Experiences with MDM development based on ASAM ODS

Introduction

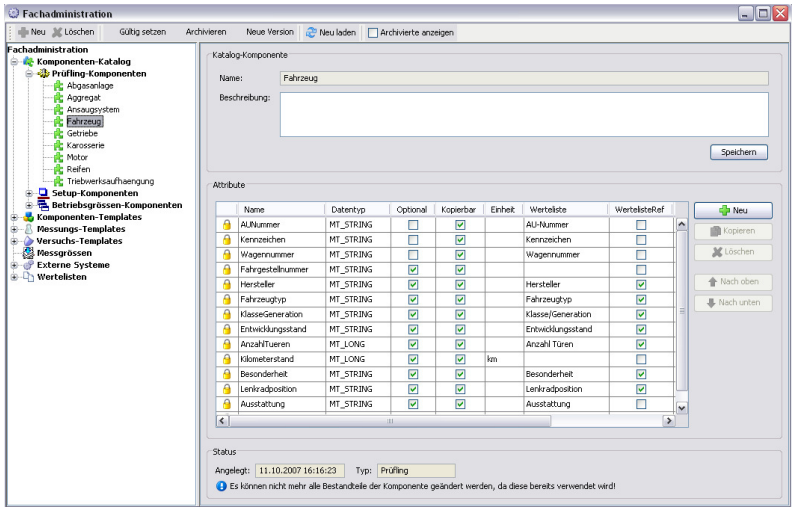
- Name: Christian Rechner
- Junior Software Developer
- EPOS CAT GmbH
- 1 year experience in programming client software with ASAM ODS (mainly JAVA)

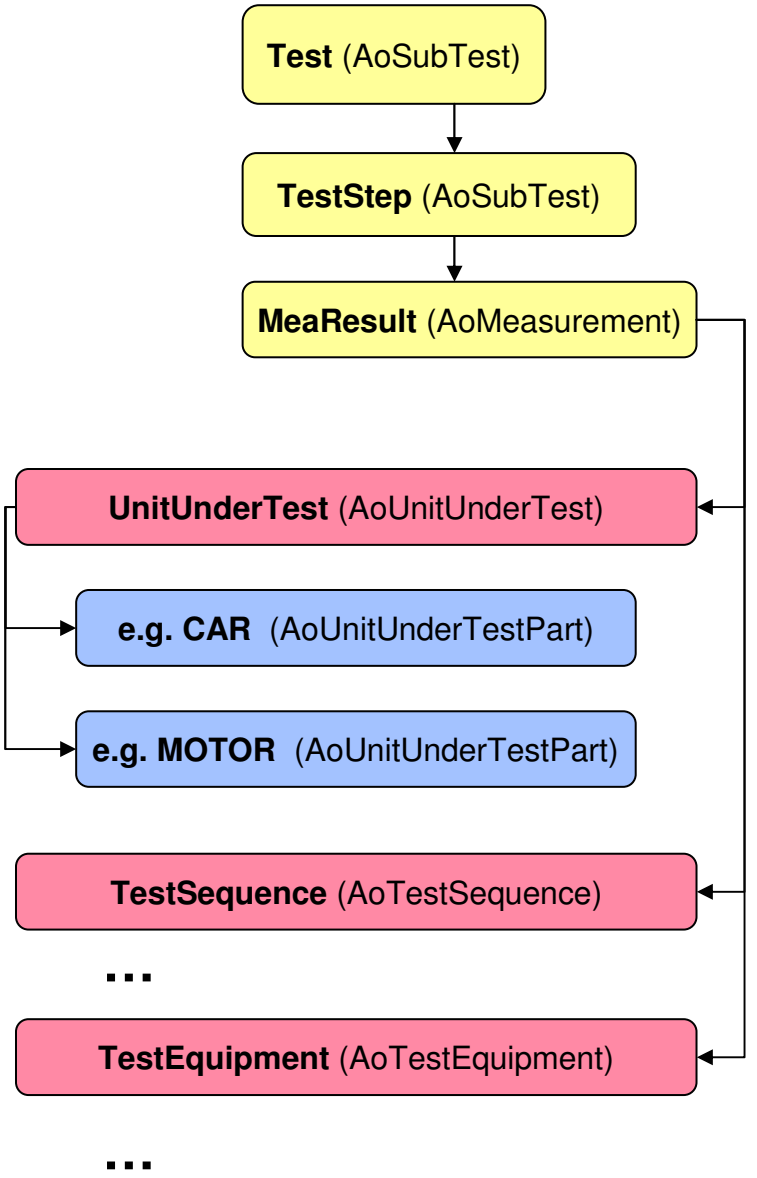
Summary

- Architectural overview
- Experiences in the software development process with ASAM ODS
  - Database design
  - Data access layer
  - User interface
- Usage of Tools
- Benefits of MDM compared to proprietary ODS models
- Conclusion

The MDM idea

- Standardized and fixed application model
- Changes of the model are allowed only in adding/removing/altering application elements and attributes of descriptive data
- Changes can be made at runtime with a graphical user interface
- Additional metadata is stored in the database (like value-lists or templates)





Architecural overview

Special data viewer

Special report generation

Import / export

Project specific

Navigator | Search | Administration | Test planning | Generic data viewer

MDM-Client-Framework

MDM-API (Client- and Server-Part)

MDM platform

ODS-Server (Athos)

Oracle-DB

File-Storage (measurement mass data)

Document-File-Storage (docu)

Standard ASAM-ODS

Database design

- Different approaches
  - Classic ER-design -> fit to ODS afterwards
  - Manual ATFX editing -> generate database automatically
  - Graphic design of model with UML -> generate ATFX -> generate database
- Validation is crucial: use tools (like ODS-Checker or Athos)
- Think about developing a (semi-)generic model (like MDM)
- **NEVER violate the standard!**

Implementing the data acess layer

- CORBA communication layer fits well
- Performance is very crucial
- Never use direct database access
- Do not fall into the trap of using the ODS object model for sophisticated data access
- Make extensive use of extended queries
- Programming with CORBA structures can be painful => write tools to wrap them into objects of the programming language

=> ODS is very fast, just use it right!

Performance comparison

Object-oriented

```
ApplicationElement aeTest = mdmCache.getApplicationElement("Test");
InstanceElementIterator iterTest = aeTest.getInstances("*8K0*");
for (int i = 0; i < iterTest.getCount(); i++) {
   InstanceElement ieTest = iterTest.nextOne();
   InstanceElementIterator iterTestStep = ieTest.getRelatedInstancesByRelationship(Relationship.CHILD, "*");
   for (int a = 0; a < iterTestStep.getCount(); a++) {
      InstanceElement ieTestStep = iterTestStep.nextOne();
      System.out.println(ieTestStep.getName());
   }
}
```

=> 10.000 ms

Query based

```
// build query
ExtQuery query = mdmCache.createExtQuery();
query.addSelect("TestStep", "Name");
query.addJoin("Test", "TestStep", "TestSteps");
query.addStringCond("Test", "Name", SelOpcode.LIKE, "*8K0*");

// execute query
ExtQueryResult res = query.execute();

// print result
while (res.next()) {
   String tsName = res.getStringVal("TestStep", "Name");
   System.out.println(tsName);
}
```

=> 200 ms

Implementing the data acess layer

- Even for Update- and Inserting data access it can be useful NOT to use to ODS object-model
- Install performance critic server process on the same machine as the ODS server to prevent network traffic
- Caching mechanisms
- For large file-transfers of mass data use socket communication instead of CORBA
- Think about developing an extra layer for a known model (like the MDM API)
- Release management is vital

Implementing the data acess layer

- Optimize storing the channels in reordering them
- Deletion of elements can take a long time

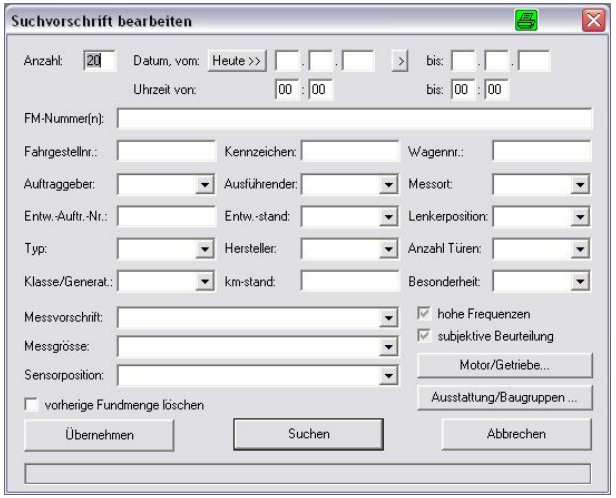
Example of MDM-API usage

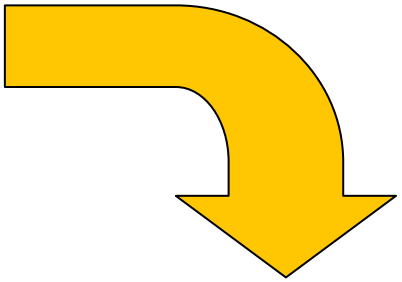
```
env.startTransaction();

Project project = env.getProjectByName("TestProjekt");
if (project == null)
     project = env.createProject("TestProjekt");

StructureLevel sl = project.getStructureLevelByName("TestLevel");
if (sl == null)
     sl = project.createStructureLevel("TestLevel");

// create Test out of TplTest
TplTest tplTest = env.getTplTestByNewestValid("K6 Simulation");
Test test = sl.createTestByTpl("TestVersuch", "In Bearbeitung", tplTest, "Gemessen");

// create MeaResult (AoMeasurement)
TestStep ts = test.getTestStepByName("K6 Simulation");
MeaResult result = ts.createMeaResult("Messergebnis");
result.setMeasuringPerson("Max Mustermann");
SubMatrix subMatrix = result.createSubMatrix("SubMatrix");
// create Channel 1 (AoMeasurementQuantity + LocalColumn)
Quantity quaChan1 = env.getQuantityByNewestValid("Kanal 1");
MeaQuantity meaChan1 = result.createMeaQuantity(quaChan1);
meaChan1.addFloatValues(subMatrix, new float[] { 4, 7, 1, 3 });
// Time (AoMeasurementQuantity + LocalColumn)
Quantity quaTime = env.getQuantityByNewestValid("Time");
MeaQuantity meaTime = result.createMeaQuantity(quaTime);
meaTime.addFloatLinear(subMatrix, 0, 5);
meaTime.setIndependent(true);

env.commitTransaction();
```

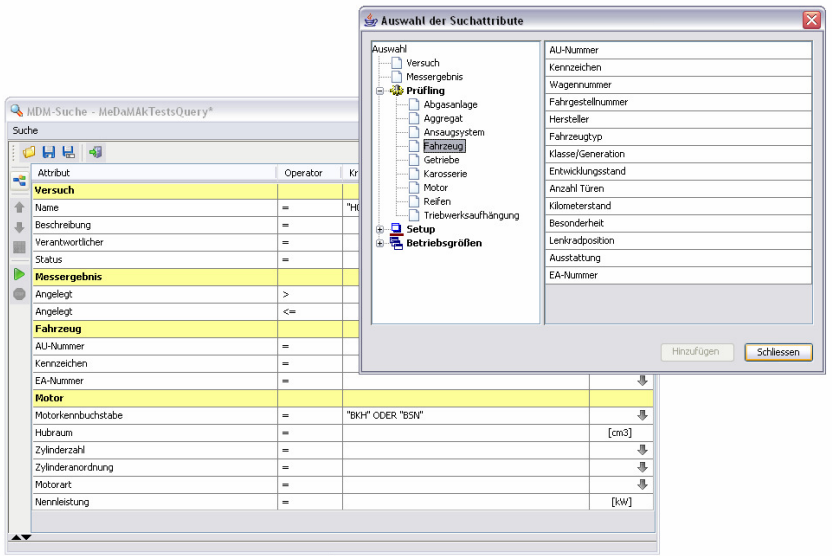
User interface

- Develop independent components
- Sophisticated generic user interfaces working on all ODS models are often not possible
- Find the bridge between reusability and user friendlyness
- Experience has proven that most projects have similar requirements
- For special needs use plugin concept

Example of a generic search mask







Tools

- Make extensive use of tools
- Tools are available for all steps in the development process:
  - ASAM ODS checker
  - UML2ATFX
  - ATFX2Database
  - ATFX exporter
  - ASAM ODS database browser
- If no tools for your problem are available, write your own and contribute them to the MDM community

Benefits of MDM

- Provides the bridge between usability and standardization
- Comes with a set of often required components, even user interfaces
- With the MDM-API, real object oriented and performant data access is possible
- Provides easy access to a very complicated issue
- 100% ASAM-ODS standard
- Proven in a couple of productive systems
- The development steps until client-programming can be omitted

Conclusion

- ASAM ODS is preferable
- Desirable tools:
  - Standardized Open-Source and Client APIs for specific programming languages (e.g. Java) that cover query-wrappers and caching mechanisms
  - Open-Source ASAM-ODS-Server (usable with freeware databases like MySQL or PostgreSQL)
- ASAM ODS has very good documentation


CAT SERVICE
Codierung laden
Anzeige
Unbenannt - Termin
Datei Bearbeiten Ansicht Einfügen
Speichern und schließen
Termin
Checkliste
erledigt
Fahrzeugdaten
Fahrzeugbezeichnung
Radstand
Stirnfläche
Reifen vorne
Reifen hinten
Portal Test
Fahrzeug: PKW
Testart: Frontal
Signalbeschreibung
Objekt: Fahrzeug
Sitzposition: vorne links
Meßstelle: Kopf
Dummy Typ: Dummy
Dimension: Beschleunigung
Filterklasse: 1000
Ergebnisse:
Maximale Beschleunigung
Werte
[GRAPH: CRASH1.LPO]

Thank you