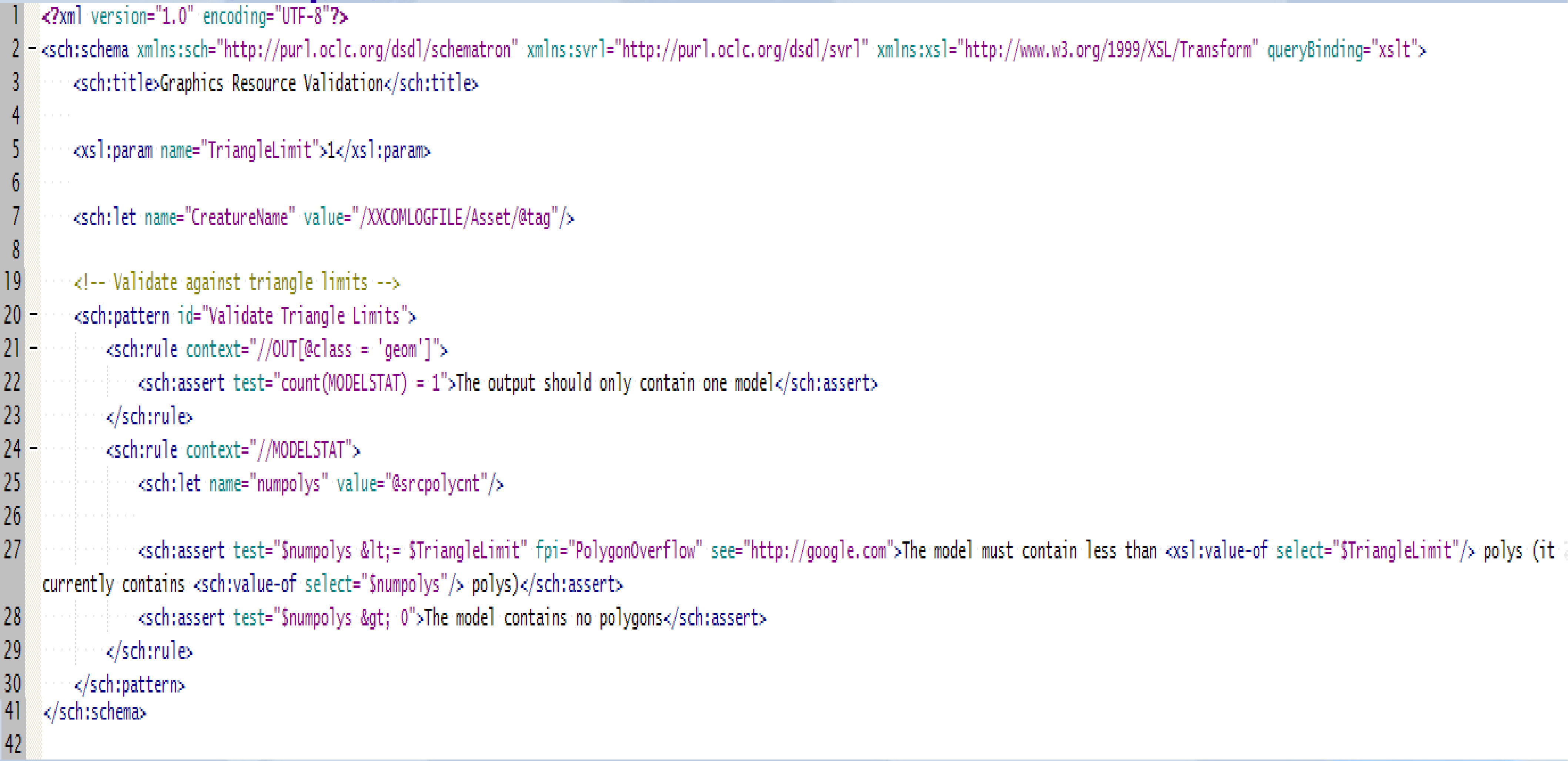
```
Group type DATA      18796 bytes.
Group type CHIP      336368 bytes.
Total RAM footprint  355164 bytes.
```

- **.SKELETON SKELETON Anim:Anims :-** line 13

```
    Warning #0112  (8)       removing polygon with less than 3 nodes (collapsed) from mesh def:l_toe tip 14
```

Schematron

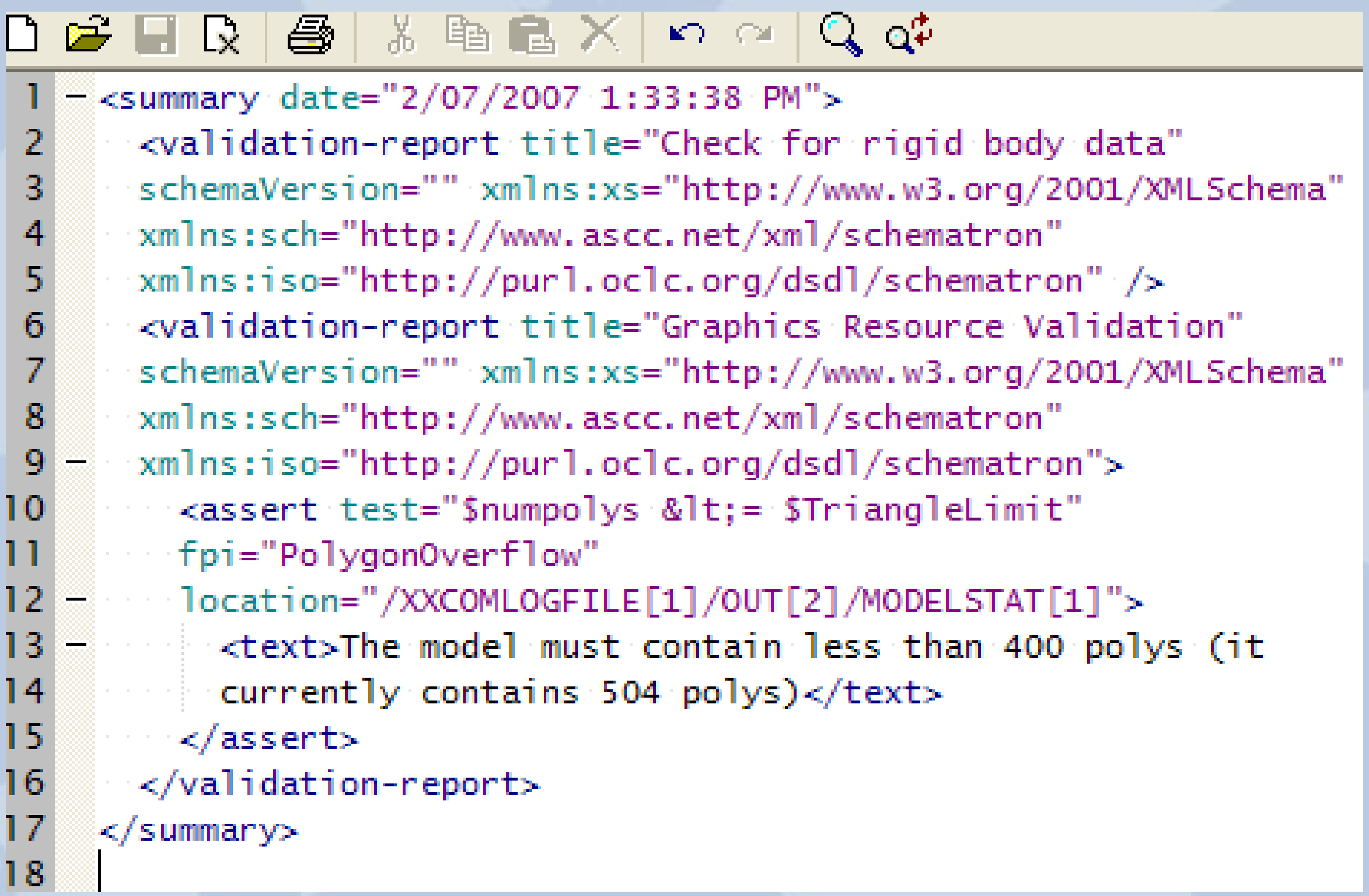
- Example



```
<?xml version="1.0" encoding="UTF-8"?>
<sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron" xmlns:svrl="http://purl.oclc.org/dsdl/svrl" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" queryBinding="xslt">
    <sch:title>Graphics Resource Validation</sch:title>

    <xsl:param name="TriangleLimit">1</xsl:param>

    <sch:let name="CreatureName" value="/XXCOMLOGFILE/Asset/@tag"/>

    <!-- Validate against triangle limits -->
    <sch:pattern id="Validate Triangle Limits">
        <sch:rule context="//OUT[@class = 'geom']">
            <sch:assert test="count(MODELSTAT) = 1">The output should only contain one model</sch:assert>
        </sch:rule>
        <sch:rule context="//MODELSTAT">
            <sch:let name="numpolys" value="@srcpolycnt"/>

            <sch:assert test="$numpolys &lt;= $TriangleLimit" fpi="PolygonOverflow" see="http://google.com">The model must contain less than <xsl:value-of select="$TriangleLimit"/> polys (it
currently contains <sch:value-of select="$numpolys"/> polys)</sch:assert>
            <sch:assert test="$numpolys &gt; 0">The model contains no polygons</sch:assert>
        </sch:rule>
    </sch:pattern>
</sch:schema>
```

Schematron

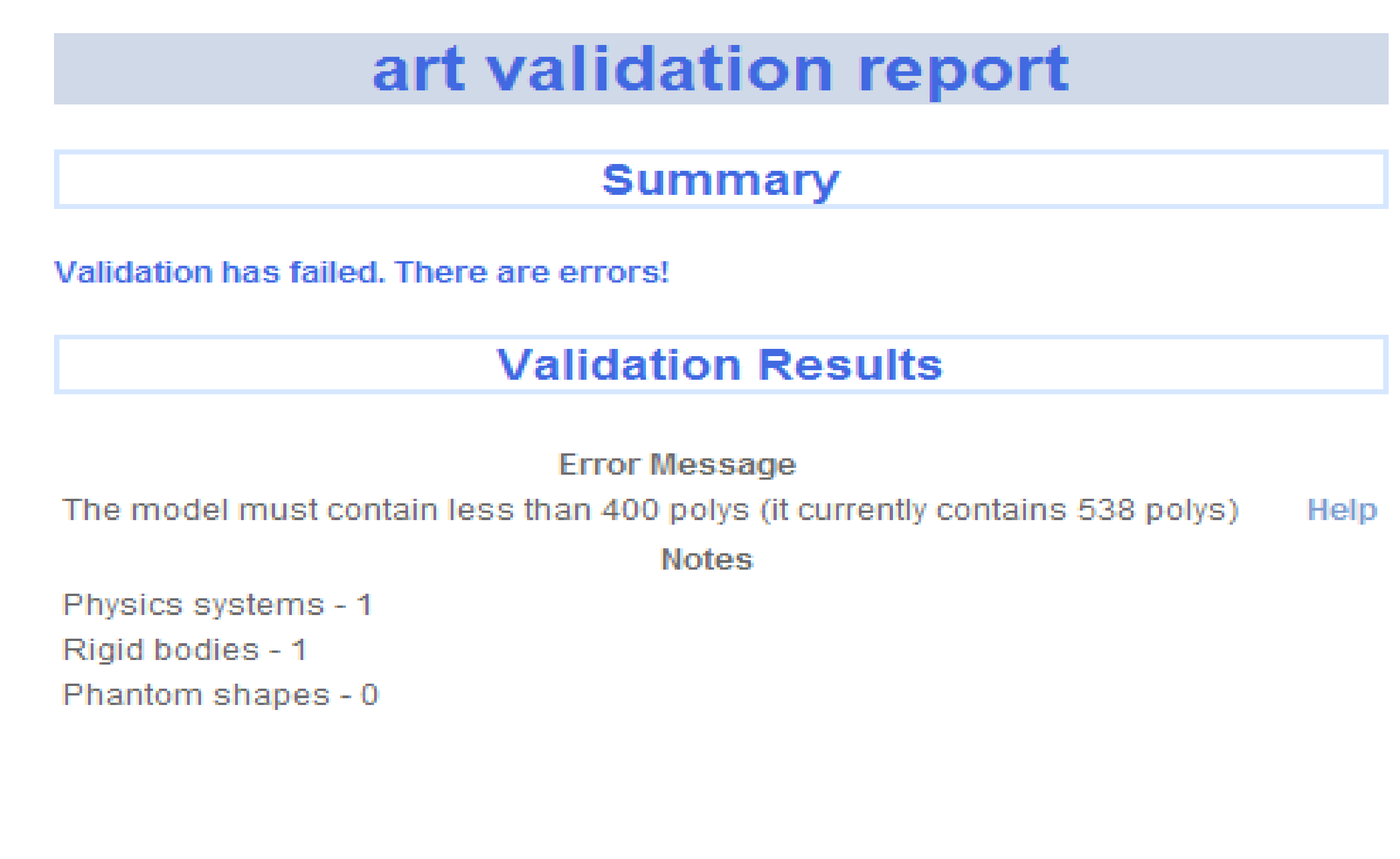
- Example



```
<summary date="2/07/2007 1:33:38 PM">
  <validation-report title="Check for rigid body data"
  schemaVersion="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sch="http://www.ascc.net/xml/schematron"
  xmlns:iso="http://purl.oclc.org/dsdl/schematron" />
  <validation-report title="Graphics Resource Validation"
  schemaVersion="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sch="http://www.ascc.net/xml/schematron"
  xmlns:iso="http://purl.oclc.org/dsdl/schematron">
    <assert test="$numpolys &lt;= $TriangleLimit"
    fpi="PolygonOverflow"
    location="/XXCOMLOGFILE[1]/OUT[2]/MODELSTAT[1]">
      <text>The model must contain less than 400 polys (it
      currently contains 504 polys)</text>
    </assert>
  </validation-report>
</summary>
```

Schematron

- Example



art validation report

Summary

Validation has failed. There are errors!

Validation Results

Error Message

The model must contain less than 400 polys (it currently contains 538 polys) Help

Notes

Physics systems - 1

Rigid bodies - 1

Phantom shapes - 0

SCons

- URL: http://www.scons.org/
- Build tool
  - *The SCons utility builds software (or other files) by determining which component pieces must be rebuilt and executing the necessary commands to rebuild them.*
  - Replacement for MAKE, but oh so much more.
  - Written in Python (http://www.python.org/),
  - And 'scripted' in Python.

SCons Features

- Main features from Tantalus point of view.
  - MD5 instead of timestamps.
  - Build Cache
    - Shared build.
  - Automatic and extensible dependency generation.
    - Very important when most of your tool-set is custom.
  - Build processes can be Python functions, as well as external processes.
  - Multiple targets from a build process.
  - 'Tools' allow reuse.

SCons Features

- Other features.
  - Direct build from central repositories.
  - Parallel builds.

SCons

- Issues
  - Speed
    - Slow startup time, compared to make.
      - However, false economy, when cache taken into account.
      - Subset targets, caching.
    - Python profiler plus –debug with 14 metrics.
  - Not Final (version < 1.0)
    - Bugs - support.
    - Upgrading to new version maintenance.

SCons

- Case Study Results
  - Size of script.

```
–     lines    code     doc  comment   blank  file
–      1617    1028      89      138     362  SConstruct
–      1418     897      81      107     333  tantalus_undisclosedproject.py
–       112      56      36        0      20  tantalus_fmod.py
–        48      23      12        2      11  tantalus_havok.py
–        37      18      12        0       7  tantalus_lua.py
–       897     526      84       92     195  tantalus_maya.py
–       227     139      17       20      51  tantalus_mercury.py
–       114      64       7       14      29  tantalus_revolution.py
–        90      57       7        1      25  tantalus_schematron.py
–       110      68      13        2      27  tantalus_utils.py
–       105      62      17        3      23  tantalus_xml.py
–      4775    2938     375      379    1083  total
```

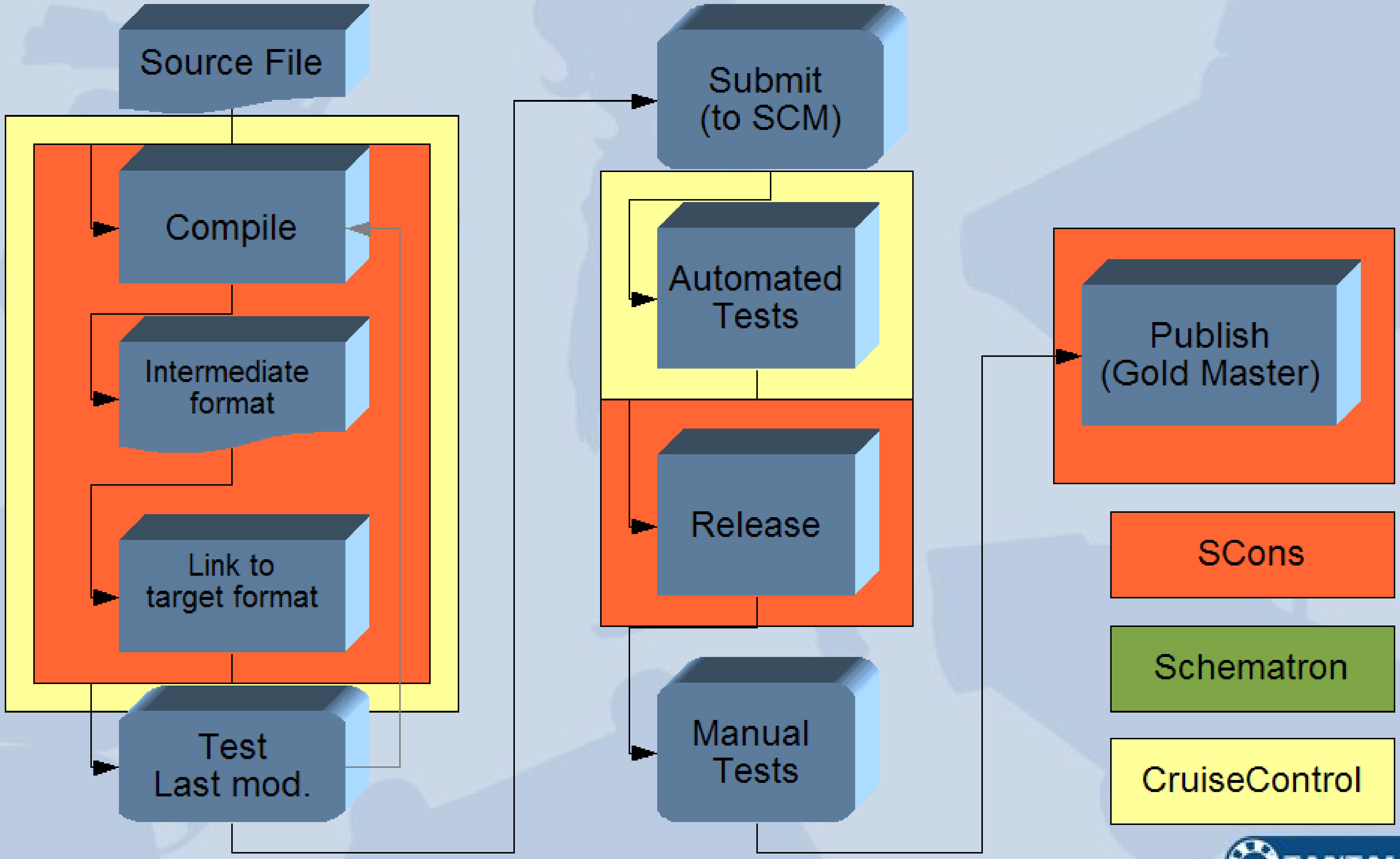
SCons

- Case Study Results
  - Python mania.
    - One large process.
      - Issues with this.
        - Memory leaks.
    - Reduces steps, hence build rules, dependency tree etc.

SCons

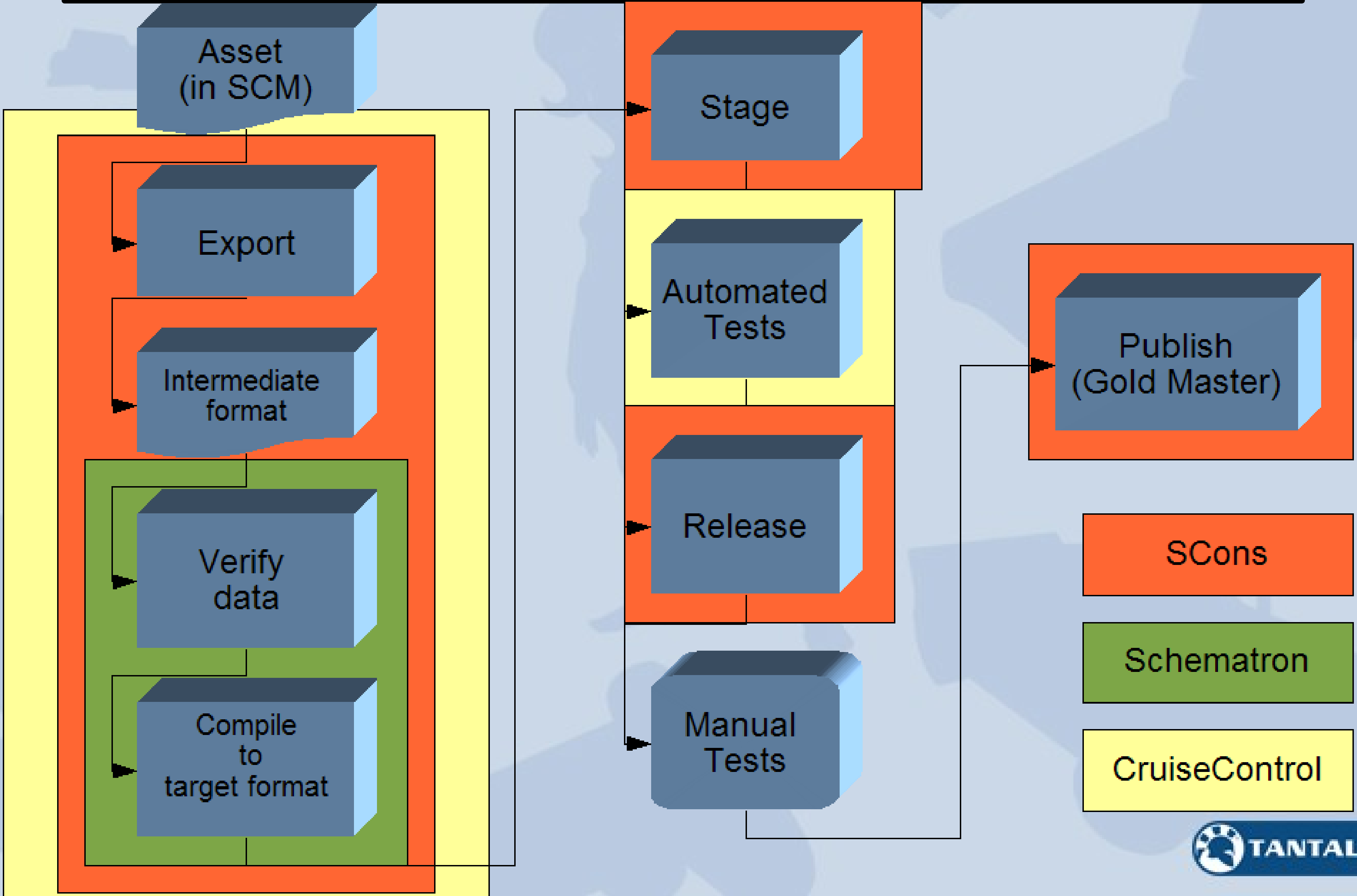
- Case Study Results
  - Cruise Control 'Integration'.
    - Distributed / parallel build.
  - Large projects
    - From monolithic script to reusable modules.

Code Build pipeline



TANTALUS

Art Build pipeline



Asset (in SCM)

Export

Intermediate format

Verify data

Compile to target format

Stage

Automated Tests

Release

Manual Tests

Publish (Gold Master)

🟥 SCons

🟩 Schematron

🟨 CruiseControl

Conclusion

- CruiseControl
  - Continuous Integration really is a must, and this is a really good way to do it.
  - Easy to set up.
- Schematron
  - As of writing this has been set up and is part of the pipeline, but not really given a rich set of rules yet.
  - At the start of a project, it can be a bit harsh enforcing rules which are arbitrary at the time.

Conclusion

- SCons
  - A larger task than originally envisioned.
    - Learning curve, more jargon, we wrote the SConstruct file twice. Included learning Python.
  - Distributed, parallel build system for free.
  - Improved our build pipeline:-
    - Fully integrated
    - Reusable
    - Documented
- Python
  - Custom tool set integration we didn't see coming.

Questions

- URL's
  - http://en.wikipedia.org/wiki/XML
  - http://en.wikipedia.org/wiki/XSLT
  - http://cruisecontrol.sourceforge.net/
  - http://confluence.public.thoughtworks.org/display/CCNET/Welcome+to+CruiseControl.NET
  - http://www.schematron.com/
  - http://www.scons.org/
  - http://www.python.org/