

# Automated Management of Verification Waivers

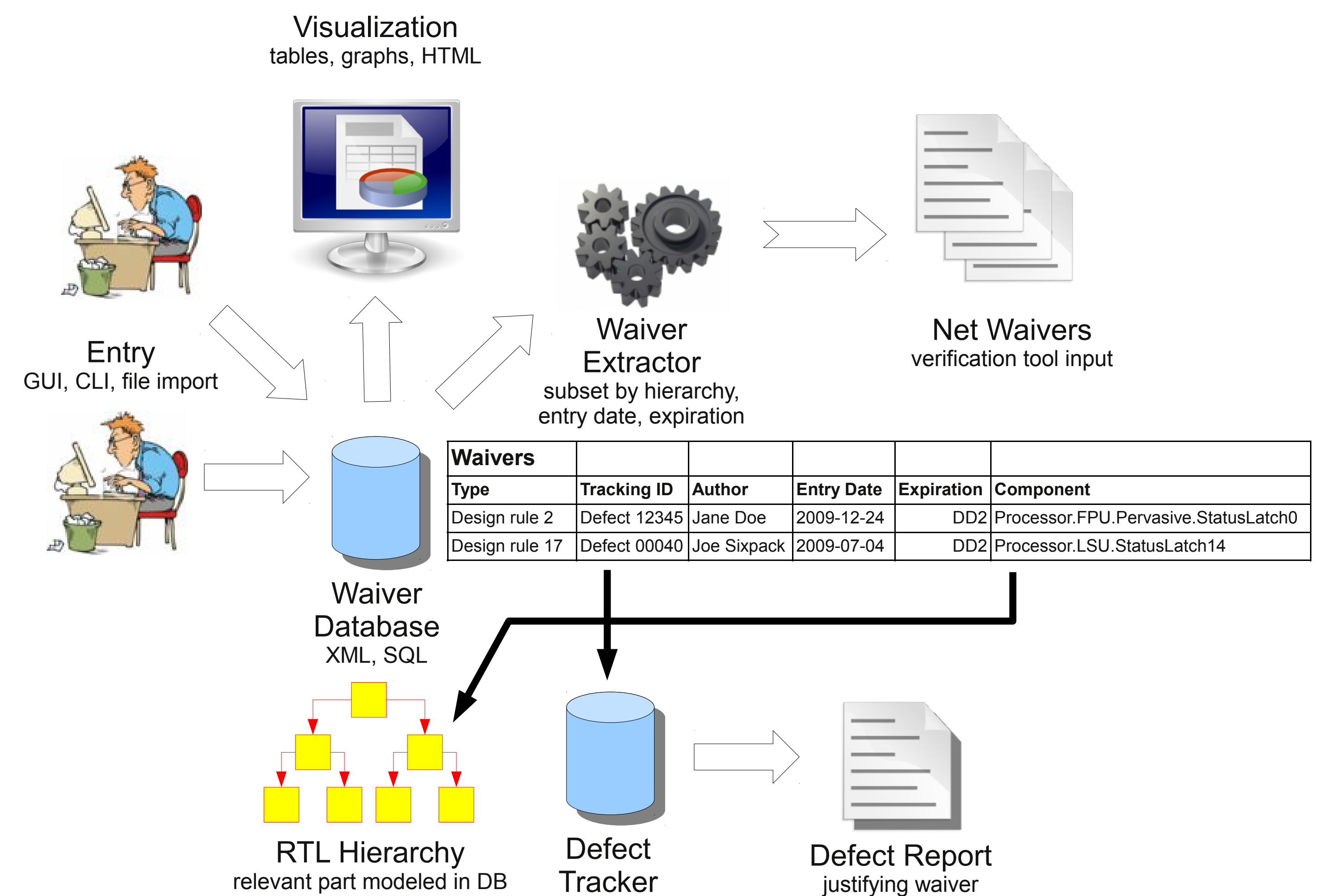
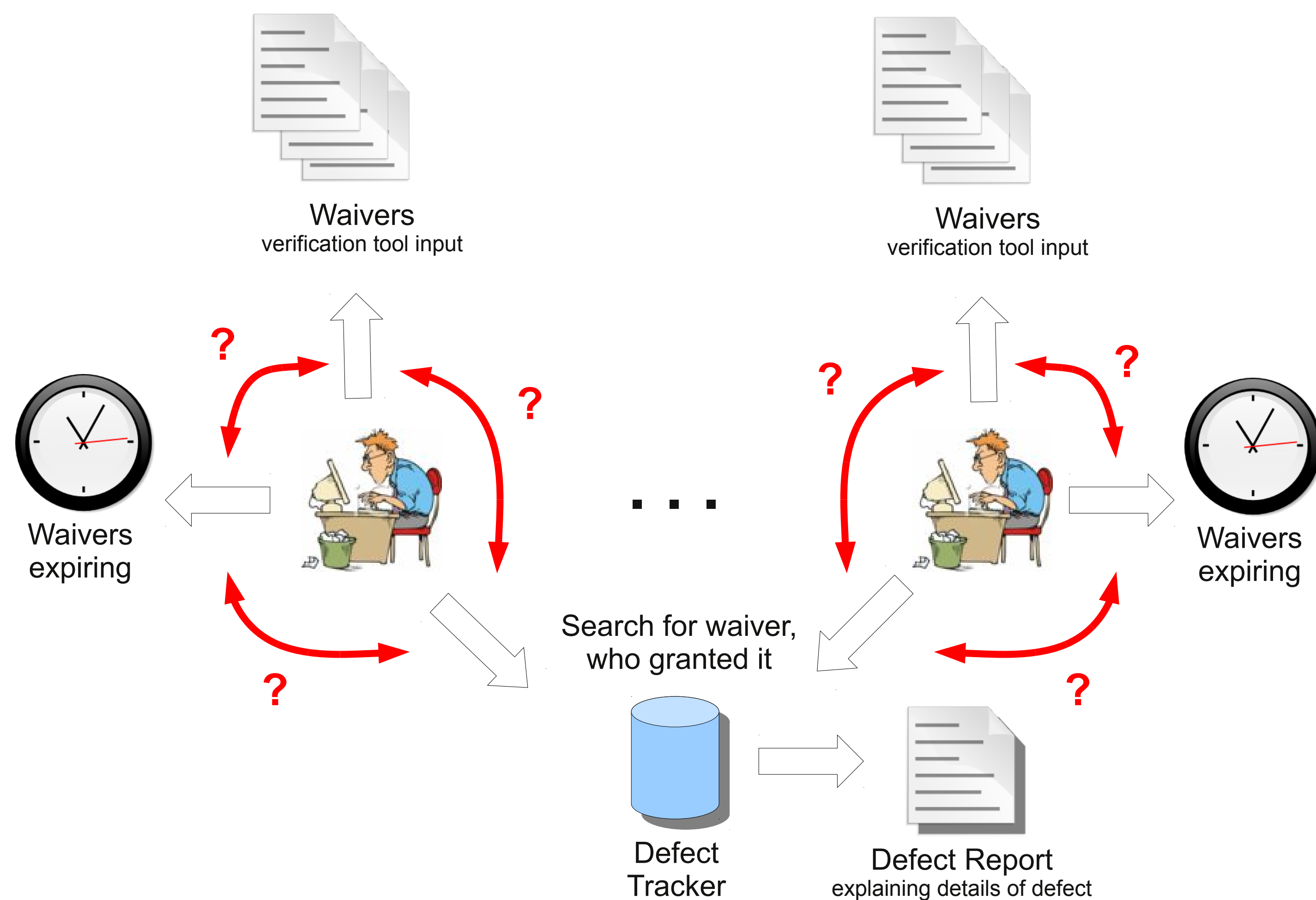
Gero Dittmann  
 IBM STG, Research and Development  
 Böblingen, Germany  
 gdittma@de.ibm.com

## Problem: Distributed, Unstructured Information

- Verification: large number of waivers of various types
- Information often distributed across many files in various locations, owned by different engineers.
- Waivers need to be propagated up the entity hierarchy.
- Redundancies across files, difficult to keep in sync
- Hard to keep track:
  - Which defect report documents, justifies the waiver?
  - Who has granted the waiver and when?
  - Is the waiver still valid or has it expired?
  - Has this error been waived before?
- Lingering but expired waivers mask errors
- Current situation: manual processes
  - very tedious and time-consuming
  - error-prone
- [US 2005/0188336]: waiver database in the loop
  - stores no meta-data at all

## Solution: Centralized, Structured Database

- Structured according to chip hierarchy (RTL)
- Associates waivers with elements in the hierarchy
- Stores meta-data with each waiver:
  - Type
  - Granting person
  - Granting date, time
  - Expiration (date or release name)
  - Defect report ID
- Waiver lists extracted automatically from database
  - No more problems hidden by expired waivers
  - Filters out-of-scope waivers
  - Prunes hierarchy names for sub-component verification
- Correlates waivers and defects: no more searching.
- Documents waiver responsibility.
- Tracks number of waivers over time for project management.
- Implementations: XML files and an SQL database



## CheXML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<chexml chip="MyProcessor" project="NextServer">
  <entity name="chip">
    <entity name="chiplet1" />
    <entity name="chiplet2">
      <waiver type="designRule17" tracker="123" expiration="" author="Joe Sixpack" date="2011-04-01" />
    <entity name="FPU">
      <waiver type="unconnected" tracker="456" expiration="DD1" author="Jane Doe" date="2010-12-24" />
    </entity>
    <entity name="LSU" />
  </entity>
</chexml>
```

## XML Tool Support

- Parsers available for any programming language
- Using simple XSLT stylesheets for transformations, e.g., waiver extractor
- XPATH: grep that understands hierarchy
- Easily extended, imported into other tools
- Editor support through schemas

## SQL Mapping

- Two tables: entities, waivers
- Entity hierarchy modeled as adjacency list (pointers to parent).
- Waivers associated with entities by foreign keys.
- Waiver properties stored in columns of the waiver table.