Modeling Scientific Experiments In VL-e

Ammar Benabdelkader and Ersin C. Kaletas

Universiteit van Amsterdam

Outline

- VL-E environment for e-Science
  - 4-iter VL-e Architecture
- Approach for modeling experimental information
  - Experiment Procedure
  - Experiment Context
  - Computational Processing
- MAGE: Design and Implementation
  - MAGE-ML: Example
- Applicability of MAGE in VL-e
- MAGE/VL-e Integration

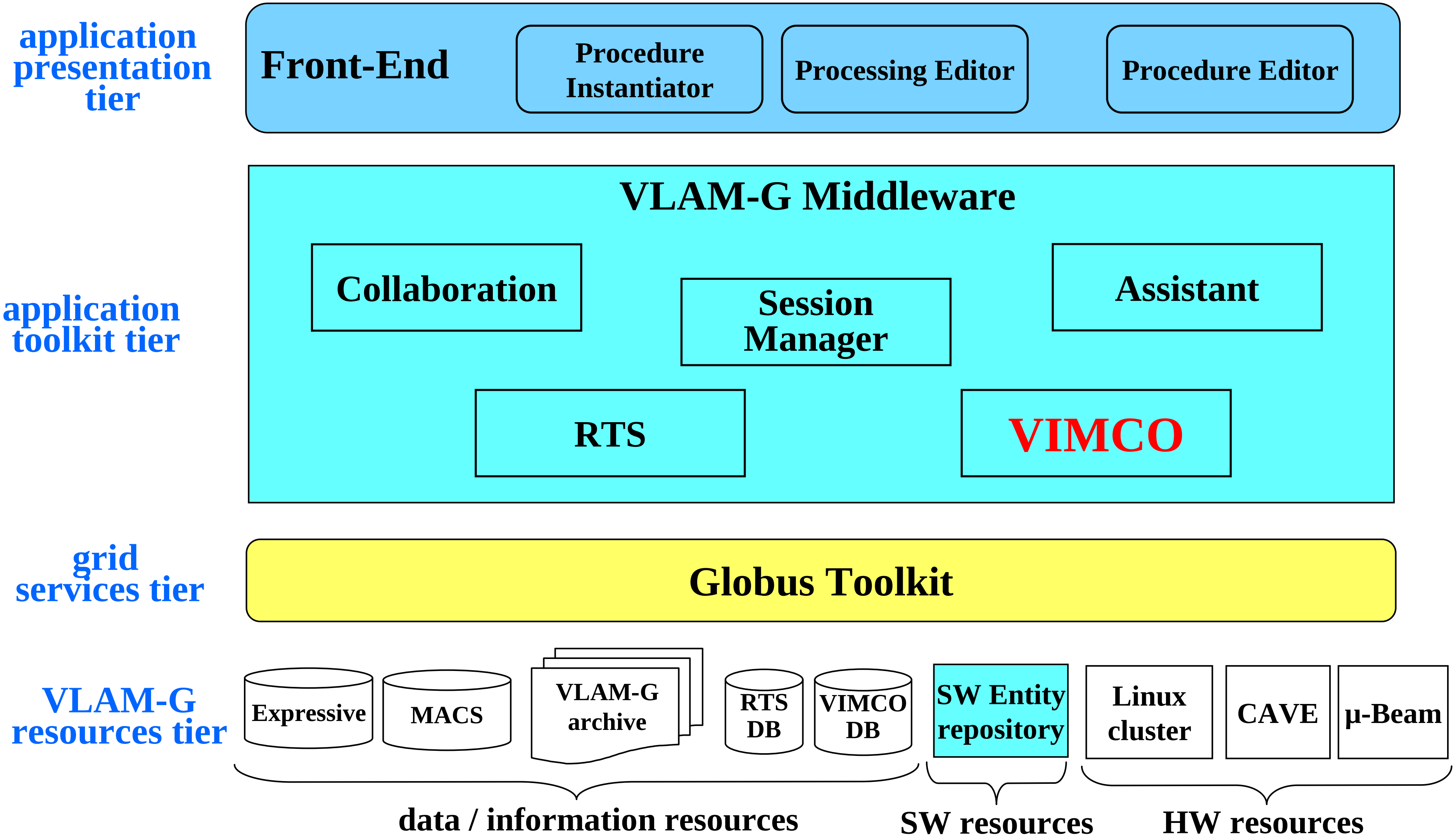
VL-e: Grid-based Virtual Laboratory Environment for e-science

A support environment for collaborative experimental science

(Dutch ICES/KIS project 1999-2003)

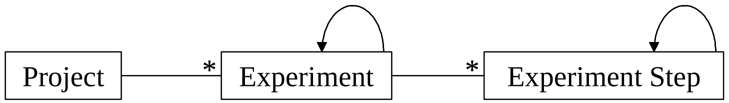
- Enable VL-e users to define, execute, and monitor their *collaborative experiments* by providing:
  - **location independent** experimentation
  - **familiar** experimentation environment
  - **assistance** during experimentation
- Designing, developing & integrating middleware to bridge the gap between Grid-Services and Application layers

4-tier Architecture of VLAM-G



Proposed Experiment Model

General Structure of Scientific Experiments



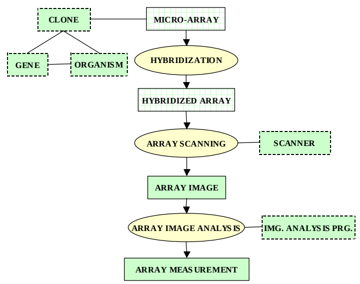
- *Scientist* performing the experiment (owner of the experiment)
- *Input* (e.g. data acquired from a device)
- A set of *activities* (e.g. computational processes), applied on the input
- *Output* (e.g. data obtained by applying a process on an input data)
- Devices (*hardware*) (e.g. to acquire data, or to perform a process)
- *Software* (e.g. to control a specific device, or to perform a specific process)
- *Conditions* and *parameters* for the processes, devices and software
- A ***recursive flow of processes and data*** where a specific order is followed during an experiment

Step-by-step definition of an experiment

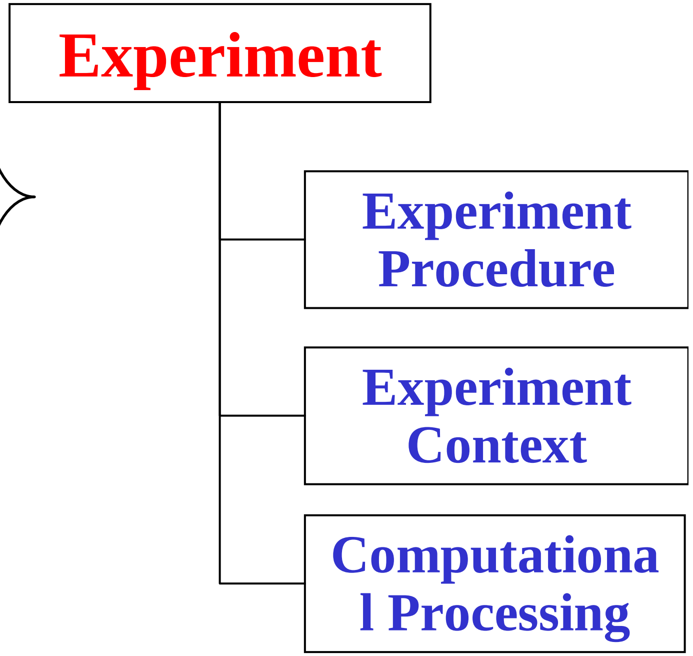
- Steps involved in an experiment
- Attributes of a step
- Relationships among steps

Common Aspects of Scientific Experiments

Process Data Flows in Experiments



uniform, *generic*, *step-wise* modeling of scientific experiments



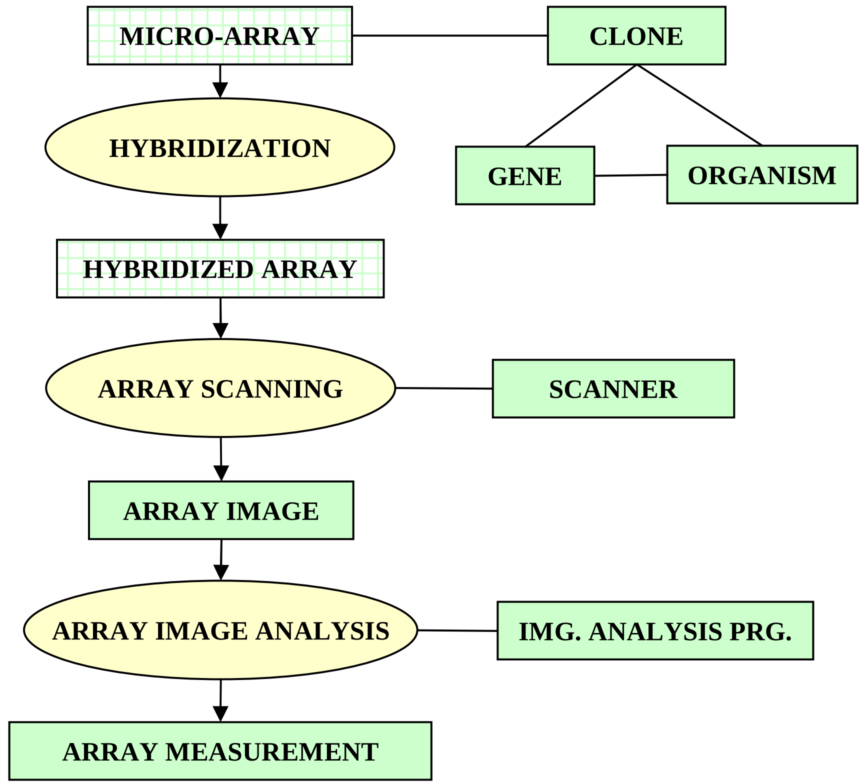
Ersin Cem Kaletas, UvA 2003

Experiment Procedure

- defines the approach to solve a particular scientific problem by defining the *typically involved steps*

➡ standardizes the experimental approach

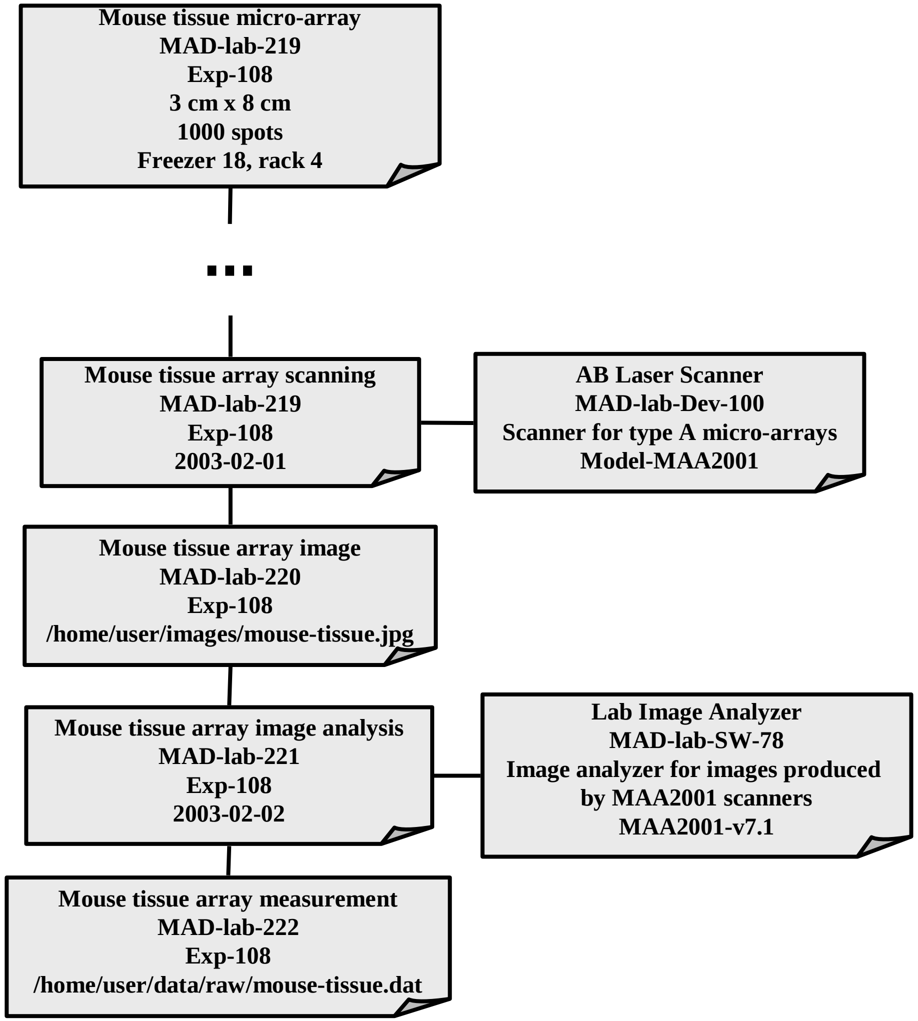
for experiments of the same type



Experiment Context

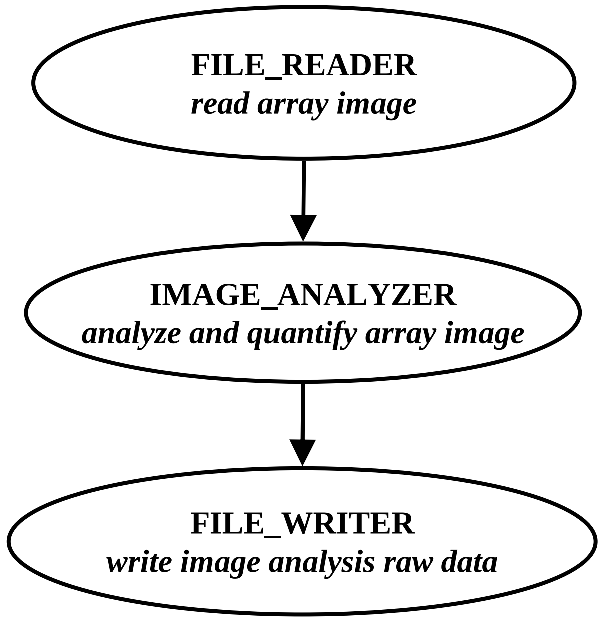
- an *instantiation* of an experiment procedure
- describes the solution – describes the *accomplishment of a particular experiment*, by providing descriptions of each step involved in the experiment

➡ provides the context for a particular experiment



Computational Processing

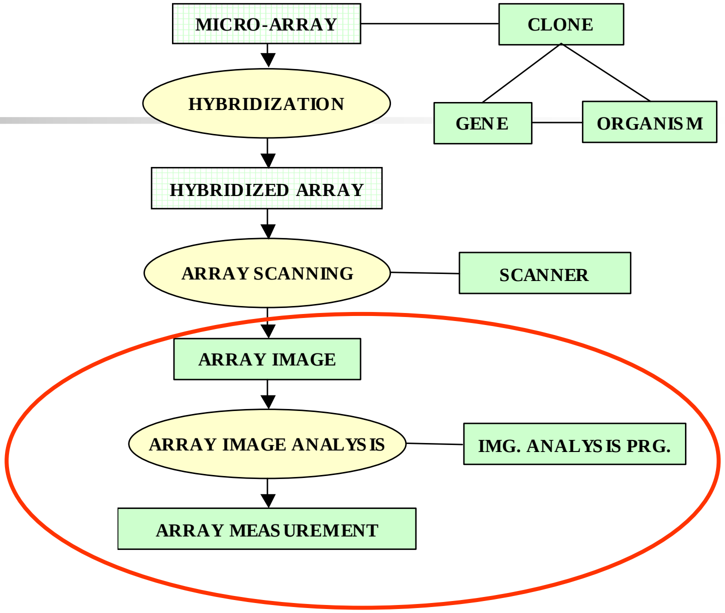
- during its processing and analysis, *data flows from one process to another*
- data flow is represented by a *directed graph*
  - Nodes => computational processes
  - connecting arcs => data flowing through the processes

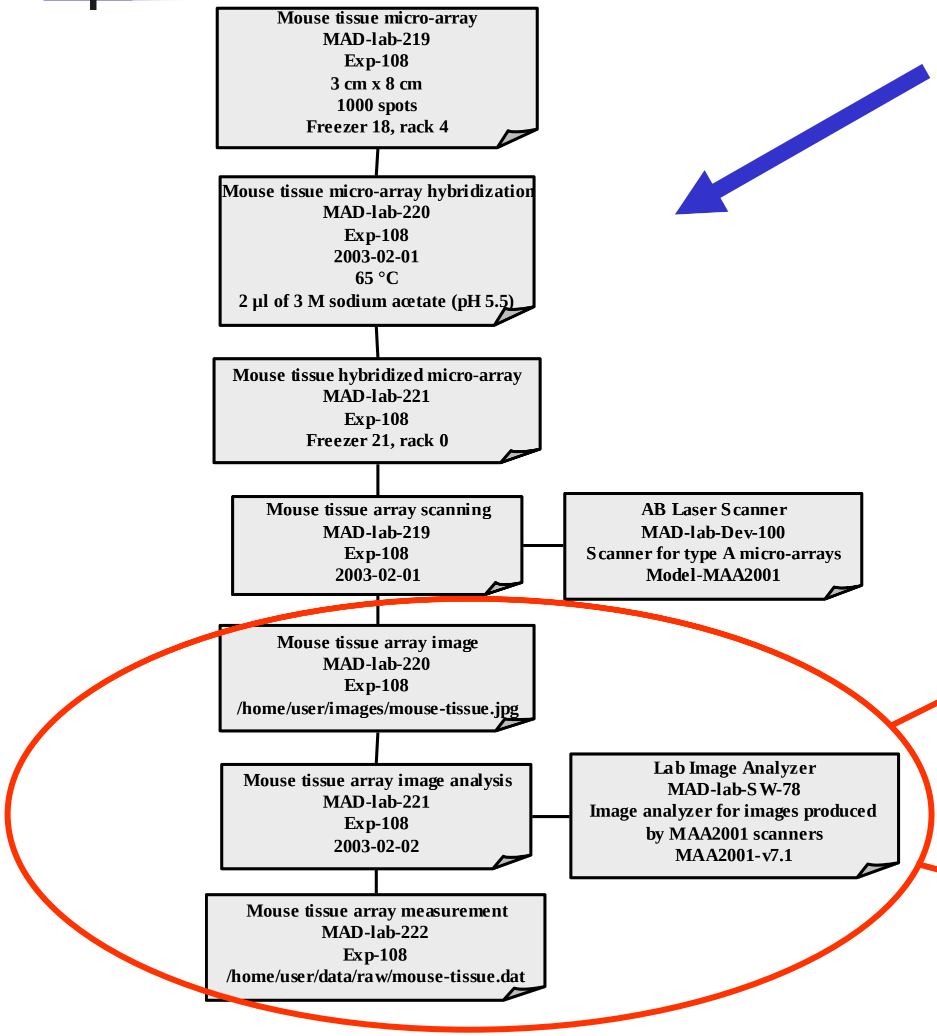


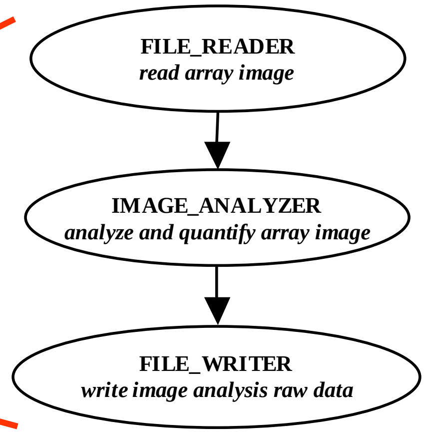
➡ This *data flow graph* is called computational processing

Experiment Context

Experiment Procedure







Computational Processing

MAGE: Design and Implementation

- A number of MAGE-family tools are freely available, from which gene-related applications can benefit.
- Compliancy of MAGE to the standards
  - Design level (MAGE-OM and MAGE-ML)
  - Implementation Level (ArrayExpress, DTD, and MAGE-ML)
- MAGE: Design level
  - MAGE-ML uses defined relationships that are implemented by the application
    - MAGE-ML does not fully comply to XML o-o descriptions and, even, to MAGE-OM
    - MAGE-ML cannot be easily used by standard XML DB tools.

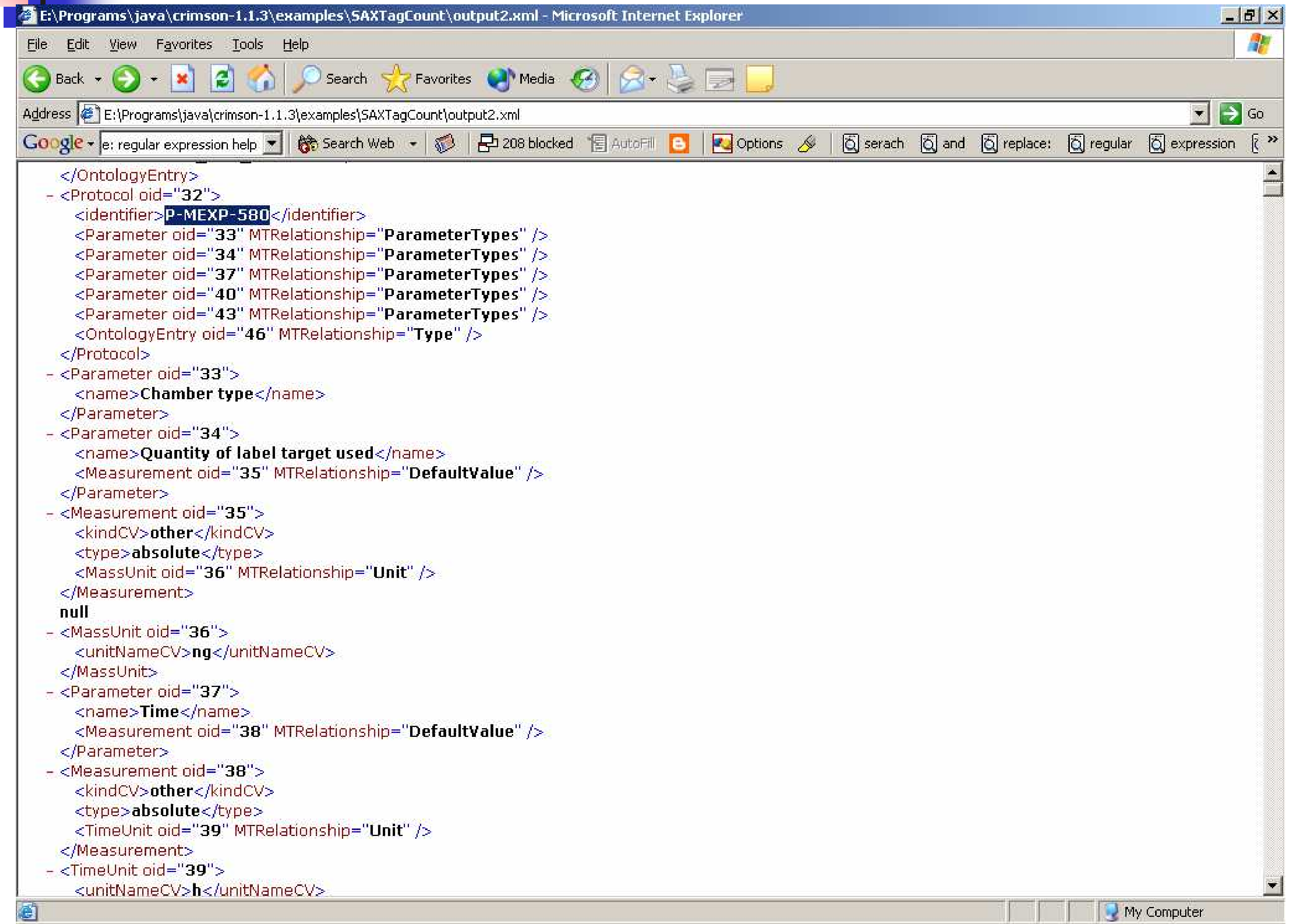
MAGE: Implementation

- MAGE-OM DTD uses defined relationships that are implemented by the application
  - MAGE DTD difficult to understand and to use
  - MAGE DTD advantages Vs. MAGE DTD complexity?
  - MAGE-ML does not comply to XML o-o descriptions
  - Future vision towards XML schemas?
- ArrayExpress database is of a higher complexity, due to the number of tables and intermediate tables (relationships, inheritance, etc)
  - MAGE-OM strength resides in its object-model (relationships associations and class inheritance)
  - The implementation of MAGE, using a relational DBMS, partially weakens its strength,
  - Future vision towards an object-oriented implementation of MAGE?

MAGE-ML: Example


```
- <Protocol identifier="P-MEXP-580" name="Hybridization protocol" text="VIB-MAF Manual hybridization v1<br> <br> A. Reagents:<br> <br> (1) PolydT35 1 ug/ul (Genset),<br> (2) Mouse COT-1 DNA(R) 1 mg/ml (Invitrogen, 500 ug, Cat. No. 18440-016),<br> (3) Human COT-1 DNA(R) 1 mg/ml (Invitrogen, 500 ug, Cat. No. 15279-011),<br> (4) Microarray hybridization buffer (Amersham Pharmacia Biotech, Cat. No. RPK0325),<br> (5) Formamide (Sigma, Cat. No. F-9037),<br> (6) Glass cover slips (Amersham Pharmacia Biotech, Cat. No. RPK0180),<br> (7) Rubber cement (e.g. Artos Strasbourg - Art'Kol KLT1).<br> <br> Use Mouse COT DNA on Mouse slides, Human COT DNA on Human slides and no COT on Arabidopsis slides.<br> <br> <br> B. Protocol for manual hybridization:<br> <br> (1) Make 30 ul hybridization mixture per slide with 25% probe, 25% buffer and 50% formamide.<br> (2) Dissolve the Cy3 and Cy5 sample together in 1 ul of polydT35, 3 ul of Mouse or Human or no (Arabidopsis) COT, water to make a total volume of 7.5 ul, 7.5 ul of Microarray 4x hybridization buffer and 15 ul of formamide.<br> (3) Denature at 96C for 3 minutes.<br> (4) Keep on ice until hybridization.<br> (5) Apply the sample on the slide, cover with a glass cover slip and seal with rubber cement.<br> (6) Keep the slide overnight in a humid cabinet at 42C.<br> <br> <br> C. Protocol for post-hybridization washes:<br> <br> (1) Carefully remove the rubber cement sealing.<br> (2) Remove the cover slip by dipping the slide in preheated (55C) 1x SSC/0.2%SDS.<br> (3) Wash the slide 10 minutes at 55C in 1x SSC/0.2%SDS, 10 minutes at 55C in 0.1x SSC/0.2%SDS, 10 minutes at 55C in 0.1x SSC/0.2%SDS and 1 minute at 37C in 0.1x SSC.<br> (4) Submerge the slide in distilled water at room temperature and dip a few times.<br> (5) Dry slide by centrifugation.<br> (6) Slide is now ready for scanning.">
  - <ParameterTypes_assnlist>
      <Parameter identifier="PARAM:CHAMBER_TYPE:1866" name="Chamber type" />
    - <Parameter identifier="PARAM:Quantity of labeled target used:1867" name="Quantity of label target used">
      - <DefaultValue_assn>
        - <Measurement type="absolute" kindCV="other">
          - <Unit_assn>
              <MassUnit unitNameCV="ng" />
            </Unit_assn>
          </Measurement>
        </DefaultValue_assn>
      </Parameter>
    - <Parameter identifier="PARAM:Time:1869" name="Time">
      - <DefaultValue_assn>
        - <Measurement type="absolute" kindCV="other">
          - <Unit_assn>
              <TimeUnit unitNameCV="h" />
            </Unit_assn>
          </Measurement>
        </DefaultValue_assn>
      </Parameter>
    - <Parameter identifier="PARAM:Volume:1871" name="Volume">
      - <DefaultValue_assn>
        - <Measurement type="absolute" kindCV="other">
```

MAGE-XML: Example



Applicability of MAGE in VL-e

- Attempts towards new in-house development, based on the current MAGE implementation, will face complex issues related to ArrayExpress database complexity and MAGE-ML user-defined relationships (as described in the previous 2 slides)
- In VL-e we will use the current implementation of MAGE, in order to make use of the freely available tools.
- In VL-e we will re-implement MAGE, using an object-oriented DBMS and a fully standard o-o XML format.
  1. MAGE-OM has to be implemented using an object-oriented DBMS (Matisse and ODL)
  2. MAGE-ML has to be translated into an object-oriented XML format
  3. Experiments in the object-oriented database should be easily translated into MAGE-ML.

MAGE/VL-e Integration