

**Compliance of SOAP/XML
Standards does not release from
extensive testing**

Holger Guhl

Senior Smalltalker at Georg Heeg eK

www.heeg.de



mailto: holger@heeg.de

In collaboration with

Stephan Melzig

stephan.melzig@globalfoundries.com

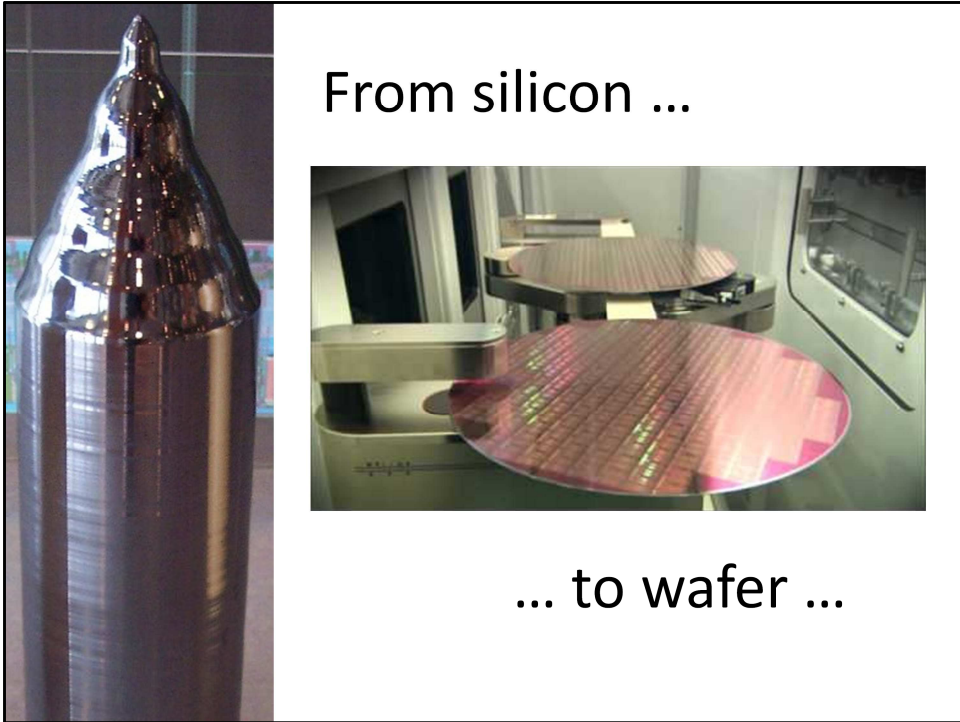
Member Technical Staff Factory Automation

GLOBALFOUNDRIES Dresden www.globalfoundries.com

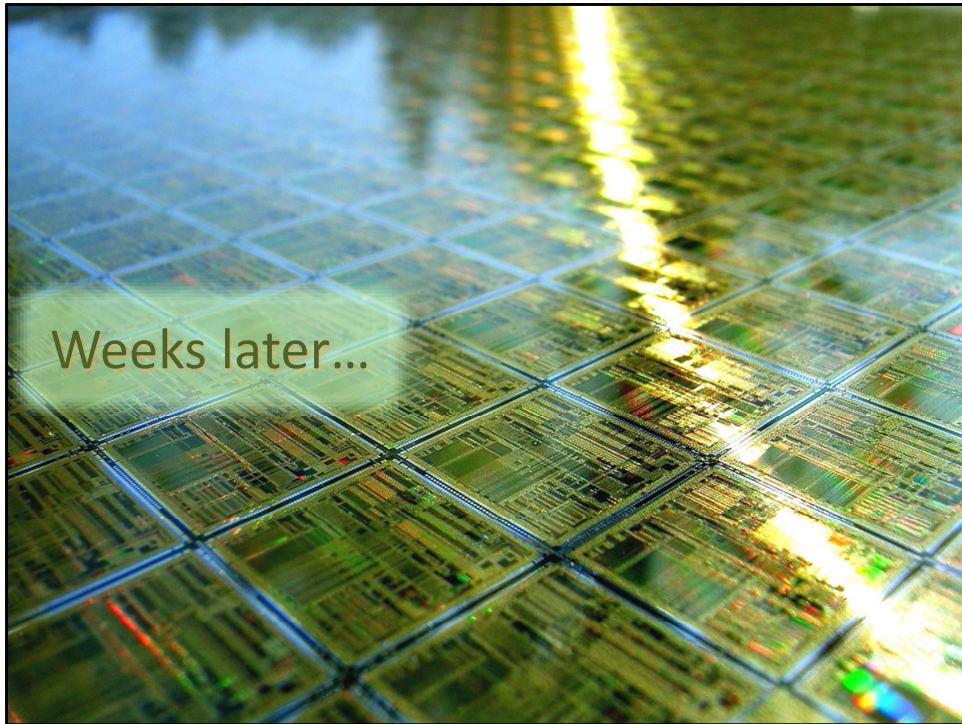


What is Globalfoundries? What do they do?

- Former AMD, since 2008 extracted from AMD as own company
- Semiconductor manufacturing, chip production for computers, tablets, phones, etc
- 80,000 300-mm wafers per month, equivalent to 180,000 200-mm wafers per month (<https://de.wikipedia.org/wiki/Globalfoundries>)
- Technology: 45 nm, 40 nm, 32 nm, 28 nm, SOI (silicon on insulator)
- Cleanrooms class T1-T3
- thousands of manufacturing tools for wafer/semiconductor processing (wafer sorter, stocker, transport system, furnace, implanter, polisher, litho cells, etc)



What does Globalfoundries do? Computer chip manufacturing.
Here: From Silicon to wafer...



What does Globalfoundries do? Computer chip manufacturing.

Here: Completely finished wafer. It takes weeks (months) to carefully build all semiconductor layers for a computer chip.



What does Globalfoundries do? Computer chip manufacturing.

Here: Cleanroom with huge production tool on the left, aisle full of control equipment, transportation units below the ceiling

About...



Migration to VisualWorks 7.10

Focus on SOAP/XML

About this talk

With this contribution we want to share **experiences** during **migration** of a **large industry project**. Globalfoundries has decided in 2014 to migrate their Baseline software to VisualWorks 7.10. While a lot of aspects had to be worked on, we want to focus on SOAP/XML features. The basic reason is that the Baseline makes heavy use of SOAP/XML. Second, this field turned out to be a challenge.

Originally, it was planned that **Stephan Melzig**, a project developer from the Globalfoundries team presents the talk. Unfortunately, he could not attend the conference. We agreed that I take over the presenter role. Nevertheless, we can promise authenticity because the author was part of the migration team.



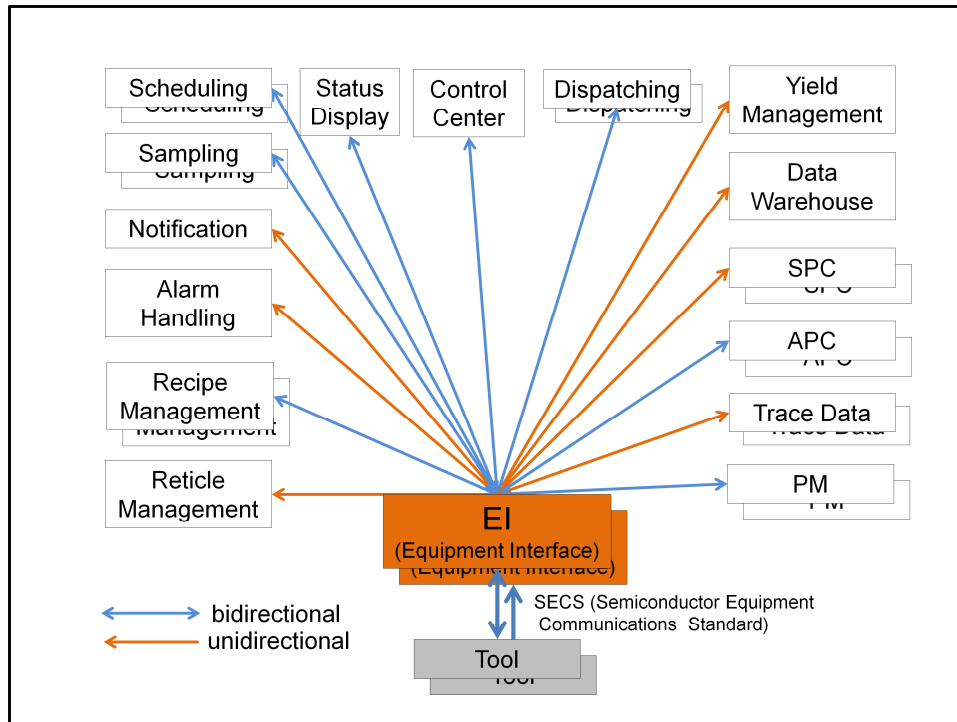
About Globalfoundries software environment

- Heterogeneous mixture of applications
- Large variety
- Fruitful, exotic, all flavors and „species“
- no common vendor
- many self-developed



About Globalfoundries software environment

- Supports semiconductor **industry needs**: Control of wafer processing
- Large **variety** of tools
- **Specialized** for its task
- **Pragmatic** and fruitful
- **No** gimmicks



Globalfoundries/Dresden - Overview

Explain Equipment Interface (EI), how is it integrated, main focus on:

- Number of connected systems
- Type and languages of other systems (c#, Java, ..)
- Transport means (MQ, HTTP)
- Number of EI Instances

Heterogeneous Applications

Various languages
(**VisualWorks**, Java, C#, script languages)

Communication via
SOAP/XML over MQ Series

Heterogeneous Applications from all kind of software development cultures.
Common concept: All communication via SOAP/XML, transmitted over MQ Series
(Websphere MQ)

What is...

Baseline?

Baseline is the Globalfoundries project title for a software that provides a framework for programming semiconductor and chip production machinery. Each physical machine is controlled by an **Equipment Interface (EI)** which is a machine type specific adaption of the Baseline framework. Simple adaptations can be accomplished by plain configuration, but usually, refined and extended behavior must be programmed as extension of Baseline classes.

Some figures:

EI Instances:

3200 instances in 2 sites:

- Dresden (Germany)
- Malta (US, New York State)

SOAP/WSDL Connections

- 23 systems addressed from EI
- 13 systems calling the EI
- 57 EI SOAP Services provided

Baseline Migration to VisualWorks 7.10

Project: Migration to VisualWorks 7.10

- Modernization, actual platform was VisualWorks 7.5, no longer supported by Cincom
- Done by Georg Heeg eK

Problems:

- Numerous interconnected systems, often no longer well understood (black boxes)
- Non-standard implementation of external systems (e.g. namespace problems with MDS)
- Fire and Forget Interfaces: No feedback in integration tests, Support on external side necessary
- Applying better standards or changes on Smalltalk/EI side creates problems on external side, e.g. because XML is parsed/processed „somehow“
- Not enough Unit Tests which are based on real scenarios
- Some changes in XML (as transmitted via comm channels) cannot be avoided by Unit Tests and require Integration Tests
- Heeg cannot do the VW Migration „autonomously“ due to these reasons



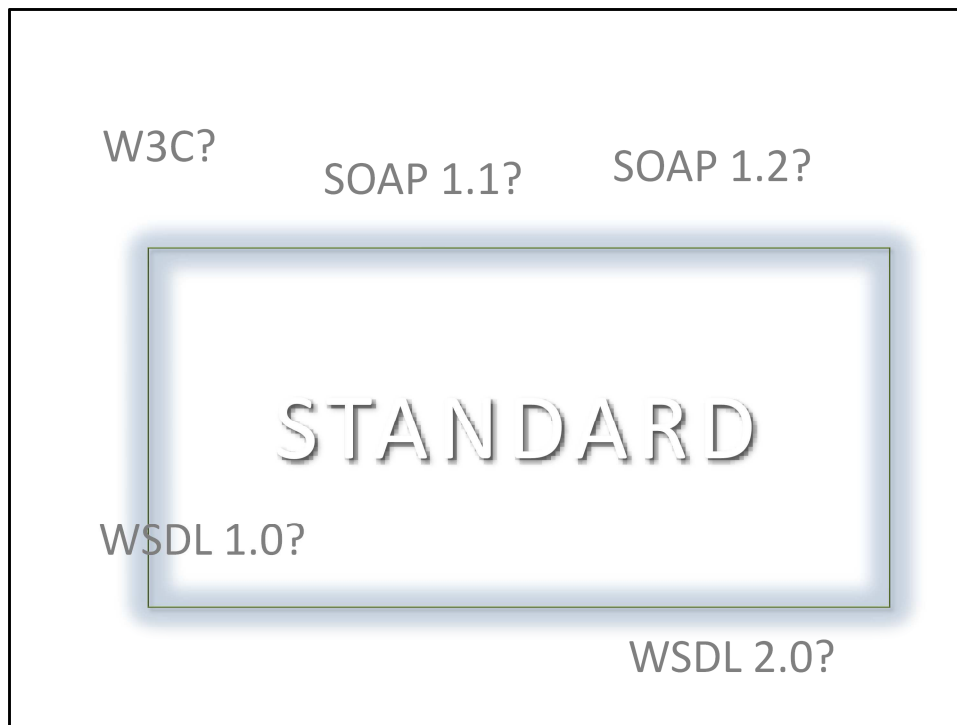
SOAP/XML is a



STANDARD

The promise of and objective behind XML standard:

- It allows to connect applications independent from the technology.
- Interfaces (API, features, services) are described with a Webservice schema (WSDL).
- Many commercial applications provide a web service (SOAP) interface.
- Defines arbitrary complex structures and the operations on them.



The technology was adapted at Globalfoundries (AMD) in 2004. Although SOAP/XML concepts were mature at that time, the understanding and technical application was not wide-spread in the industry.

Problems:

- Complexity of an overwhelmingly rich language
- Verbose notation
- Impossible to fully assimilate all details
- No practice in dealing with it
- Poor understanding of concepts and how to do it
- Poor XML modeling skills or experience
 - Artix (IONA) offered a WSDL schema with 1.4 MB source
 - Internal tool applied a naïve approach, working directly on the DOM model (XML tree) instead of adding an abstraction layer.
- **Total dependency on tools**
- Satisfying tools were not available for all applications in the game



Choose your tools well

- Good: Maybe expensive but worth each cent
- Bad: Crapware, purchased at 0 dollar, but paid with time and extra work; often introduced by software cultural factions
- Ugly: Freeware which may work well but hard to use efficiently

SOAP/XML Support in VisualWorks 7.2-7.5

- 👎 Not perfect nor mature
- 👍 Good performance for every day job
- 👍 Tools for schema integration
- 👍 Concept of object-to-XML translation

- 👍 Helped to cope with poor schemas of other tools

VisualWorks 7.2 was the first release to support the technology:

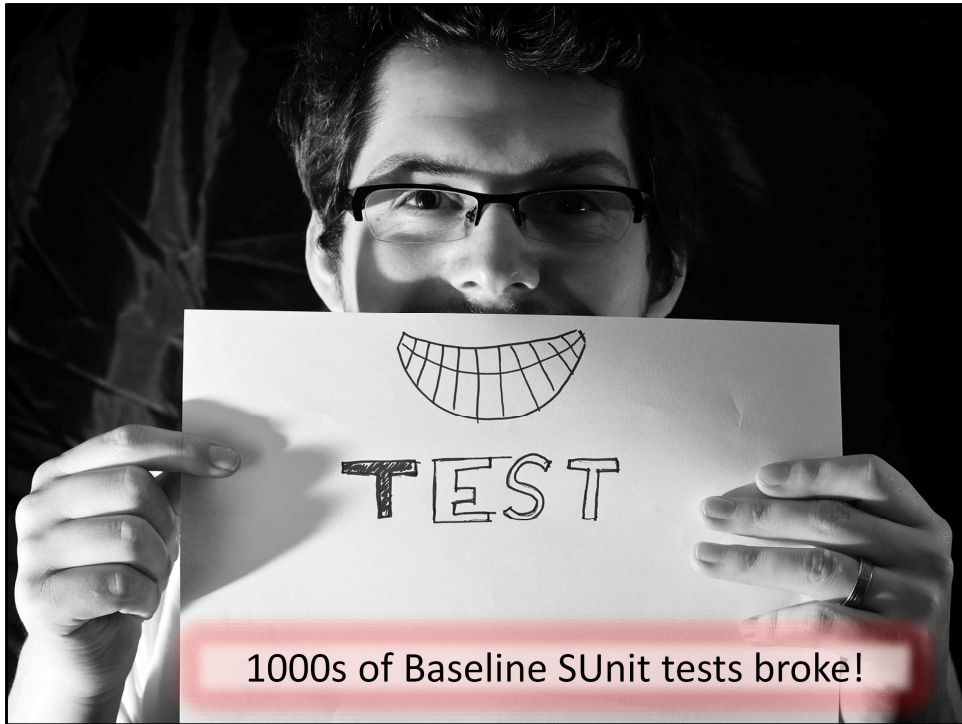
- Far from being perfect or mature
- Lots of bugs, gaps and deficiencies
- But lots of tools that performed well, doing a thorough job

SOAP/XML Support in VisualWorks 7.10

- Mature, W3C compliant
- Improved quality and performance
- Excellent Cincom Engineering Support
 - Thanks, Tamara!

VisualWorks support for Webservices has improved with each release. In VW 7.10 it is

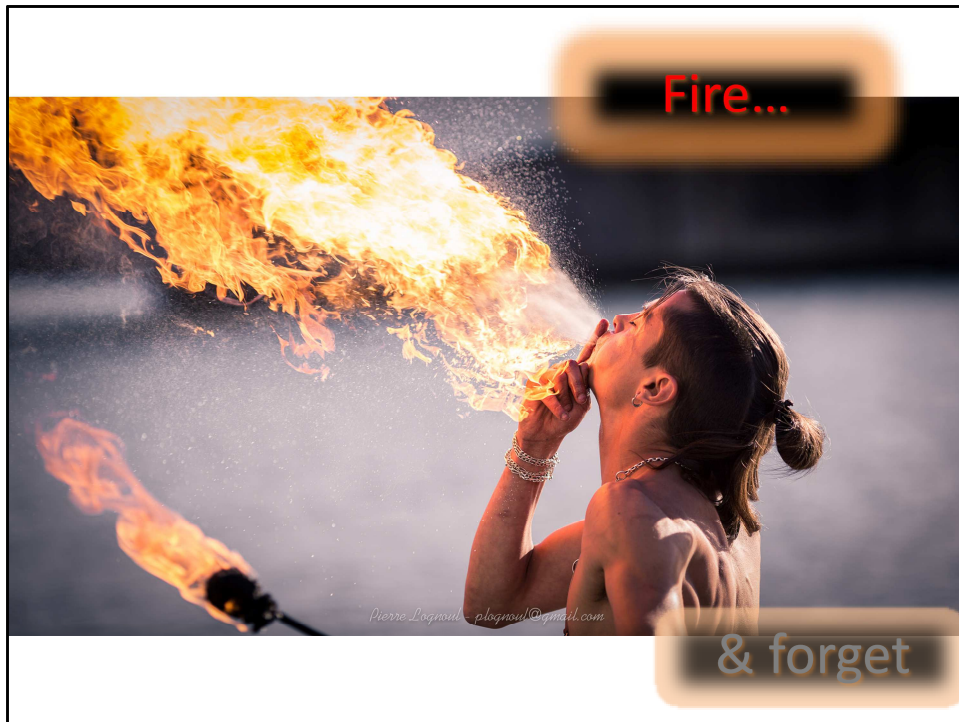
- Mature and reliable
- Feature gaps filled
- Cincom Engineering (Tamara Kogan) helped to overcome problems found during migration



The Globalfoundries can be proud about thousands of SUnit tests, many of them focusing on SOAP processing. We were very confident that these tests would help the migration. We were shocked after the first attempts to run: A significant number of them finished with failure or error.



- **Main “problem”:** Tools improved! VisualWorks no longer accepts „old sins“ (bad format, non-standard use)
- **XML streams were recorded** as input and result for SUnit test – correctness **assumed** but not verified. That was reasonable because it worked.
- Developer **knowledge** about XML did not improve as might necessary.
- **Feedback about XML deficiencies** of other external applications
 - not or only poorly communicated
 - not rendered into an impulse to revise and “paying the technological debts”.
 - Some applications experience the “legacy” fate:
 - No further development
 - No budget
 - Experts (knowledge about those applications) lost.



Fire & forget = Send a message to an external application without feedback

- No result message
- No return code

You cannot know

- When and whether the message has arrived
- Whether it was formally correct or invalid
- Whether it was processed or rejected

⇒ **The hardest scenario to test**

You need high level API at the external application to access and check data about the expected result. Alternatively – and that is the usual case – you need manual inspection.



Conclusions



- Grow your expert knowledge
Find, educate, acquire your experts. Nothing is more serious and dangerous than having no clue.
- Keep the experts! No matter how easy or sophisticated your application is, if the knowledge goes away, the maintenance is harder or even in danger.
- For a **key topic** you **must** have experts!



Standards

- Do your best to understand them (experts)
- Sound understanding allows better modeling.
- Choose alternatives that are easy to understand and communicate.
- Prefer simple over intellectual modeling
- Avoid “artwork” modeling – there is no Nobel prize for application design.

- Standards is the best approach to manage complex interfaces in a complex environment
- No guarantee for painless migration

Pay technological debts!



- Pay technological debts
- Do not allow applications to stay behind.
- No budget for improving a bad application is a poor excuse. Costs and impediments due not fixing bugs in inter-process communication are way worse.



Conclusions:

- Improving Unit and Integration tests is necessary
- Applying standards does not reduce nor replace testing (the XML/SOA safety promise is sometimes treacherous)
- Standards is not the cure-all, because in such a complex environment it cannot be enforced everywhere without compromise



Conclusions:

- Improving Unit and Integration tests is necessary
- Applying standards does not reduce nor replace testing (the XML/SOA safety promise is sometimes treacherous)
- Standards is not the cure-all, because in such a complex environment it cannot be enforced everywhere without compromise

Happy End:
Project successfully finished

- Project finished successfully, Globalfoundries effort was more than expected/planned