



science + computing

| A Bull Group Company



ASAM Transport Format in XML (ATFX) Overview and Best Practices

Dr. Peter Stoll

science + computing ag

IT-Service and Software

Tübingen | München | Berlin | Düsseldorf

Content

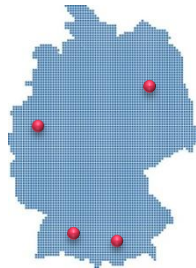
- CAT data management @ s+c
- ASAM ODS ATFX @ s+c
- Use cases
- Capabilities & Features
- Difficulties
- How to use (for developers)

science + computing at a glance

Founded in

1989

Offices



Tuebingen
Munich
Berlin
Duesseldorf

Employees

251

Shareholder

Bull S.A. (100%)

Turnover 09/10

24,82 Mio. €

Core Competences

IT-Services, Software Development, Consulting

Markets

Automotive, Mechanical Engineering, Aero, Microelectronics,
Chemistry, Pharmaceuticals, Biotechnology, Public Sector



Our Customers



science + computing

| A Bull Group Company



CLAAS



NEC



FEV

EvoBus

BEHR



Bremen, Hamburg

Beelen

Duisburg

Branch Office
Duesseldorf

Cologne

Aachen

Mannheim

Stuttgart

Headquarters
Tübingen

Wolfsburg

Branch Office
Berlin

Alzenau

Frankfurt

Ingolstadt

Branch Office
Munich



Deutscher
Bundestag



NUKEM **Continental**



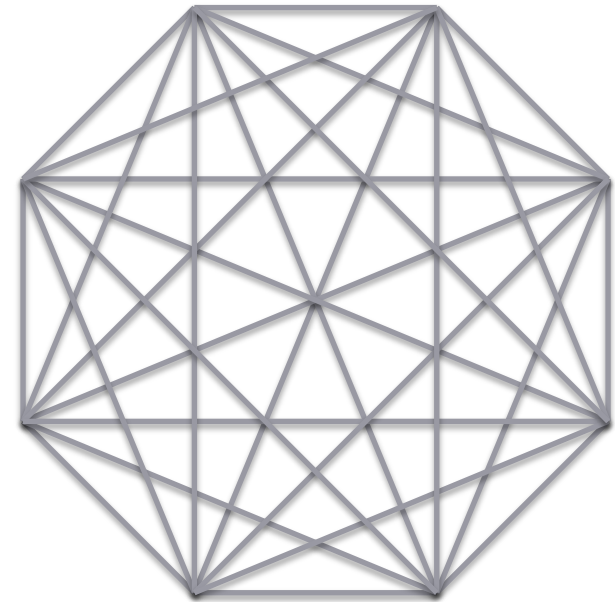
Infineon



Valeo

CAT Consulting, Operations, Software Development since 1997

- Operations of CAT Clients, Servers, Databases, Test- and Measurement Stands
- Consulting in IT-related CAT Topics
- Project Management for CAT Projects
 - Renewing of Test Stands, Software projects
- Automation of Engineering Processes
 - For Test- / Measurement Stands and Analyses
- Software Development
 - Analysis Software
 - Test Stand Programming
- Projects with ASAM ODS
 - Integration, Migration, Development



Measurement Data Management using ASAM ODS

- Numerous Projects since 2000

- Active Member of ASAM



- Active Member of the MDM Community



- Software Development Team for CAT and Data Management

The short version: What is the ASAM Transport Format in XML (ATFX)?

Introduction to the ASAM ODS Standard:

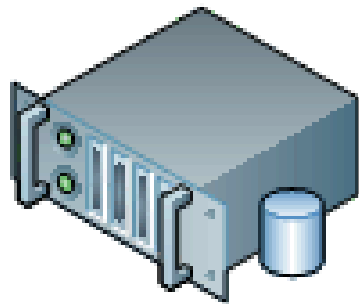


*“The ASAM ODS standard has the fundamental quality of storing data with an architecture independent method. This leads to great advantages when **exchanging data** between different sources and possible prospective customers.”*

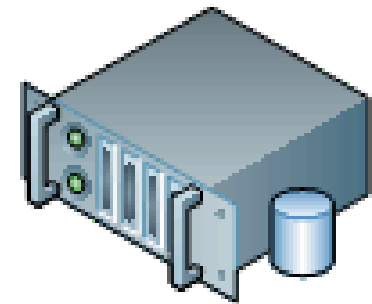
- ATFX **is** the recommended transport format to **exchange data** with ASAM ODS Systems!
- Successor of ATF/CLA

Typical Use-Case 1 (2)

Transport between ODS databases



ASAM-ODS Server
Vendor XYZ

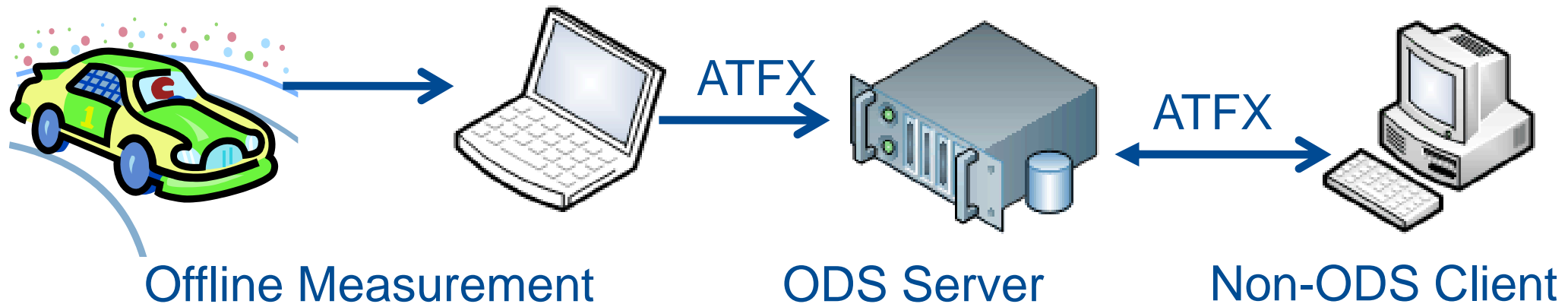


ASAM-ODS Server
Vendor ABC

- “A standardized, easy to use, text-based exchange format in order to exchange ASAM ODS data ... between different systems and different platforms.”
- Transport data from one ODS database to another

Typical Use-Case 2 (2)

Import/Export to ODS databases



- Transfer of measurement data from decentralised, off-line measurement equipment to a central ODS server
- Export/Import data from/to ODS database in case a system
 - is not capable to interface to the server through ODS API (CORBA)
 - is currently not connected to the server through a network
- **ATFX provides an interface of non-ODS-Systems to ODS**

Capabilities of ATFX

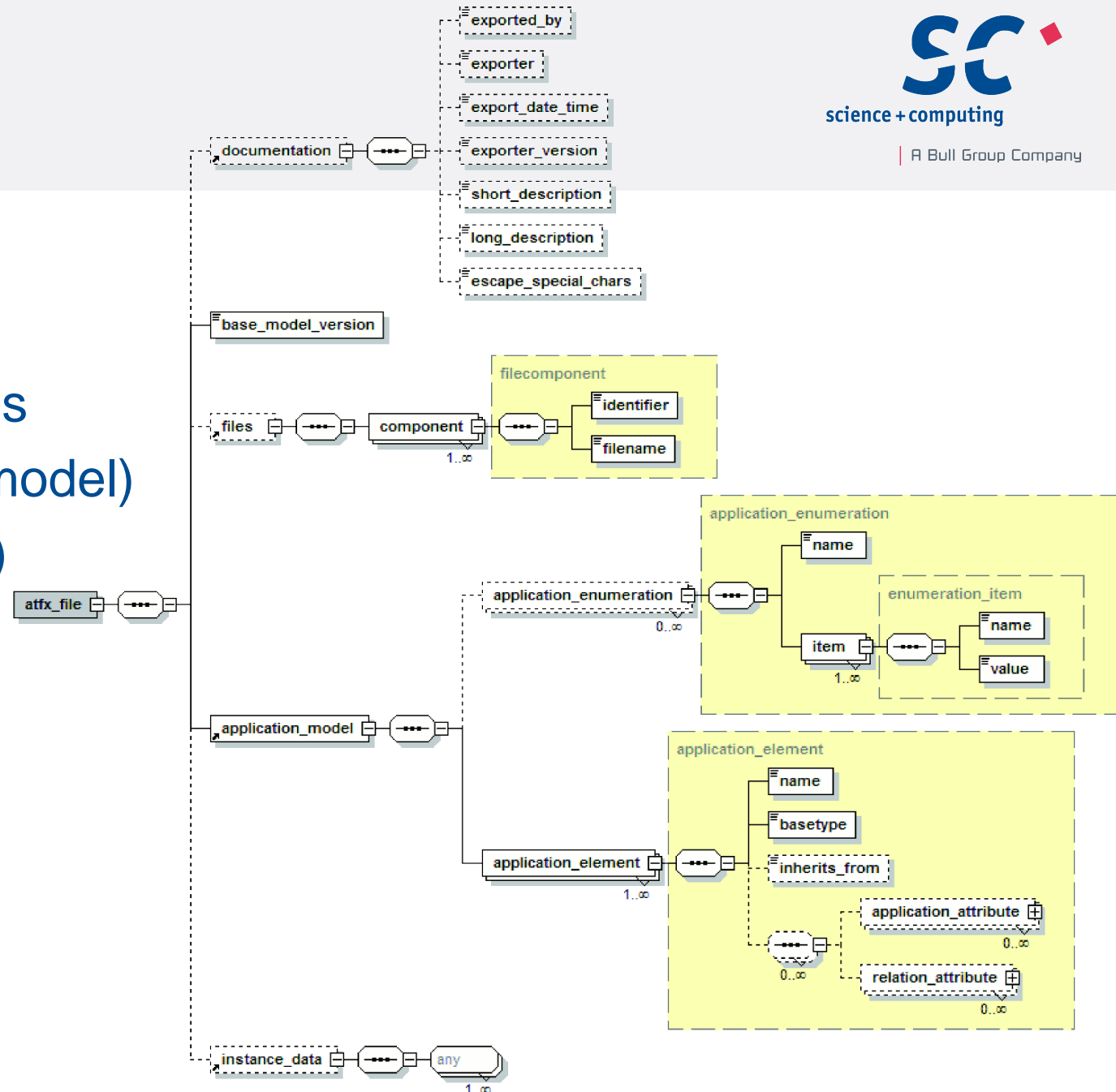
- Portable, standardized (ASAM), complete
 - Transfer off all ODS-data between all available platforms and operating systems
- Hybrid XML/binary data format
 - Most information in XML
 - Mass data optionally in separate binary files in a standardized way
- Each file includes the data model (some semantic of the data)
- Syntax and semantic partially defined by XML-Schema (XSD)
- Can “easily”(?) be created and read by any application and is human readable
- No need to know about CORBA or RPC (the ODS-internal protocol)

May be the first and easy step for a products ASAM ODS compatibility

ATFX Content

Important content

- Documentation
- External (binary) files
- Model (application model)
- Data (instance data)



External File Components

Binary Data Files

ATFX main file (XML)

```
<atfx_file ...  
...  
<component>  
  <identifier>myfile</identifier>  
  <datatype>ieeefloat8</datatype>  
  <length>11</length>  
  <description>my data</description>  
  <inioffset>0</inioffset>  
  <blocksize>8</blocksize>  
  <valperblock>1</valperblock>  
  <valoffsets>0</valoffsets>  
</component>  
...
```

references

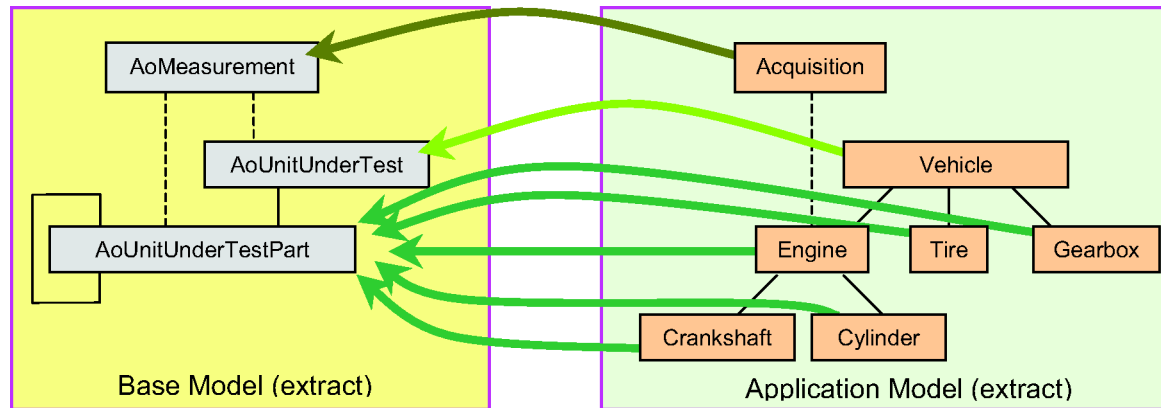


External binary data file

```
2A1B 929A EABB 7A9C 23AC 9495  
34AE 5AA2 8F2F CEE7 DA19 D415  
1D20 CB4E C6F3 EA74 5CC1 ACFA  
6061 795D 3497 3F3D 3AEC 8B6E  
8C61 1E30 0EDC 478C AB64 2057  
...
```

- ATFX can refer to large external datasets of measurement data
- ATFX describes exactly how the binary data is stored
- ATFX (ATF/XML) is a hybrid data format!
- **Well suited for large data sets**

Semantics to the data: The Application Model



- Each ATFX-File contains the application model for this data
- The application model defines a data model, comparable to the definition of database tables in SQL or Class Diagram in UML.
- Data types are well defined (int, float, ... arrays, ~ 30 types) for all data
- Relations are defined between data
- A “Base Model” exists, which shall be extended

All data in ATFX-Files have well defined syntax, defined data types or relations and thus defined semantics!

Semantics: The Base Model

Base model predefined in ODS

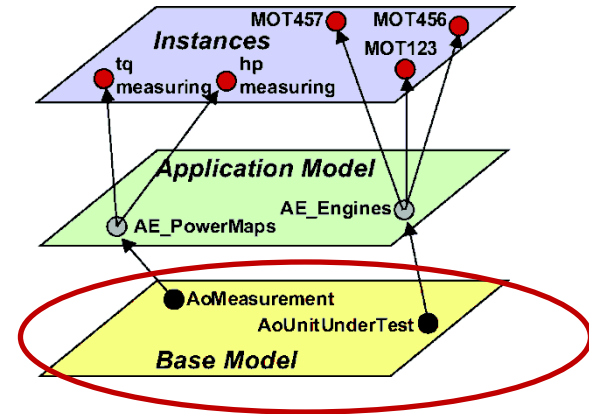
Only the application extensions must be included in the ATFX File.

Advantage

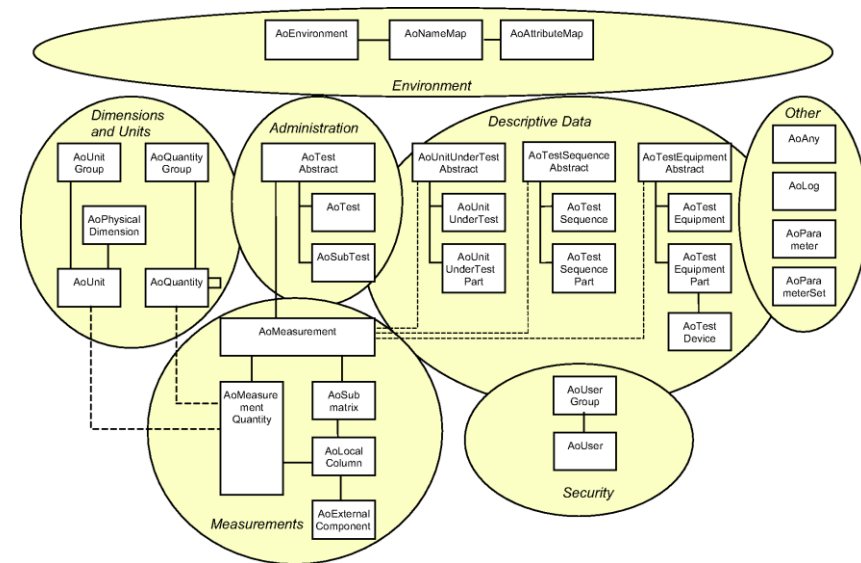
- Some common semantics across all ATFX-Files

Drawback

- One needs to know the base model in order to understand the data (e.g. is it a list of 2 floats or a complex number?)
- The base model is complex, approx. 30 Tables/Classes with 10-20 entries each (attributes or relations)



Not in ATFX File



Base Model, Overview

Semantics: The XML-Schema (XSD)

- An XML-Schema exists for ATFX-(XML)-Files
- XML-Schema defines additional semantics to the data
- But: XML-Schema is incomplete for application data (instance-data)!

What you might expect:

```
<data type="vehicle">  
  <val attr="speed">123.456</val>  
  <rel to="wheels">1 2 3 4</rel>  
  ...
```

What it is in ATFX:

```
<Vehicle>  
  <speed>123.456</speed>  
  <wheels>1 2 3 4</wheels>  
  ...
```

- **Each application defines a new set of XML-Tags!**
- Idea: Create your own XML-Schema for just your application
- Practice: General purpose (inter-application) reader/writers are more difficult to develop, since the complete XML-Schema is not available at compile time, e.g. code generation from XSD-Schema (as JAXB) not possible

A few difficulties

- Incomplete XML-Schema: Instance only partially defined in general
- Namespaces changed from version to version, though namespaces are usually ignored
- Escaping rules for instance data tags required, since information is coded in XML-tags: `<body force>` results in `<body_20force>`
- No extension elements are defined (as `<xsd:any>`), violations of standard are common if non-standard information is added
- Unnecessary complexity through duplicated, but similar XML elements. Approx. 90 predefined XML-tags need to be understood to read a ATFX file
- Knowledge of the complex base model is required to get data types

ATFX does the job, but to create feature-complete writers and parsers it is not "easy"

- **Ascoba (application), based on ATHOS (-library):**
Part of Avalon ASAM-ODS Server (<http://highqsoft.de>), responsible for import/export of ATFX in Avalon.
- **openATFX:** <http://openatfx.sourceforge.net>
Under active development, written in Java provides a standard ODS-Interface (Corba) to ATFX
- **Write your own** writer or parser based on any XML-library.
Feasible way if limited set of ATFX-files needs to be dealt with.
Creating a 'writer' (output) is much easier than a 'parser'
- **atfxlib** (myself, science+computing ag), almost complete.
Provides a simple standard Java-API to ATFX without dependencies to ASAM ODS Software

Best practices when dealing with ATFX

Takes into account that most ATFX-files violate the ATFX-standard in some way. Based on real-world examples:

- Use a modern XML library (support of `<xs:include>` needed)
- Ignore namespaces. If not possible, first remove namespaces
 - Even examples from the standard violate the namespace rules (namespace of HelperSchema.xsd is never respected). ODS 5.1 and 5.2 documents do not differ
- Allow extra elements within `<documentation>` and ignore order
 - Some put extra information there
- Ignore case for (some) enums
 - most documents use wrong case at `instance_data/.../component/datatype`
- Escaping: ignore case
 - e.g. accept `"_5f"` and `"_5F"`

Summary

- ATFX is a good way to interface ASAM ODS systems for non-ODS aware software
- The ATFX standard is complete, stable and mature
- The application model and the XML-Schema give semantics to the data
- The ATFX Format has some edges which make it not easy to create parsers, while creating writers is less a problem



Thanks for your interest in ATFX.

Dr. Peter Stoll

science + computing ag
www.science-computing.de

Phone ++49-(0)7071 9457-0
info@science-computing.de