IMMW-15

Architecture and Features of an Extensible Measurement System Framework

Jerzy M. Nogiec

Batavia, August 2007

Vision

- Replace the existing magnetic measurement systems by new systems, which are developed based on the same technology and allow for high level of reuse and accommodation of various hardware and data processing solutions.
- Allow for easy extensions and modifications needed in R&D.
- Allow for flexibility in data management.
- Base the solution on open systems and standards.

Extensible Measurement System (EMS)

EMS is a component-based framework for building measurement, data acquisition and dataflow processing systems. The goal of the EMS project was to design, implement, and deploy a system that is extensible, flexible and dynamic. EMS can be used to develop configurable and dynamically reconfigurable systems. It is perfectly suited for building a family of systems.

Framework – the skeleton of an application that can be customized and reused.

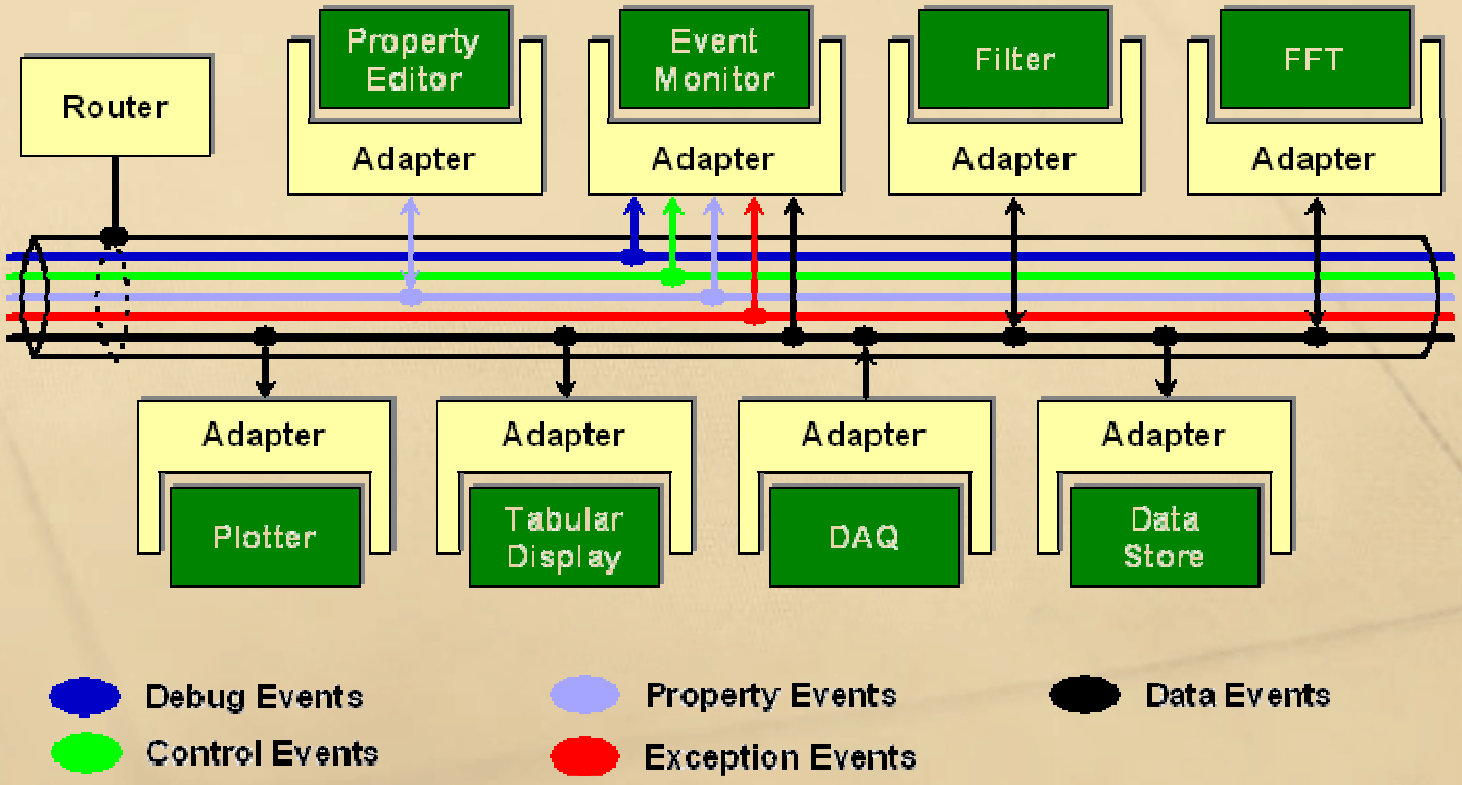
Component – an independently released software module suitable for composition (together with other components) into multiple applications.

Family - a group of systems built based on common software assets, such as frameworks, components, or libraries.

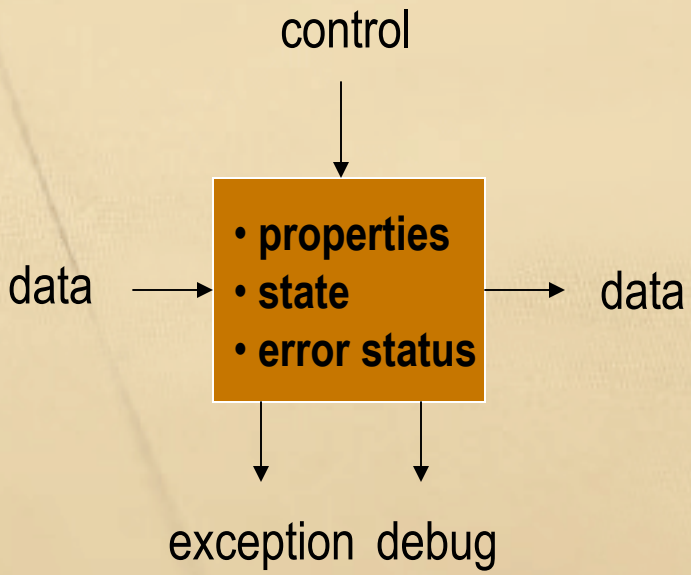
EMS Architecture

Batavia, August 2007

Architecture

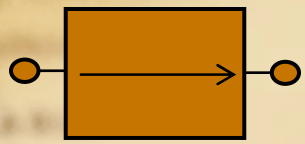


Component

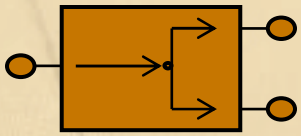


Components have properties and state. Typical components input, process, and output data. Their behavior depends on their state and property values. Components can be directed to perform certain actions by sending control events to them. Components may also output debug and exception information.

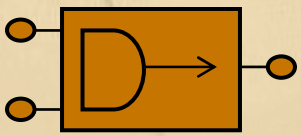
Dataflow



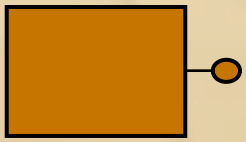
filter



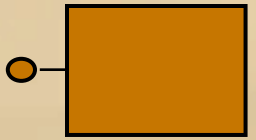
splitter



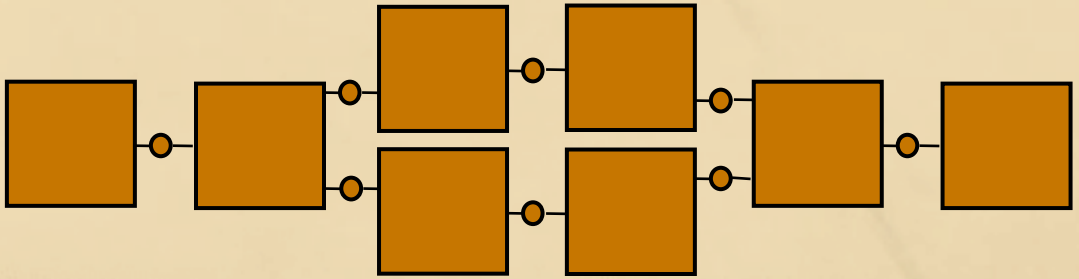
merger



source



sink

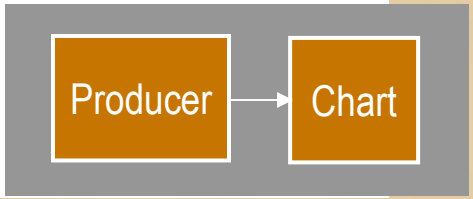


There are several categories of data processing components:

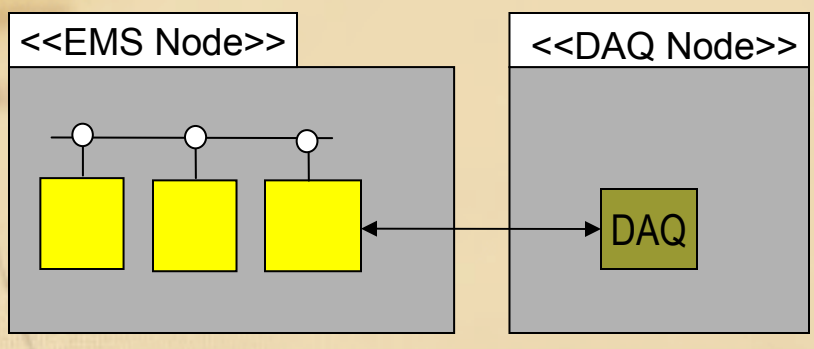
- Data sinks
- Data sources
- Data filters/processors
- Data splitters
- Data mergers

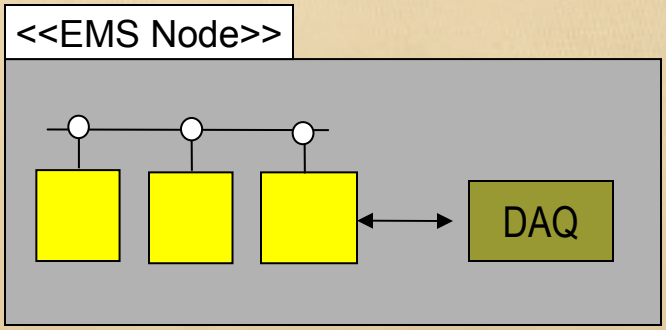
XML Dialect

```
<configuration version="0.1" title="Display Test XML">
    <!-- Component definitions -->
    <component id="Producer"  class="ems.core.components.SimpleDataGenerator">
          <property name="title" value="Data Generator Component"/>
    </component>
    <component id="Chart"  class="ems.measurement.chart.ChartDataDisplay">
          <property name="XPosition" value="0"/>
          <property name="YPosition" value="200"/>
          <property name="title" value="Plot Display"/>
    </component>

    <!-- Routing information -->
    <route type="Data"      origin="Producer" destination="Chart"  />

    <!-- Control signals -->
    <control signal="init"  destination="!"/>
    <control signal="start" destination="!"/>

    <groups>
      <group name="Display" position="778,423" size="548,219">
         <node name="Producer"/>
         <node name="Chart"/>
      </group>
    </groups>
</configuration>
```



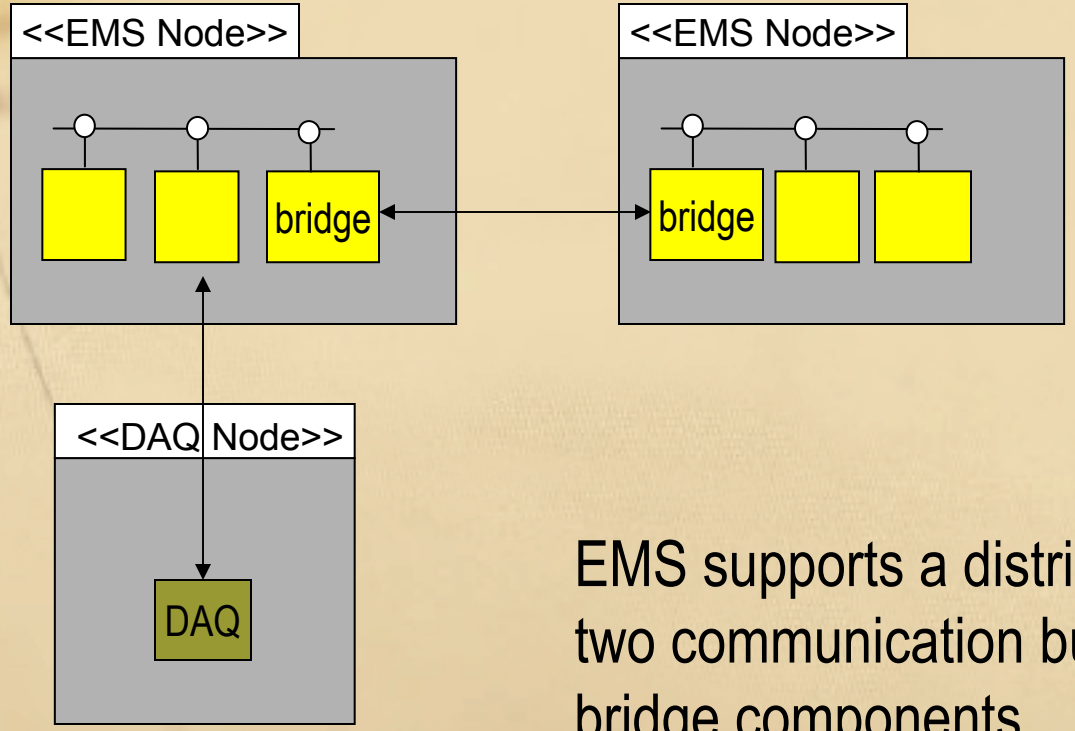
DAQ Organizations





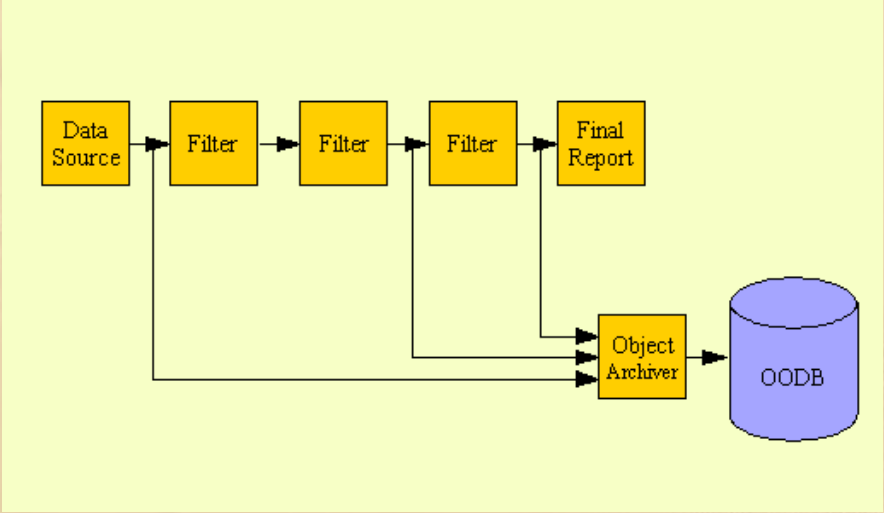
EMS supports both distributed and local DAQ architectures.

Distributed EMS



EMS supports a distributed solution where two communication buses are connected via bridge components.

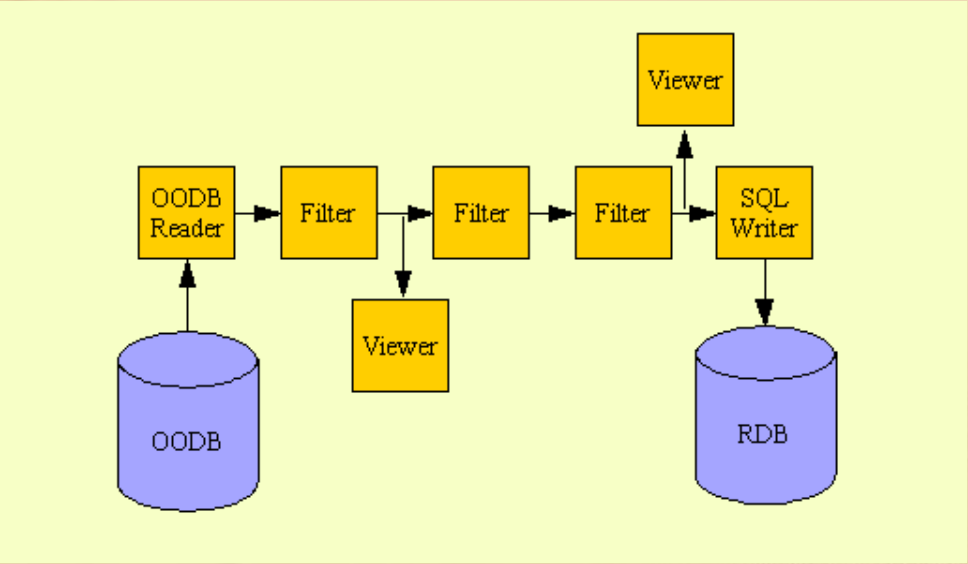
Streaming to OODB



Data from any point in the data flow can be archived in OODB with no need to do any special preparations of the database (no alterations to DB schema, no programming of DB access, etc.)

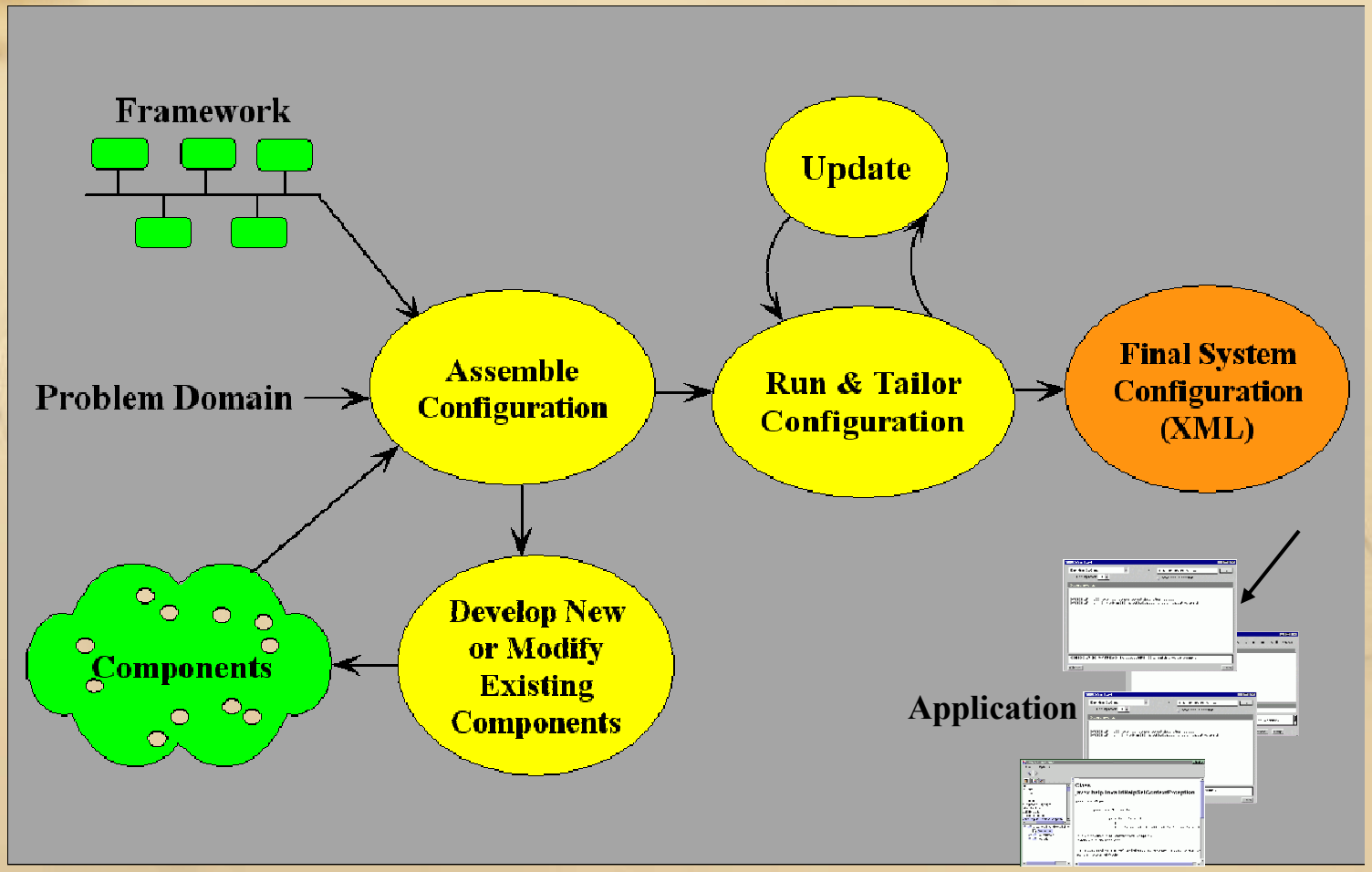
OODB offers fast persistence of data streams.

Data Publishing



- Data (final results) can be loaded either on-line or off-line to a relational database.
- A component can be set up to apply the same SQL statement to a stream of data.

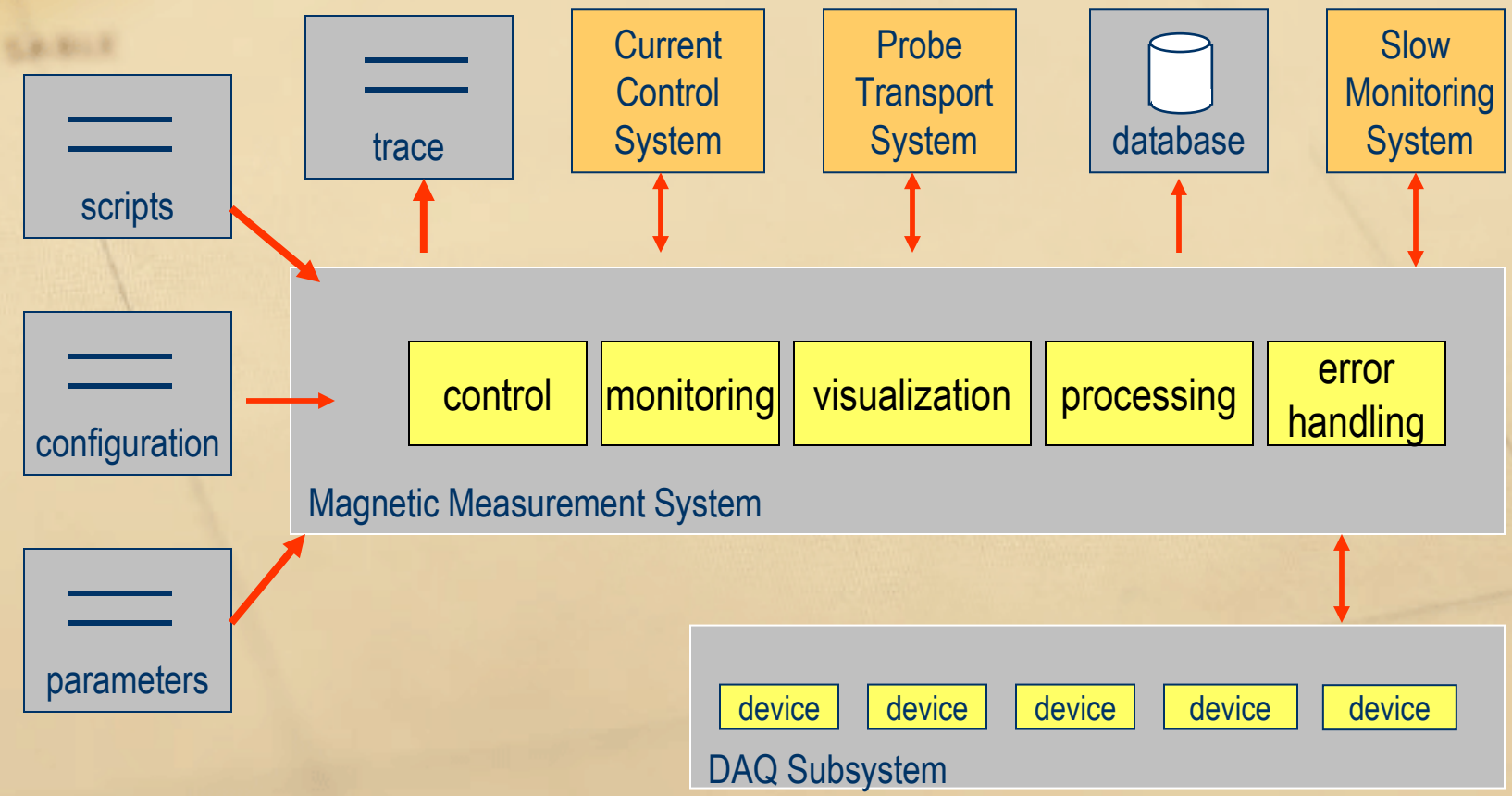
Application Lifecycle



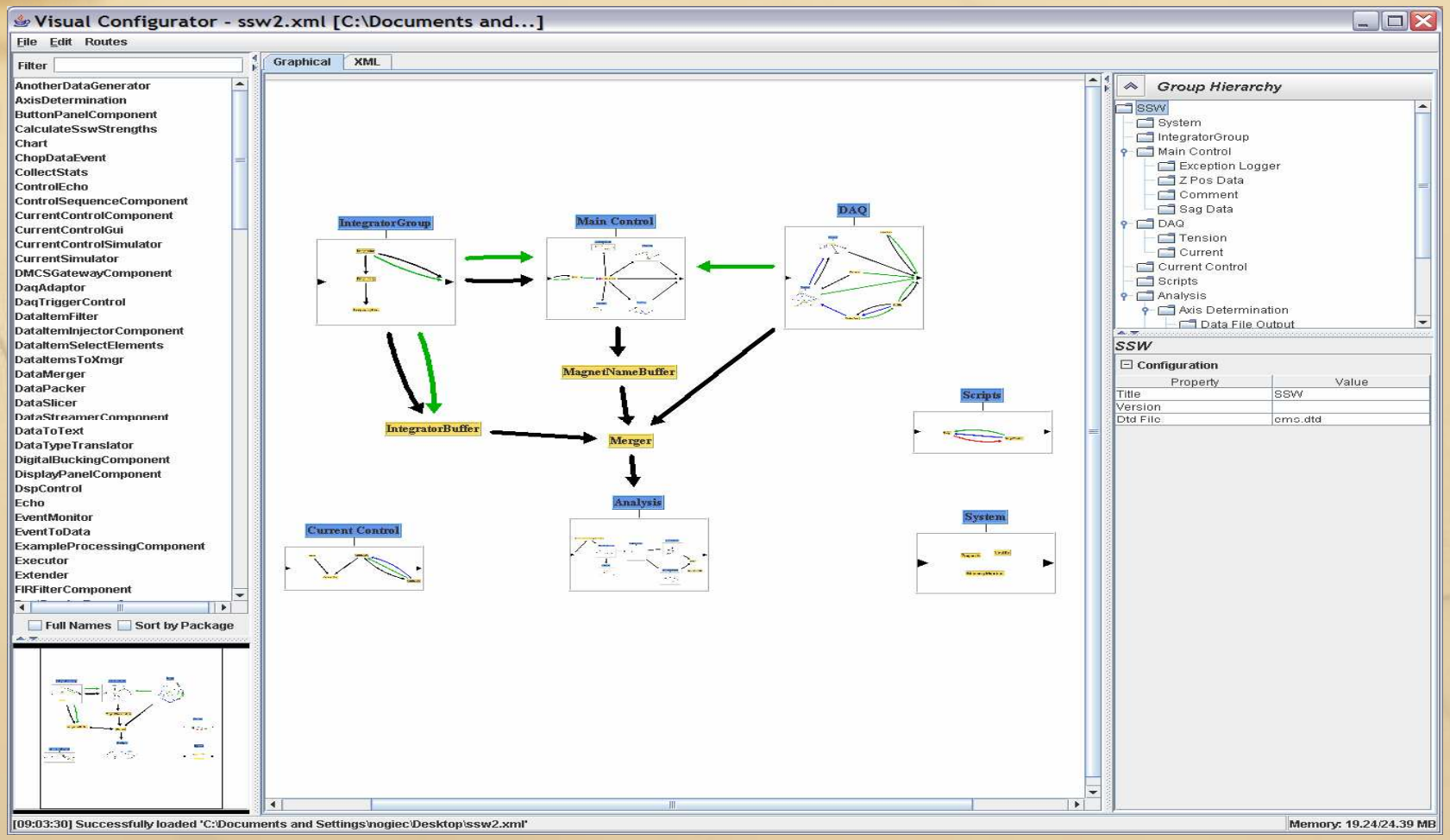
Configuring Applications

Batavia, August 2007

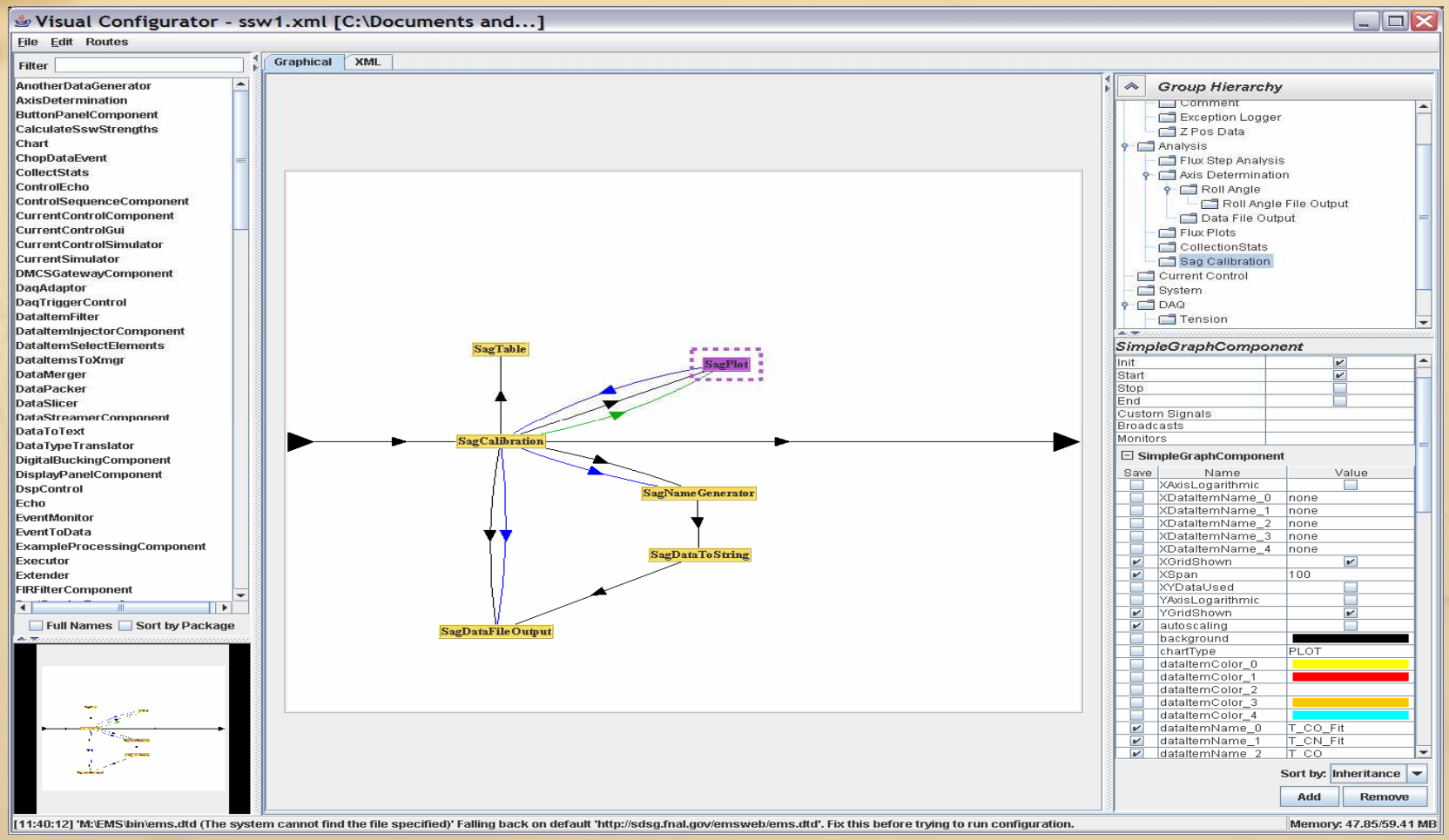
Magnetic Measurement System



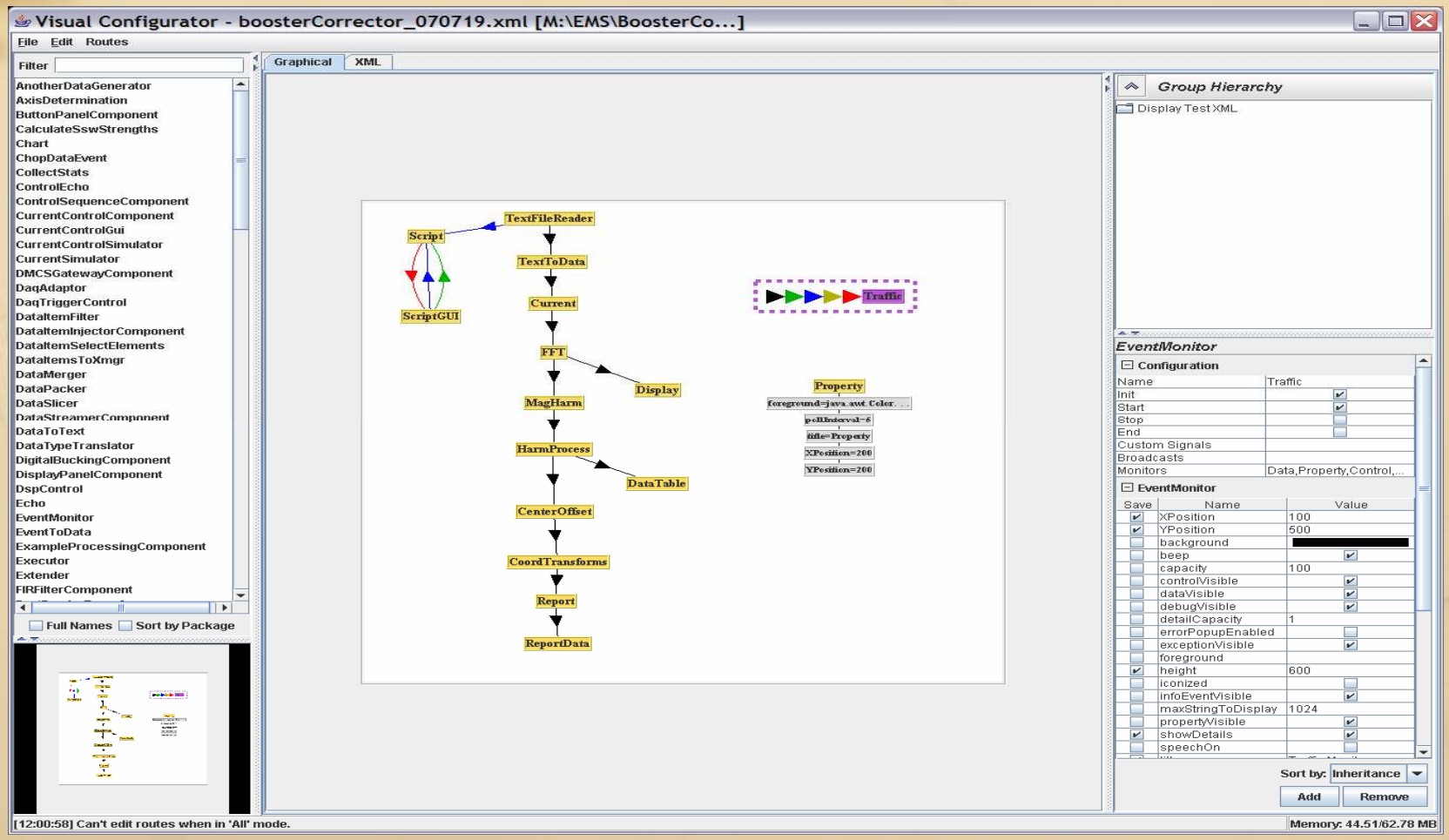
Configuration with Groups



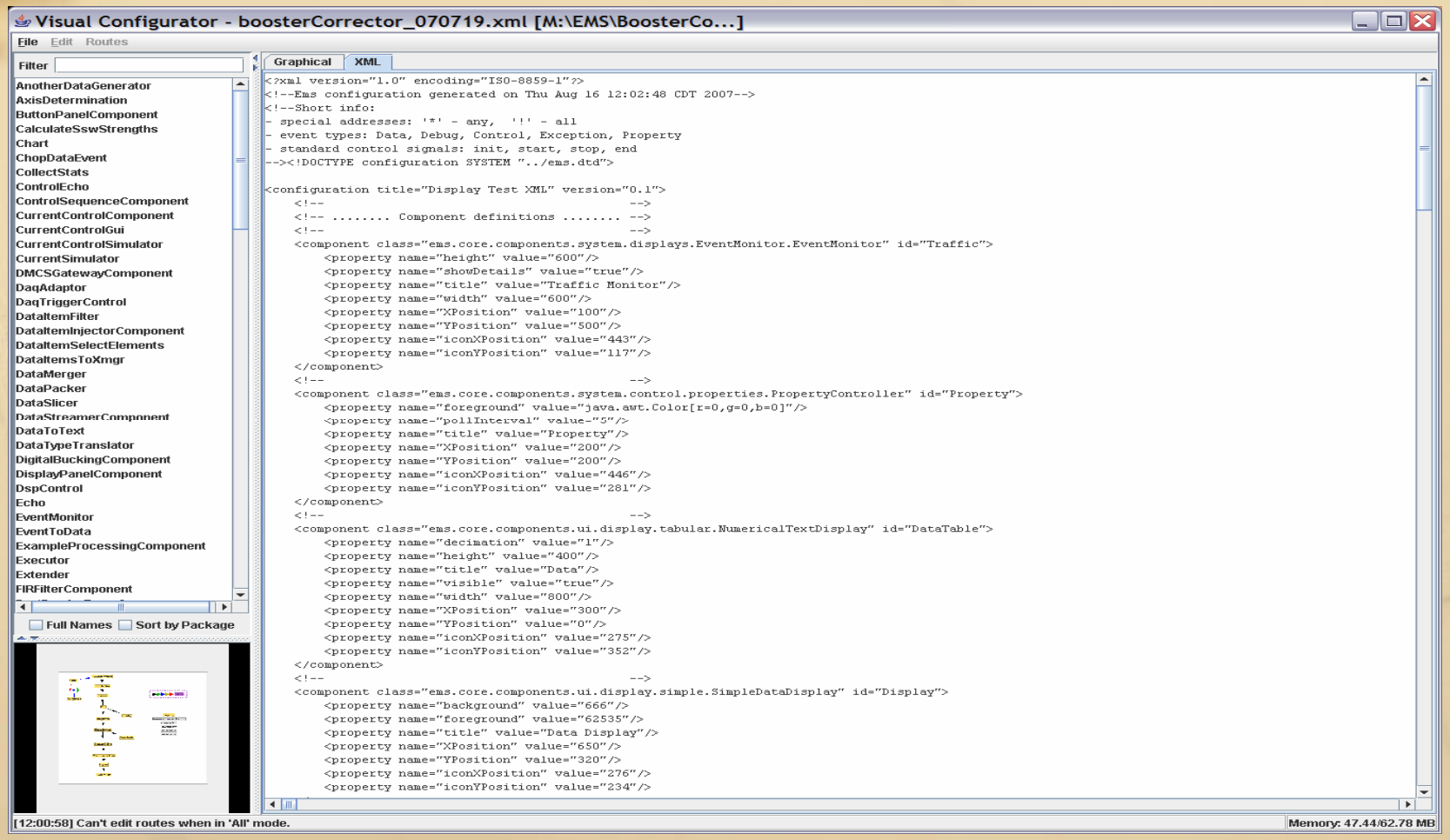
Configuration Group View



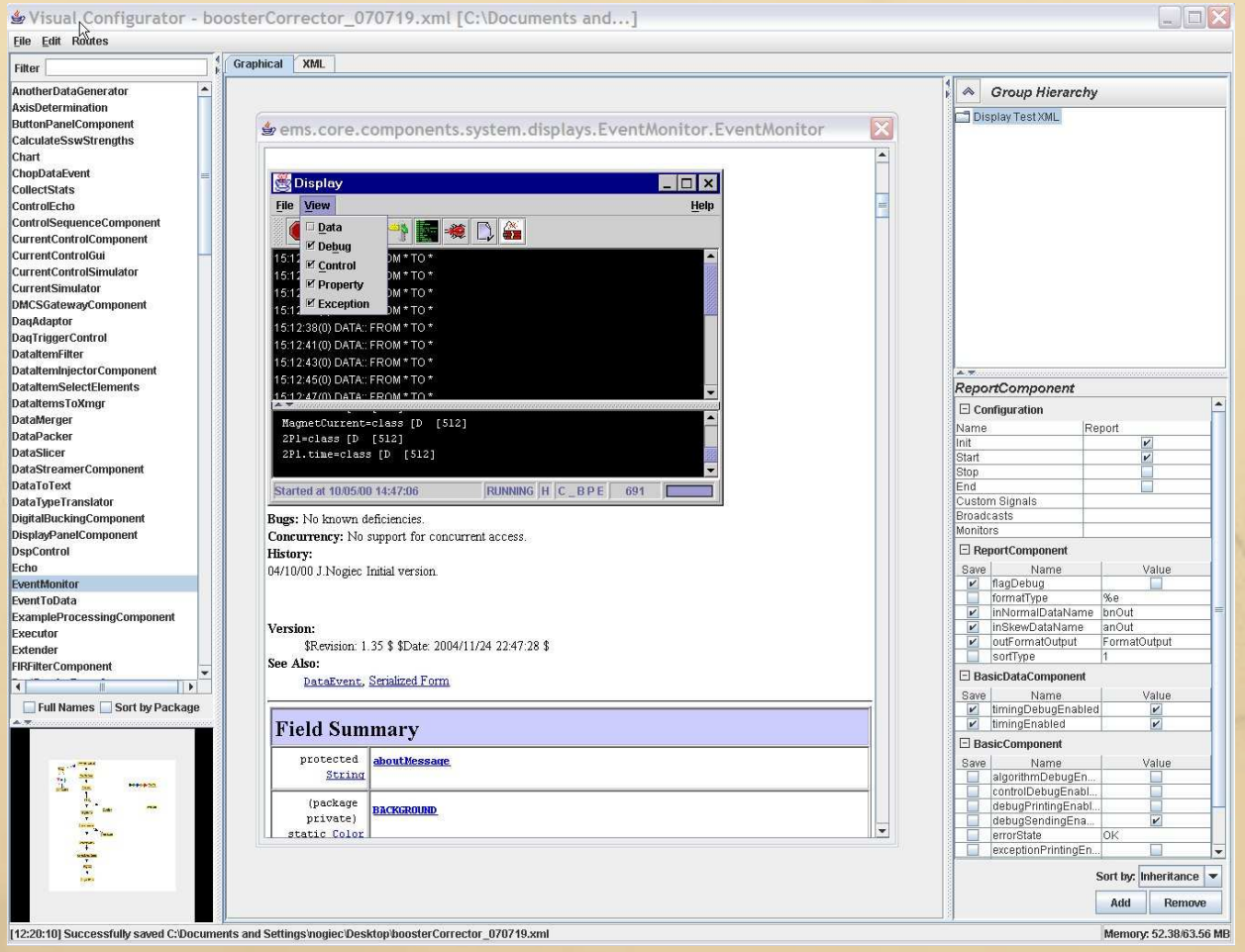
Configuration with No Groups



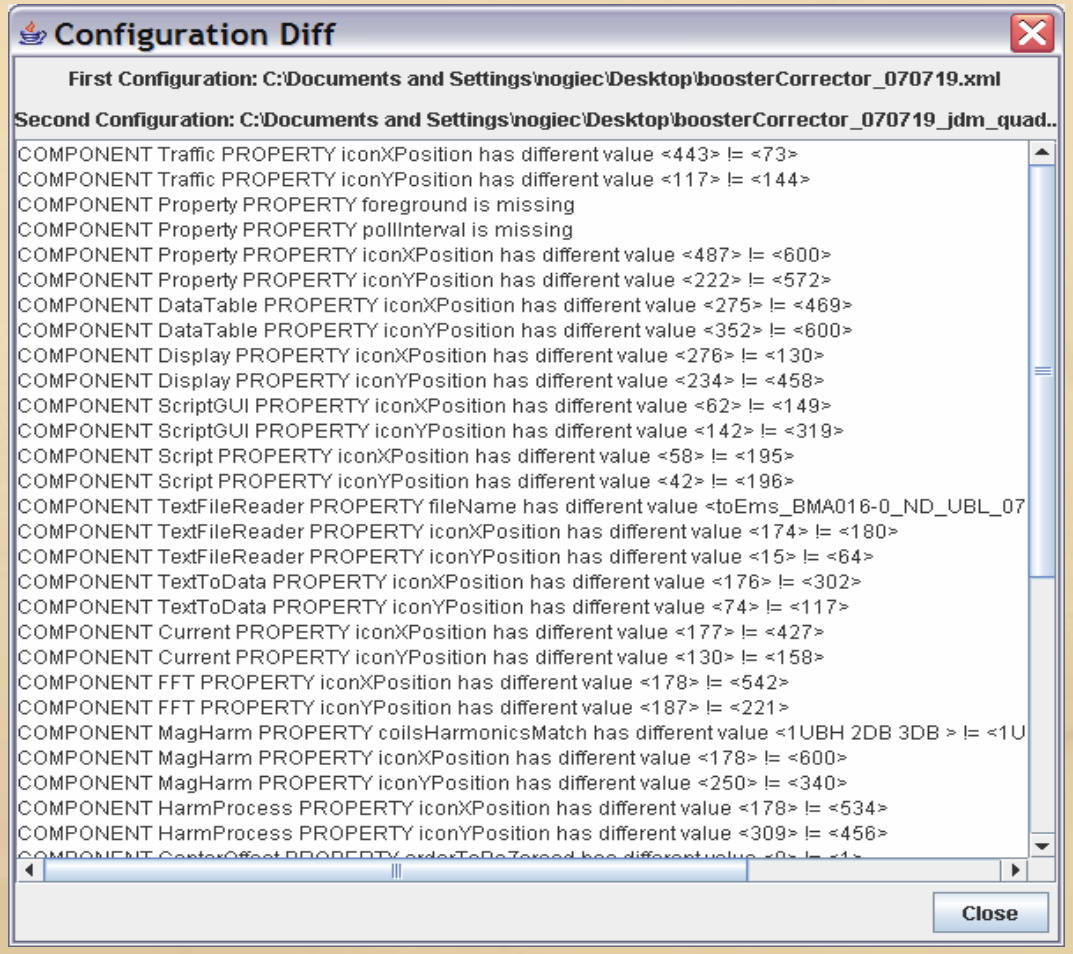
XML Configuration View



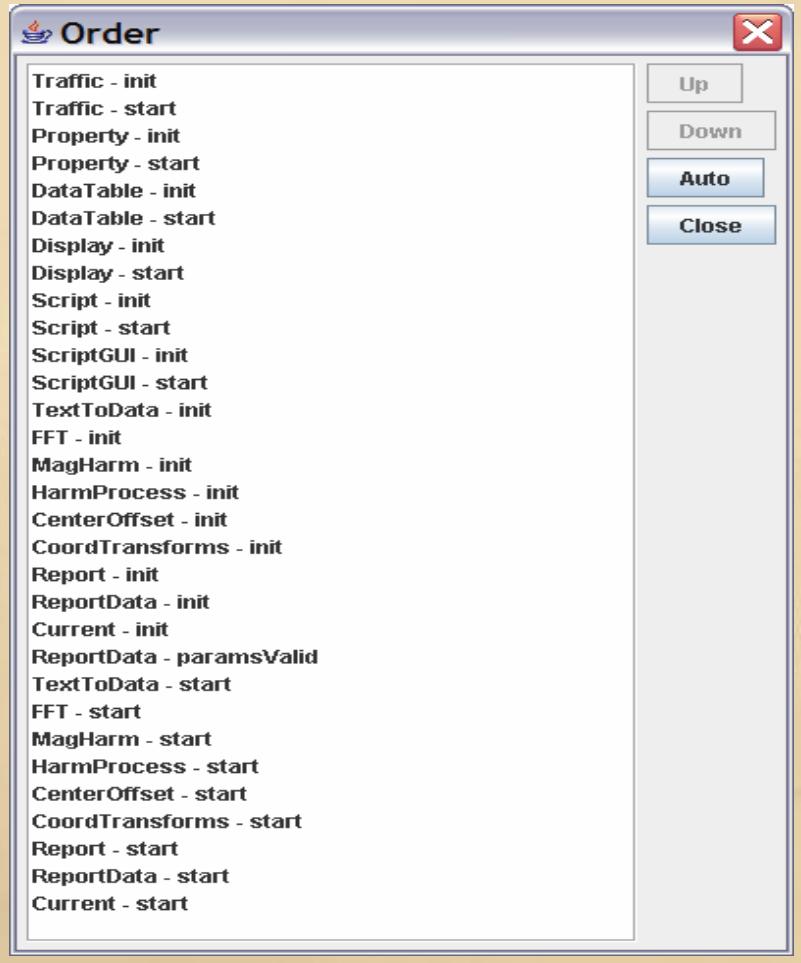
On-line Component Documentation



Configuration Comparison



Control Signal Sequencing

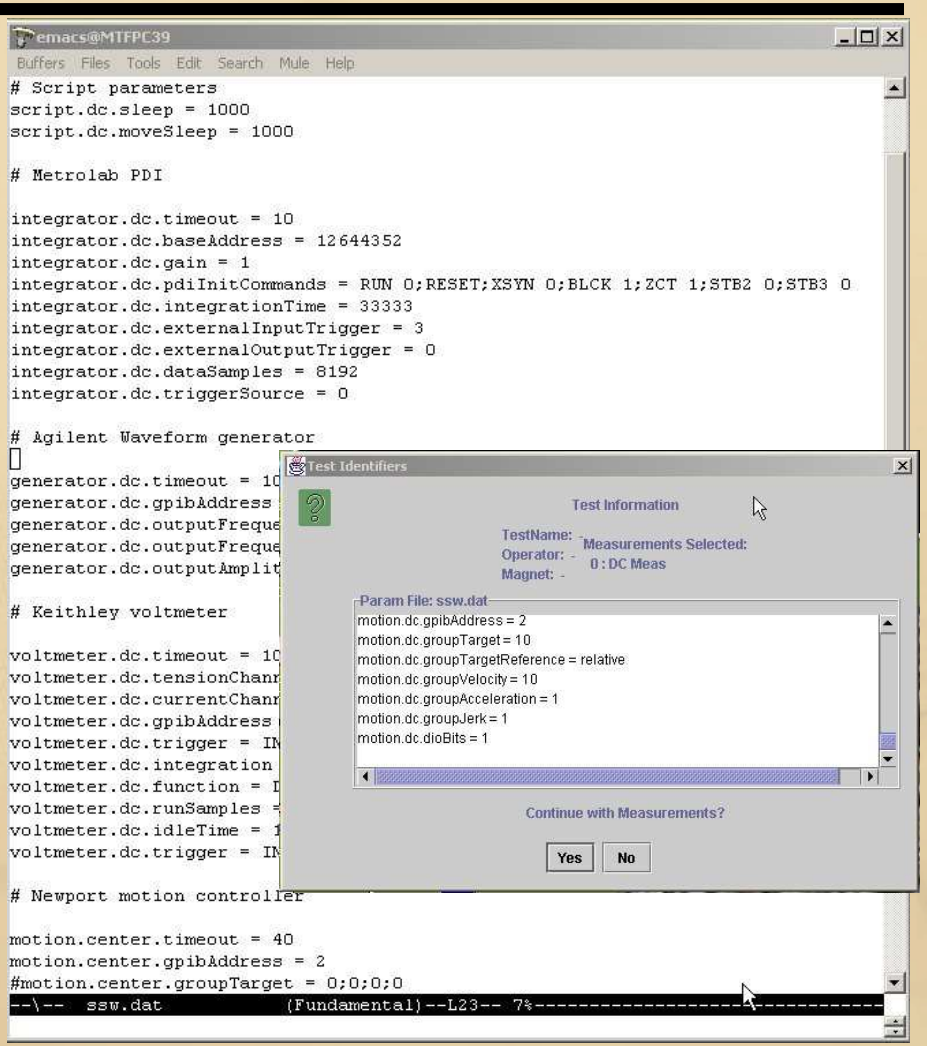


Running Applications

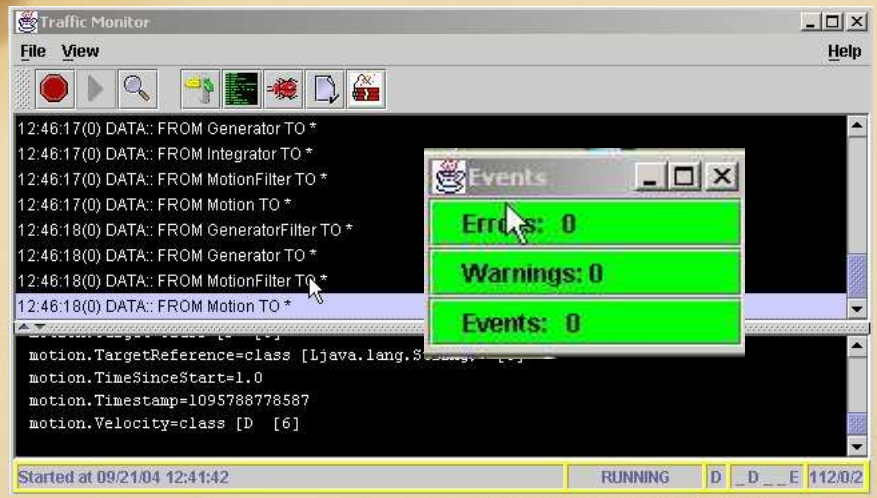
Measurement States and Parameters

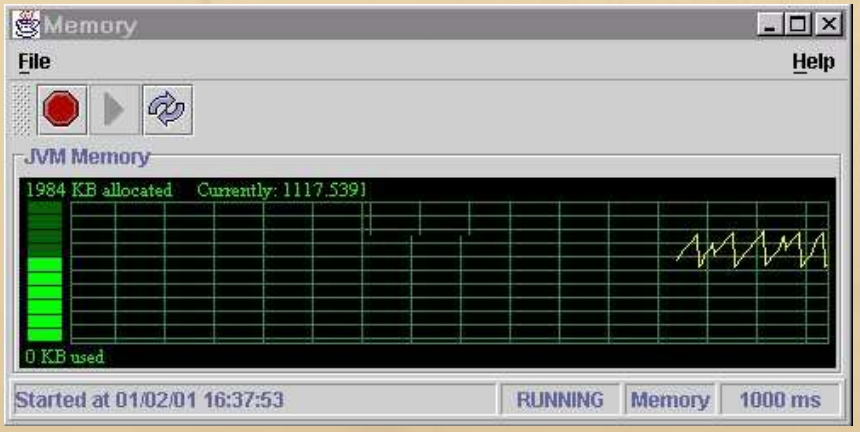
- Measurement parameters are implemented as property values of components specified for named system states.
- Parameters can be reviewed before each measurement and updated.
- A hierarchical notation is used for parameters.
- Properties/parameters can be also given on the command line:

```
ems.bat ssw.xml "Store.fileName=my_ssw.dat"
```



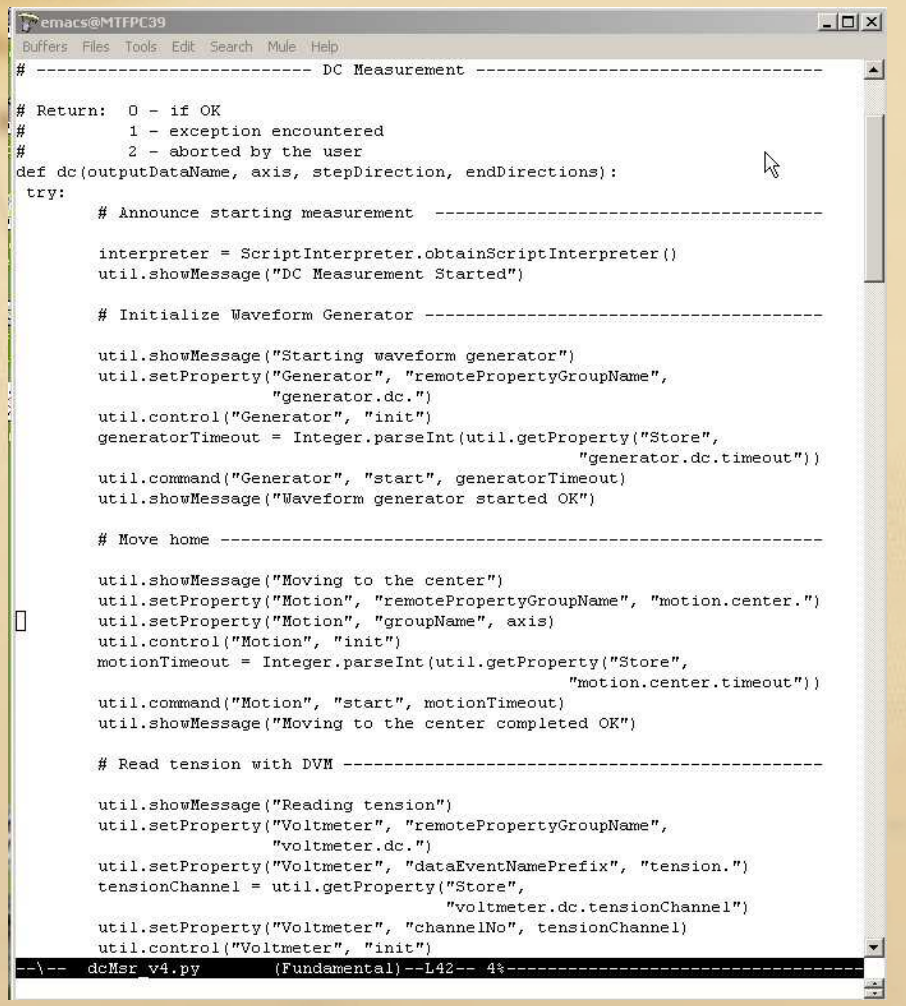
Run-time Application Monitoring





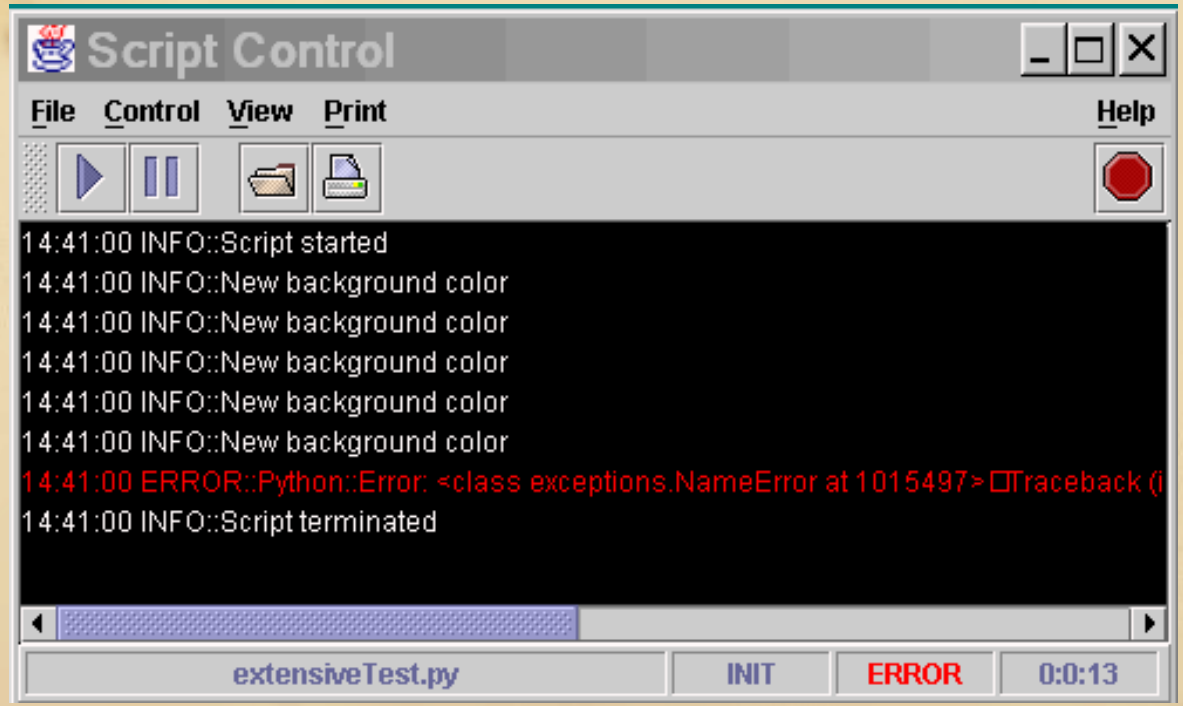
- Communication on the bus can be selectively monitored using the event monitor component.
- Memory usage can be monitored using the memory monitor component.
- The processing time used by a component can be monitored.
- Debugging and/or exception information can be routed to a separate display or I/O component.

Scripting



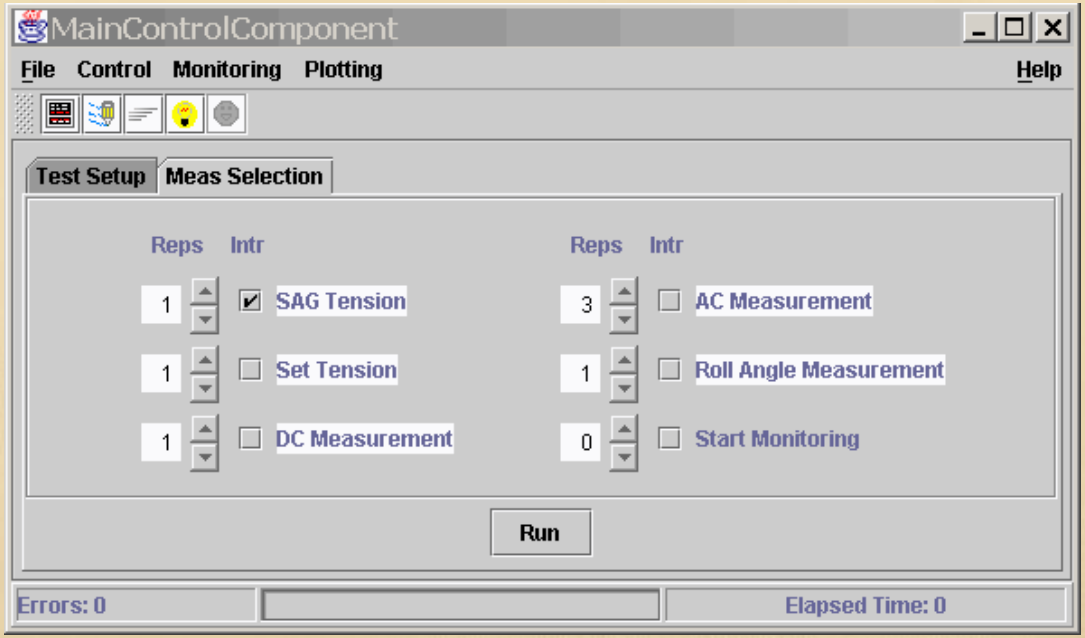
- Scripting allows for automation of measurements and for quick construction of new tests or data processing logic.
- The script interpreter is responsible for interpreting Python/Jython scripts.

Script Console



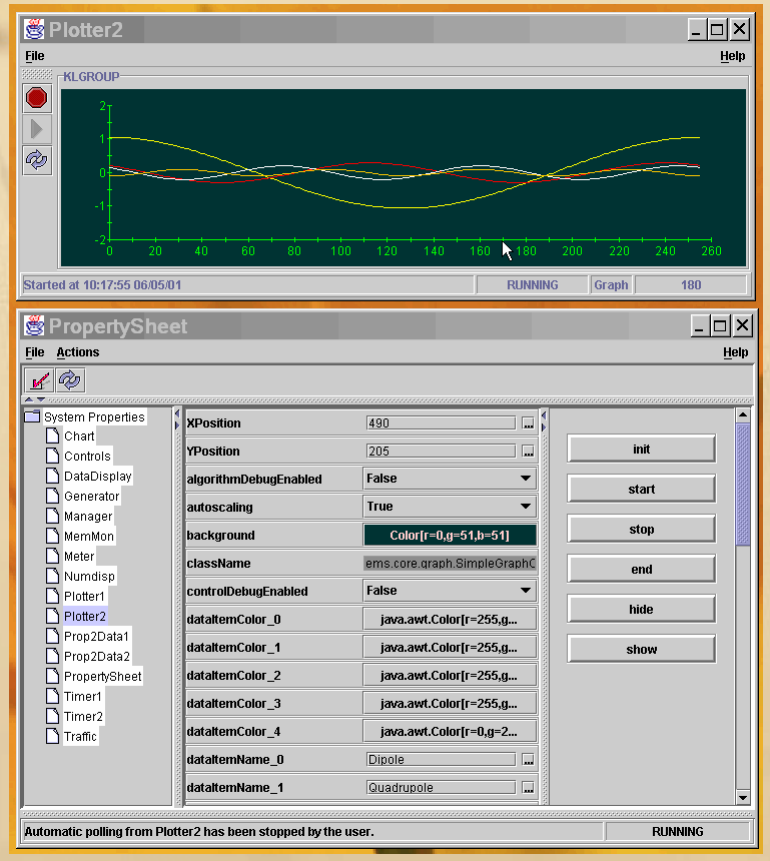
The script control panel component allows for selecting, running, and monitoring a script.

Configurable Sequences of Measurements



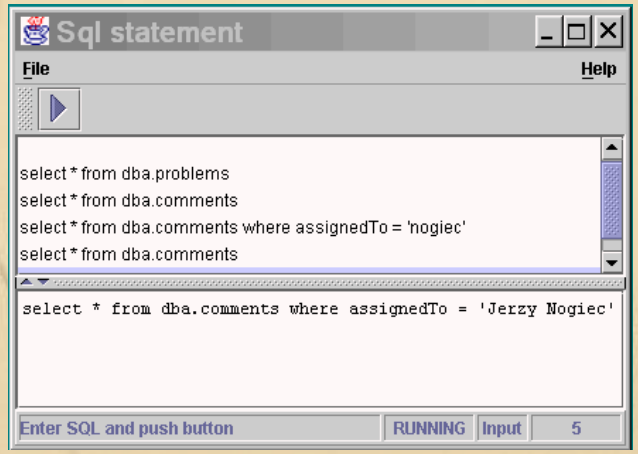
- A script can run in interactive or automatic mode.
- A sequence of measurements can be selected.
- Measurements can be repeated.

Run-time Tailoring

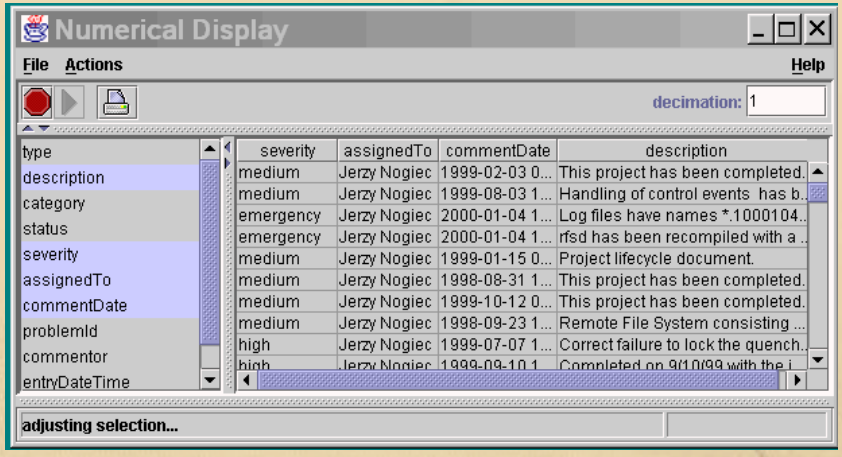


Each component has a set of properties that both control its behavior and exhibit its state. These properties can be introspected and modified at run-time.

Ad-hoc Database Queries

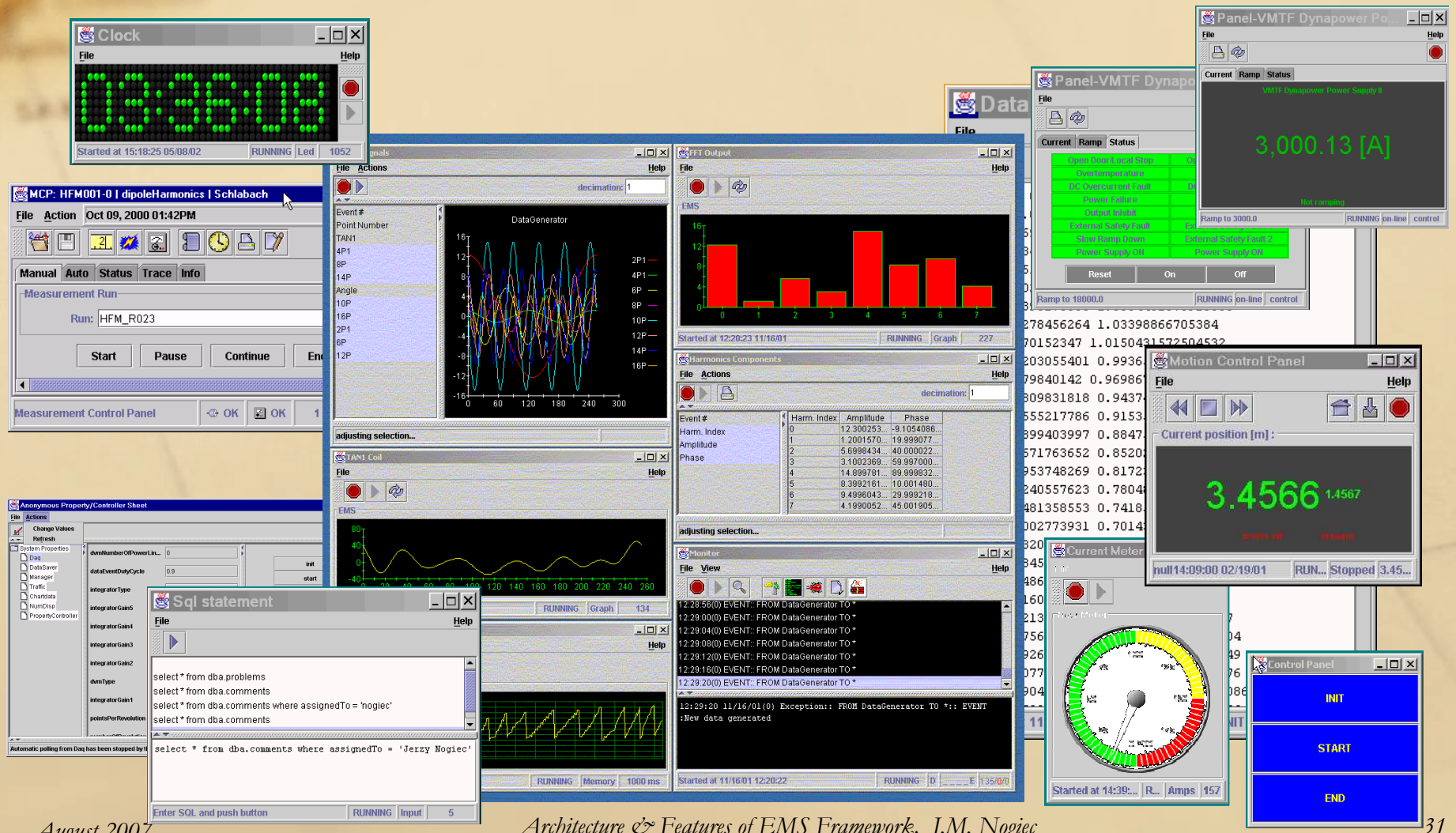


ListInput – query input

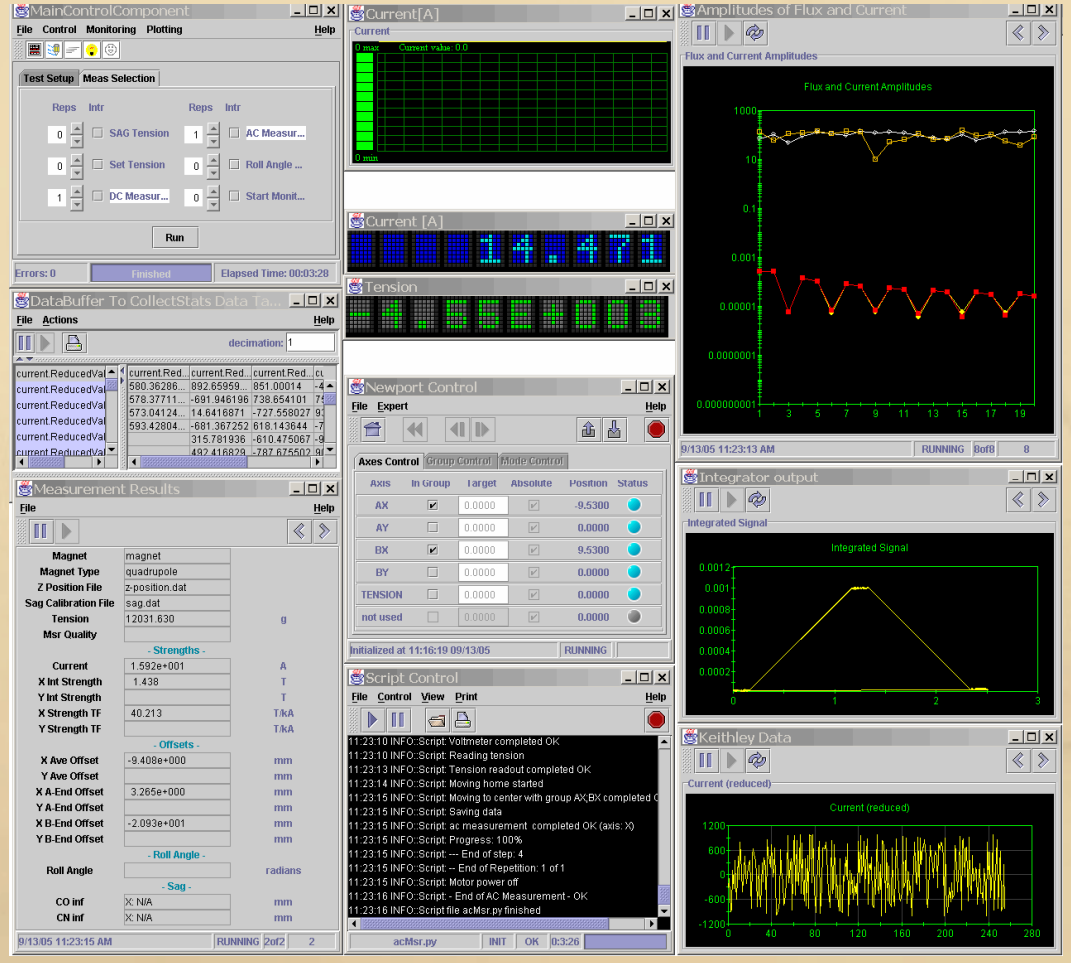


TabularDisplay – query results

Visual Components



Example UI: Tile Configuration



WrapUp

Batavia, August 2007

Features

- Multilanguage implementation: Java (framework), C/C++(DAQ), Python (scripting)
- Framework portability (Linux, Windows, Solaris)
- Self-describing data (collections of named data items)
- Non-programmers can visually configure (build) measurement applications.
- The user can save any new data without the need to modify the database schema.
- Universal components accept any collection of data items.
- Applications can be further tailored at run-time.
- Alternative configurations allow for several versions of the same measurement .
- Various DAQ solutions, either local or remote.
- Testing in the simulation mode
- Measurement sequencing via scripting
- Reuse of components for on-line and off-line processing

Summary

- EMS is a flexible component-based framework for building dynamic systems.
- EMS is suitable to be both a R&D and production measurement framework.
- The system has been applied to building measurement systems where continuous data streams are processed in real-time.
- EMS, due the selection of its platform (Java), guarantees long-term supportability of solutions, and independence from changes in OS trends and proprietary technologies.
- EMS promotes reuse and RAD.

Questions & Comments